UNIVERSITY OF CALGARY

Uncertainty Models in the Context of Biometric Authentication Systems

by

Shawn C. Eastwood

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

APRIL, 2019

Abstract

This thesis focuses on developing computationally efficient machine reasoning models. These models are based on causal graphs with various metrics of uncertainty. The application of such models is decision making in a multi-sensor, multi-source system. In particular, we consider examples of biometric-enabled systems for human identification where false passes and false rejects are always present.

Two main problems are addressed in this thesis: the potential lack of data that is needed to build an accurate model, and the computational complexity (worst case computing time) of the process of deriving conclusions from the model (uncertainty inference).

To tackle the first problem, this research suggests the use of advanced models of uncertainty. These models require the development of a taxonomy of various approaches to quantifying uncertainty with the aim of being tolerant to incomplete data. Tasks related to uncertainty model design include but are not limited to:

- Model training, which is the generation of uncertainty models from raw data and expert knowledge.

- Major approaches to quantifying uncertainty include but are not limited to: probability distributions, fuzzy probability distributions, credal sets, probability interval distributions, Dempster-Shafer models, and Dezert-Smarandache models.

To address the second problem, this work develops a platform and software to perform the calculations related to the uncertainty models in a computationally efficient manner. Tasks related to the usage of uncertainty models include but are not limited to:

- Uncertainty inference, which is the calculation of likely outcomes and uncertainty values when provided with both a model of the scenario under consideration and observed evidence. This thesis covers some approximate approaches to uncertainty inference.

- Data/information fusion, which is a subset of uncertainty inference that involves the

process of collecting uncertainty values or observations from various sensors, and then generating a "recommendation".

To address the problem of computational complexity, approximate approaches will be developed and utilized in this thesis. These approximate approaches are formulated with the aim of reducing the computational complexity, while maintaining a reasonable degree of accuracy. Examples of applications of the proposed theoretical developments, including risk assessment tasks in biometric-enabled systems, are provided.

Acknowledgements

I would like to thank my supervisors Dr. Yanushkevich and Dr. Shmerko for their invaluable feedback in the preparation of this thesis. I also thank NSERC for their generous support of my research through the Postgraduate Scholarship-Doctoral (PGS D) award.

This PhD thesis is dedicated to my Mom and Dad.

Table of Contents

List of Figures and Illustrations

List of Tables

Notations

In addition to standard mathematical notations, the following notations related to sets and random variables will be used:

- Given conditions A and B: $A \wedge B$ denotes A **and** B; and $A \vee B$ denotes A **or** B.
- Given sets A and B, the set $A \setminus B$ will denote the "set difference" between A and B: set $A \setminus B$ will contain all elements that belong to A, but do not belong to B.
- Given sets A and B, the set of all functions from A to B will be denoted by $\{A \to B\}$ or B^A.
- $\mathcal{X}$ will denote the set of all random variables under consideration.
- A set that contains a single variable $\{x\}$ will be written as x instead of $\{x\}$.
- For an arbitrary set of variables $\mathcal{Y}$, the expression $\text{Val}(\mathcal{Y})$ will denote the set of all possible complete assignments to the variables in $\mathcal{Y}$. To avoid ambiguity, it will be assumed that the domain of each variable is disjoint from all other domains. If $\mathcal{Y} = \emptyset$, then $\text{Val}(\mathcal{Y})$ will contain a single element: the empty assignment ϵ. It will also be assumed for the sake of simplicity that $\text{Val}(\mathcal{Y})$ is always finite.

Notations related to assignments are:

- A specific assignment from $\text{Val}(\mathcal{Y})$ will be denoted by simply listing the assigned values. The domains of each variable are assumed to be mutually disjoint, so there is no ambiguity as to which value is being assigned to each variable. The empty assignment is denoted by ϵ.
- Given an arbitrary assignment V, $\text{Var}(V)$ will denote the set of variables that are assigned values by V.
- For arbitrary sets of variables $\mathcal{Y}, \mathcal{Z}$ and assignment $V \in \text{Val}(\mathcal{Y})$; $V[\mathcal{Z}]$ will denote the assignment to the variables in $\mathcal{Y} \cap \mathcal{Z}$ formed by dropping from V the assignments to variables not in $\mathcal{Z}$. Note that it is not necessarily the case that $\mathcal{Z} \subseteq \mathcal{Y}$.
- Given disjoint sets of variables $\mathcal{Y}_1$, $\mathcal{Y}_2$, ... and $\mathcal{Y}_n$; and assignments $V_1 \in \text{Val}(\mathcal{Y}_1)$, $V_2 \in \text{Val}(\mathcal{Y}_2)$, ... and $V_n \in \text{Val}(\mathcal{Y}_n)$; the tuple $\langle V_1, V_2, ..., V_n \rangle$ will denote the assignment to the variables in $\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup ... \cup \mathcal{Y}_n$ formed by combining the assignments V_1, V_2, ... and V_n.

Notations related to sets of assignments is:

- Given an arbitrary set of variables $\mathcal{Y}$, the expression $\text{Set}(\mathcal{Y})$ denotes the power set of $\text{Val}(\mathcal{Y})$ with the empty set removed: $2^{\text{Val}(\mathcal{Y})} \setminus \{\emptyset\}$.

- Given a set of assignments S, all assignments will cover the same set of variables, unless explicitly stated otherwise that the set is "mixed". This set of variables is denoted by $\text{Var}(S)$.

- Given a set of assignments S, S can be cylindrically projected and extrapolated to the set of variables $\mathcal{Y}$ to get a set of assignments $S[\mathcal{Y}]$ defined as follows: $\text{Var}(S[\mathcal{Y}]) = \mathcal{Y}$ and an assignment $V \in \text{Val}(\mathcal{Y})$ is a member of $S[\mathcal{Y}]$ if and only if there exists $V' \in S$ such that $V[\text{Var}(S) \cap \mathcal{Y}] = V'[\text{Var}(S) \cap \mathcal{Y}]$, which means that V and V' agree on the variables in $\text{Var}(S) \cap \mathcal{Y}$.

- Given sets of assignments S and T not necessarily over the same set of variables, the set of assignments $S|T$ denotes S restricted to T: $\text{Var}(S|T) = \text{Var}(S)$ and an assignment $V \in \text{Val}(\text{Var}(S))$ is a member of $S|T$ if and only if $V \in S$ and there exists $V' \in T$ such that $V[\text{Var}(S) \cap \text{Var}(T)] = V'[\text{Var}(S) \cap \text{Var}(T)]$, which means that V and V' agree on the variables in $\text{Var}(S) \cap \text{Var}(T)$.

- Given disjoint sets of variables $\mathcal{Y}_1$, $\mathcal{Y}_2$, ... and $\mathcal{Y}_n$; and sets of assignments $S_1 \subseteq \text{Val}(\mathcal{Y}_1)$, $S_2 \subseteq \text{Val}(\mathcal{Y}_2)$, ... and $S_n \subseteq \text{Val}(\mathcal{Y}_n)$; the product $S_1 \times S_2 \times \cdots \times S_n$ will denote the set of all assignments to the variables from $\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup ... \cup \mathcal{Y}_n$ that have the form $\langle V_1, V_2, ..., V_n \rangle$ where $V_1 \in S_1$, $V_2 \in S_2$, ... and $V_n \in S_n$.

- Given sets of assignments S_1, S_2, ... and S_n not necessarily over the same set of variables; the union $S_1 \cup S_2 \cup \cdots \cup S_n$ and the intersection $S_1 \cap S_2 \cap \cdots \cap S_n$ will both denote a set of assignments to the variables from $\text{Var}(S_1) \cup \text{Var}(S_2) \cup ... \cup \text{Var}(S_n)$. Given an arbitrary assignment $V \in \text{Val}(\text{Var}(S_1) \cup \text{Var}(S_2) \cup ... \cup \text{Var}(S_n))$, $V \in S_1 \cup S_2 \cup \cdots \cup S_n$ if and only if there exists an S_i such that $V[\text{Var}(S_i)] \in S_i$, and $V \in S_1 \cap S_2 \cap \cdots \cap S_n$ if and only if for all S_i it is the case that $V[\text{Var}(S_i)] \in S_i$. In essence, when the union or intersection of sets of assignments occurs, all sets are cylindrically extrapolated to cover the same set of variables.

Notations related to probability intervals and Dempster-Shafer models are:

- Given disjoint sets of variables $\mathcal{Y}$ and $\mathcal{Z}$, and assignments $V_\mathcal{Y} \in \text{Val}(\mathcal{Y})$ and $V_\mathcal{Z} \in \text{Val}(\mathcal{Z})$, $\Pr(V_\mathcal{Y}|V_\mathcal{Z})$ will denote the probability of the assignment $V_\mathcal{Y}$ given the assignment $V_\mathcal{Z}$. $\Pr_L(V_\mathcal{Y}|V_\mathcal{Z})$ and $\Pr_U(V_\mathcal{Y}|V_\mathcal{Z})$ are tight lower and upper bounds respectively on $\Pr(V_\mathcal{Y}|V_\mathcal{Z})$, and $\Pr_I(V_\mathcal{Y}|V_\mathcal{Z})$ denotes the interval $[\Pr_L(V_\mathcal{Y}|V_\mathcal{Z}), \Pr_U(V_\mathcal{Y}|V_\mathcal{Z})]$.

- Given disjoint sets of variables $\mathcal{Y}$ and $\mathcal{Z}$, and subset $B_\mathcal{Y} \in \text{Set}(\mathcal{Y})$ and assignment $V_\mathcal{Z} \in \text{Val}(\mathcal{Z})$, $m(B_\mathcal{Y}|V_\mathcal{Z})$ will denote the Dempster-Shafer mass assigned to $B_\mathcal{Y}$ after the variables in $\mathcal{Z}$ are conditioned to $V_\mathcal{Z}$, and all variables not in $\mathcal{Y} \cup \mathcal{Z}$ have been removed via marginalization.

Chapter 1

Introduction

This chapter will introduce the goals of the work presented in this thesis, as well as the original contributions. The applications of the work presented in this thesis will be given, as well as a listing of publications that present the proposed contributions.

1.1 Background

E-borders (also known as Smart Borders and automated border control) are one of the most ambitions national and international projects which assume close cooperation between countries based on advanced border automation technologies [23, 54, 3]. The future 2020+ border automation prospects are outlined by the International Air Transport Association (IATA) in [6]. This roadmap predicts in the near future novel and effective technological solutions. As part of these future solutions, this thesis tackles the problem of risk assessment in e-border and biometric authentication scenarios. Risk is a quantity that describes the likelihood, cost, or some combination thereof, of an undesirable outcome. Uncertainty is the indefiniteness about the outcome of a situation or event. This thesis will propose approaches to analyzing the uncertainties in e-border scenarios, such as incomplete data about an individual, for the purpose of measuring the risks of decision-making regarding the individual's authentication.

Risk assessment tools should be:

- Computationally efficient:
  - Risk assessment occurs in real time.
  - Probabilistic inference using Bayesian networks is NP-hard.
- Robust and versatile:
  - Risk assessment needs to be insensitive to inaccuracies and deficiencies in the data needed to construct the mathematical model used to represent the target scenario.
- Mathematically meaningful and objective:
  - Mathematical objectivity will give concrete meaning to the posterior computed quantities and recommendations.

While there exist software suites such as "ARENA" [1] for simulating processes that have a strong random element using Monte Carlo techniques, this thesis aims for real time authentication. Real time authentication involves conditioning the mathematical models with accumulated data. Conclusions are extracted from the models themselves via inference, without the need for simulations.

1.2 Problem Formulation and Hypothesis

The primary contribution of this thesis is a taxonomy of various approaches to quantifying uncertainty with the aim of deriving accurate conclusions in a computationally efficient manner. Three main problems are addressed in this thesis: the worst case computing time (computational complexity); the potential lack of data that is needed to build an accurate model; and the need for mathematical objectivity in the computed risk values. The main application for the approaches that are developed in this thesis will be topics related to human biometric authentication. Such topics include, in particular, automated border

control (ABC), traveler risk profiling, watchlist identification, and Doddington classification (Doddington classification is described in [70]).

This thesis proposes the following approaches to the quantification of uncertainty, the generation of uncertainty models, and the derivation of conclusions (uncertainty inference):

- The modularization of Bayesian networks and other graphical uncertainty models as an approach to mitigating the computational complexity of uncertainty inference (chapter 4).

- The generalization of graphical models of uncertainty to various uncertainty metrics. Non-probabilistic uncertainty metrics address the problem of insufficient data when building a model (chapter 5).

- A taxonomy of existing and new algorithms for data/information fusion using convex sets of probabilities known as credal sets. Data fusion is a special instance of uncertainty inference. A taxonomy of fusion algorithms provides a road map for the developers of practical systems to choose an appropriate algorithm (chapter 6).

- A novel structure for mitigating the computational complexity of both model training and uncertainty inference, by building up an uncertainty model using "layers". This approach yields benefits in both the mitigation of the model's complexity, and the computational complexity of probabilistic inference (chapter 7).

1.3 Proposed Contributions

The following list describes the novel material that is introduced in the 4 content chapters.

- Modular models (chapter 4):
  - In the proposed concept of linking Bayesian network modules, the posterior probabilities stored in the intermediate nodes are used to compute the priors in the next

network using a variable function instead of a direct substitution of the posterior probability. An entire probability distribution is returned from a network/module as opposed to a single estimated value.

  - In [40, pg.288–290], it is shown that inference using an arbitrary Bayesian network is NP-hard. This makes inference using large Bayesian networks computationally intractable. By breaking the network into sub-modules using the paradigm described in our approach, the size of each sub-network is limited and inference remains computationally tractable.

- Graphical models using generalized uncertainty metrics (chapter 5):

  - The primary contribution is a survey of how graphical models that utilize non point probability models of uncertainty such as fuzzy probabilities, probability intervals, Dempster-Shafer theory, and Dezert-Smarandache theory can be established and the various operations/algorithms involved.

  - The generalization of Bayesian networks to Dempster-Shafer theory formalizes and expands the theory that is introduced in [22]. Emphasis is placed on the generalization of conditional probability tables to Dempster-Shafer theory, and their use in the Dempster-Shafer analog to Bayesian networks.

- A taxonomy and analysis of information fusion approaches (chapter 6):

  - The most important contribution of this chapter is a taxonomy and catalog of fusion approaches and algorithms that utilize "subtypes" of credal sets, in this case "probability interval distributions" and "Dempster-Shafer models". All fusion approaches will satisfy an important objective criteria, referred to in this paper as the "containment property". Special attention is paid to the computational challenges involved. Various approaches are given, which exhibit trade-offs between accuracy and computational complexity. Some of the fusion approaches

are already known to the literature (such as context specific fusion with probability intervals described in [68]), and others were created specifically for this thesis.

  - A proposed objective criteria for information fusion referred to as the "containment property" (see section 6.4 for the definitions) is given. Dempster's rule of combination is shown to violate the containment property.

  - A distinction is made between two types of information fusion, referred to as "context specific" and "general fusion". Each type of fusion has different information requirements, and the algorithms are different. Context specific fusion requires more prior information, but is less computationally intensive than general fusion. The important distinction between context specific fusion and general fusion is that context specific fusion only requires raw observations as input, while general fusion requires complete credal sets. Context specific fusion follows the hypothesis-observation models used in publications such as [18, 81] and [68, section 4, calculus], and the algorithms are generally polynomial time with respect to the size of the input. General fusion is similar to the direct Bayesian fusion of credal sets. While the direct Bayesian fusion of credal sets can be performed exactly in polynomial time [33, Theorem 2], when credal sets are restricted to specific subtypes, general fusion becomes much more difficult.

  - An NP-hard problem related to general fusion is identified and is shown to be NP-hard.

- Layered probability models (chapter 7):

  - The goal of the material presented in this chapter is to present an alternative approach to probability distributions and graphical probability models. This approach will address problems inherent to creating a graphical probability model, and performing uncertainty inference with said graphical probability model. By

decomposing a probability distribution into a stack of "layers", "unimportant" layers can be omitted which will simplify the process of uncertainty inference. Most importantly however, the process of constructing such models from data is engineered to be simple.

1.4 Publications

Publications submitted and accepted regarding the content chapters are listed below:

- Chapter 4:
  - **Conference Article:** Eastwood, S. C., and S. N. Yanushkevich, Risk Profiler in Automated Human Authentication, *IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES) at the Symposium Series on Computational Intelligence (SSCI 2014)*, FL, USA, December 2014, pp. 140-147.
  - **Conference Article:** Eastwood, S. C., and S. N. Yanushkevich, Modeling Risks in Biometric-Based Authentication Control Systems, *Fifth International Conference on Emerging Security Technologies (EST 2014)*, Spain, September 2014, pp. 2-7.
- Chapter 5:
  - **Submitted Article:** O. Obi-Alago, P. Kozlow, A. Noor, S. T. Gnanasekar, S. C. Eastwood, H. M. Wetherley, and S. N. Yanushkevich, Biometric-Enabled Risk Assessment for City Emergency Shelters, submitted to *IEEE Transactions on Computational Social Systems* during July 2018, Reference number: TCSS-2018-07-0102
  - **Submitted Article:** Eastwood, S. C., S. N. Yanushkevich, and V. P. Shmerko, Multi-State Parallel-Pipeline Traveler Screening Model and Inference Engine for

Mass-Transit Hubs, submitted to *IEEE Transactions on Intelligent Transportation Systems* during June 2018, Reference number: T-ITS-16-12-0926.R3.

  - **Published Article:** Yanushkevich, S. N., S. C. Eastwood, M. Drahansky, and V. P. Shmerko, Understanding and Taxonomy of Uncertainty in Modeling, Simulation, and Risk Profiling for Border Control Automation, *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 15, no. 1, 2018, pp. 95-109.
  - **Published Article:** Lai, K., S. Eastwood, W. A. Shier, S. N. Yanushkevich, and V. P. Shmerko, Mass Evidence Accumulation and Traveler Risk Scoring Engine in e-Border Infrastructure, *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, 2018, pp. 3271-3281.
  - **Published Book Chapter:** Eastwood, S. C., and S. N. Yanushkevich, Risk Assessment in Authentication Machines, *Recent Advances in Computational Intelligence in Defense and Security*, vol. 621, 2016, pp. 391-420.
  - **Unpublished Article:** Eastwood, S. C., and S. N. Yanushkevich, Graphical Dempster-Shafer Models of Uncertainty, 13 pages.

- Chapter 6:

  - **Unpublished Article:** Eastwood, S. C., S. N. Yanushkevich, and V. P. Shmerko, A Taxonomy of Interpretations and Algorithms for Information Fusion, 26 pages.

Chapter 2

Literature Review

The following sections will catalog existing work done in the areas covered by chapters 4 to 7.

2.1 Literature Related to Modular Models

A modular approach to graphical uncertainty models refers to dividing a large graphical model, in particular a Bayesian network, into smaller submodules for the purpose of reducing the computational complexity of probabilistic inference. Table 2.1 contains a list of existing work related to this modular approach.

2.2 Literature Related to Generalized Graphical Uncertainty Models

A generalized graphical uncertainty model is a graphical model of uncertainty such as a Markov or Bayesian network to which a non-probabilistic metric has been applied. Table 2.2 contains a list of existing work related to generalizing graphical models of uncertainty to non-probabilistic models.

Table 2.1: A summary of the literature related to modular models.

Reference(s)	Contribution
[28]	Separate Bayesian networks are connected via "virtual links". A virtual link runs from a specific node in the source network to a specific node in the destination network. The posterior probability of the source node becomes the prior probability of the destination node.
[35]	The output of multiple computational modules (which may not necessarily be Bayesian networks) are merged into a single output through the use of a relatively simple Bayesian network. The output of each module is a single value that is treated as an evidence value in the Bayesian network that is merging the outputs.
[41, 55]	A large Bayesian network is simply subdivided into smaller networks. Directed edges that exist between nodes can run from one module to another.
[8, 29, 50, 57]	"Clique trees" are used to provide structure to the process of variable elimination. Each node in the clique tree denotes a clique of nodes from the original Markov network. Each edge in the clique tree denotes the intersection between the two cliques of the connected clique tree nodes. "Messages" are passed along edges of the clique tree to enable the eventual computation of the marginal probability over each clique. This process is used in the "HUGIN" shell [8].
[42, 50, 52, 69]	A set of Markov network factors is denoted using a bipartite graph called a "factor graph", where the variables and factors form the different partitions. An iterative process referred to in [40] as "loopy belief propagation" is used to circulate "messages" between the variables and factors until convergence is achieved. Improvements to the convergence are proposed in [52]. A generalization of factor graphs that consist of multiple partitions with factors of varying complexity is used in [69].

2.3 Literature Related to Information Fusion

The problem of "fusion" stems from the need to combine information from various sources. Each of these source is assumed to provide either an observation, or a "model of uncertainty". Table 2.3 contains a list of existing work related to fusing uncertainty models.

Table 2.2: A summary of the literature related to generalized graphical uncertainty models.

Reference(s)	Contribution
[11, 53]	The formulation of fuzzy probabilities as a replacement for probability distributions.
[17, 25, 68]	The formulation of probability intervals as a replacement for probability distributions.
[18]	The transferable belief model (a generalization of Bayes' rule), ballooning extensions and a two variable implementation of the Dempster-Shafer analogs to conditional probability tables which we will refer to as "conditional Dempster-Shafer tables".
[30, 31]	Undirected (non-causal) and directed (causal) graphical models that describe Dempster-Shafer models over a relatively large number random variables. The rule of combination that is used however is not analogous to Dempster's rule of combination however.
[38, 26, 39]	Establishes a Dempster-Shafer analog to Markov networks (factors and belief functions are referred to as "valuations"). An analysis of Dempster-Shafer analogs to the well known probabilistic inference algorithms: variable elimination and belief propagation is given.

2.4 Literature Related to Layered Probability Models

The "layered probability model" that is presented in chapter 7 aims to address the difficulties in generating graphical probability models from raw data, as well as simplify the process of probabilistic inference by allowing the omission of layers. Omitting layers yields meaningful approximations. Table 2.4 contains a list of existing work related to both generating graphical probability models from raw data and work that entails the use of "layered models".

2.5 Conclusion

The literature listed in this chapter provides a survey of the current state-of-the-art methodologies related to chapters 4 to 7. This thesis in chapter 4 will present a novel method of linking Bayesian network modules from a library of modeling modules to form a "modular model" of a target scenario. This approach to model building is intended to resolve issues related to computational complexity, as well as provide a flexible tool for practitioners to

Table 2.3: A summary of the literature related to information fusion.

Reference(s)	Contribution
[18]	The transferable belief model.
[72]	Compatibility Relationships: Compatibility functions are functions that determine the "closeness" or "compatibility" between two outcomes. Compatibility functions are used to determine the most "likely" outcome from a set of seemingly contradictory input outcomes.
[73]	Uninorm Aggregation: Functions referred to as "uninorm"s were analyzed as candidates for combining several strengths of belief into a single strength of belief. A "uninorm" is a function for combing values from the range [0,1] into a single value from the range [0,1]. In a uninorm function, a specific value that serves as the identity can be arbitrarily chosen and the function built around it.
[32, 33, 34]	The fusion of credal sets as models of uncertainty and the establishment that fusion can be performed in an efficient and exact manner when credal sets are denoted by listing their extreme points.
[71, 75]	Dempster-Shafer combination/fusion (described in section 3.3.4)
[19, 66, 64]	Dezert-Smarandache fusion.
[36]	A survey of contemporary fusion approaches, most of which rely on quantities such as "strengths of belief" that are highly subjective. In addition, the heuristics used to handle strengths of belief are algorithms designed so that the outputs "make sense" as opposed to obeying an objective criteria.
[10]	The creation of a software package that calculates posterior credal sets such as "CREDO".

use when modeling scenarios. In chapter 5, the development of various uncertainty models in previous publications is combined to provide a multi-metric approach to the use of causal networks that account for variations in the available statistical data, and the requirements of the output models. Chapter 6 builds a taxonomy of information fusion approaches that are centered around convex sets of probability distributions known as "credal sets". Some of the fusion approaches are well known to the literature, while others are constructed

Table 2.4: A summary of the literature related to layered probability models.

Reference(s)	Contribution
[40]	A summary of various approaches to training Markov and Bayesian networks. Approaches that determine the numerical values that populate factors and conditional probability tables include Maximum Likelihood Estimation and Expectation Maximization. Approaches that determine the structure of Markov and Bayesian networks include a variety of optimization approaches.
[45]	[45] describes a software tool known as the "Bayesian Network Toolbox for Matlab". This multi-purpose tool box uses a variety of algorithms to compute both the parameters and structure of Markov and Bayesian networks. Parameter learning uses the well established maximum likelihood and maximum posterior estimation, as well as expectation maximization. Structure learning uses approaches that search for an optimal model while respecting specified constraints.
[49]	A stack of "Hidden Markov Models" is trained and used for a specific application, which in the context of this paper, is tracking office activity.

here to complete the taxonomy. Lastly chapter 7 provides a novel alternative to traditional graphical models of uncertainty such as Markov and Bayesian networks. This alternative addresses the shortcomings of Markov and Bayesian networks such as the difficulty of variable marginalization and model construction.

Chapter 3

Background

The purpose of this chapter is to provide the reader with the necessary mathematical and conceptual background to properly understand the contributions in the later chapters. No part of this chapter consists of original work, and exists purely as a reference. Topics covered include:

- Probability theory and probabilistic inference.
- Graphical models of probability distributions, namely, Markov and Bayesian networks.
- Non-probabilistic measures of uncertainty that serve as alternatives to probability theory. Non-probabilistic alternatives include: fuzzy probabilities, probability intervals, Dempster-Shafer models, and Dezert-Smarandache models.
- Risk analysis and risk fusion.
- Shannon entropy and information theory.

3.1 Probability and Random Variables

Consider a mathematical model that consists of a complete set of random variables $\mathcal{X}$.

Given an arbitrary assignment $V \in \text{Val}(\mathcal{X})$, the **joint probability** of V, denoted by $\Pr(V)$, is the fraction of outcomes that match V when the variables from $\mathcal{X}$ are sampled a near infinite number of times. More generally, given a set $S \subseteq \text{Val}(\mathcal{X})$, the joint probability of S is $\Pr(S) = \sum_{V \in S} \Pr(V)$. It must be the case that $\sum_{V \in \text{Val}(\mathcal{X})} \Pr(V) = 1$.

Given an arbitrary subset of variables $\mathcal{Y} \subseteq \mathcal{X}$, and assignment $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$, the **marginal probability** of $V_{\mathcal{Y}}$, denoted by $\Pr(V_{\mathcal{Y}})$ is the fraction of outcomes that match $V_{\mathcal{Y}}$ when the variables from $\mathcal{X}$ are sampled a near infinite number of times. It is the case that $\Pr(V_{\mathcal{Y}}) = \sum_{V' \in \text{Val}(\mathcal{X} \setminus \mathcal{Y})} \Pr(\langle V_{\mathcal{Y}}, V' \rangle)$. More generally, given a set $S \subseteq \text{Val}(\mathcal{Y})$, the marginal probability of S is $\Pr(S) = \sum_{V_{\mathcal{Y}} \in S} \Pr(V_{\mathcal{Y}})$.

Given arbitrary disjoint subsets of variables $\mathcal{Y}, \mathcal{Z} \subseteq \mathcal{X}$, and assignments $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$ and $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$, the **conditional probability** of $V_{\mathcal{Y}}$ given $V_{\mathcal{Z}}$, denoted by $\Pr(V_{\mathcal{Y}}|V_{\mathcal{Z}})$ is the fraction of outcomes that match $V_{\mathcal{Y}}$ when the variables from $\mathcal{X}$ are sampled a near infinite number of times, not counting the outcomes that fail to match $V_{\mathcal{Z}}$. It is the case that $\Pr(V_{\mathcal{Y}}|V_{\mathcal{Z}}) = \frac{\Pr(\langle V_{\mathcal{Y}}, V_{\mathcal{Z}} \rangle)}{\Pr(V_{\mathcal{Z}})}$.

An important property that exists between sets of random variables is the property of conditional independence:

Definition 3.1. Given disjoint sets of variables $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_N, \mathcal{Z} \subseteq \mathcal{X}$, the sets $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_N$ are **conditionally independent** when given $\mathcal{Z}$ if the following holds for all $V_1 \in \text{Val}(\mathcal{Y}_1)$, $V_2 \in \text{Val}(\mathcal{Y}_2)$, ..., $V_N \in \text{Val}(\mathcal{Y}_N)$, and $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$:

$$\Pr(\langle V_1, V_2, \ldots, V_N \rangle | V_{\mathcal{Z}}) = \prod_{i=1}^{N} \Pr(V_i | V_{\mathcal{Z}})$$

Conditional independence is denoted by $\mathcal{Y}_1 \perp \mathcal{Y}_2 \perp \cdots \perp \mathcal{Y}_N | \mathcal{Z}$.

When $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_N$ are **conditionally independent** given $\mathcal{Z}$ ($\mathcal{Y}_1 \perp \mathcal{Y}_2 \perp \cdots \perp \mathcal{Y}_N | \mathcal{Z}$), the sampling of each set $\mathcal{Y}_i$ may proceed independently of the other sets while throwing away any sample that does not match the chosen assignment to $\mathcal{Z}$.

Context specific independence [40] is a weaker form of conditional independence that

holds only for a specific assignment to the variables from $\mathcal{Z}$:

Definition 3.2. Given disjoint sets of variables $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_N, \mathcal{Z} \subseteq \mathcal{X}$, as well as an assignment $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$, the sets $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_N$ are **context specifically independent** when given $V_{\mathcal{Z}}$ if the following holds for all $V_1 \in \text{Val}(\mathcal{Y}_1)$, $V_2 \in \text{Val}(\mathcal{Y}_2)$, ..., $V_N \in \text{Val}(\mathcal{Y}_N)$:

$$\Pr(\langle V_1, V_2, \ldots, V_N \rangle | V_{\mathcal{Z}}) = \prod_{i=1}^{N} \Pr(V_i | V_{\mathcal{Z}})$$

Context specific independence is denoted by $\mathcal{Y}_1 \perp \mathcal{Y}_2 \perp \cdots \perp \mathcal{Y}_N | V_{\mathcal{Z}}$.

A common and important task related to probability models is "probabilistic inference", more generally referred to as "uncertainty inference":

Problem 3.3. Given a set of query variables, $\mathcal{Y}$, a disjoint set of evidence (observed) variables, $\mathcal{Z}$ where $\mathcal{Y} \cap \mathcal{Z} = \emptyset$, along with evidence (observations) $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$, **Probabilistic Inference** is the process of computing the **posterior probability**: $\Pr(V_{\mathcal{Y}} | V_{\mathcal{Z}})$ for all assignments $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$.

The probabilities before any evidence is applied are called **prior probabilities**.

When models that extend beyond probability theory are used, the analog to probabilistic inference is **uncertainty inference**.

3.2 Graphical Models of Probability Distributions

Bayesian networks were first introduced by Judea Pearl in [51]. Detailed descriptions of Markov and Bayesian networks can be found in [40].

3.2.1 Markov Networks

One way large probability distributions can be denoted is as the product of belief functions over relatively small numbers of variables. A belief function over a small number of variables

is called a Markov network factor, and is simply a function that returns nonnegative weights as is defined below:

Definition 3.4. A **Markov network factor (MNF)** [40, pg. 106] F is an array of nonnegative entries indexed by a small subset of variables denoted by $\text{Var}(F)$. The entries of F **do not need to be normalized to sum to 1**. Given an arbitrary assignment $V_F \in \text{Val}(\text{Var}(F))$, $F[V_F]$ will denote the entry of F that is indexed by V_F.

If $F[V_F] = 0$ for all $V_F \in \text{Val}(\text{Var}(F))$, then F is the "zero factor over $\text{Var}(F)$" and is denoted by $\mathbf{0}_{\text{Var}(F)}$.

If $F[V_F] = 1$ for all $V_F \in \text{Val}(\text{Var}(F))$, then F is the "identity factor over $\text{Var}(F)$" and is denoted by $\mathbf{1}_{\text{Var}(F)}$.

For each variable $x \in \text{Var}(F)$, it is said that F "covers" the variable x.

A MNF F is effectively a function that assigns a weight to each possible assignment $V_F \in \text{Val}(\text{Var}(F))$. The same operations of marginalization and conditioning apply to MNFs just as they apply to probability distributions, however in the case of conditioning a MNF, the values **are not normalized to sum to 1**.

In other publications such as [46], MNFs are referred to as "clique potentials".

Definition 3.5. Given a MNF F, and a set of variables $\mathcal{Z} \supseteq \text{Var}(F)$, the **vacuous extension** of F to the variables from $\mathcal{Z}$, denoted by $F[\mathcal{Z}]$, is defined by:

$\text{Var}(F[\mathcal{Z}]) = \mathcal{Z}$

For an arbitrary $V \in \text{Val}(\mathcal{Z})$, $F[\mathcal{Z}][V] = F[V[\text{Var}(F)]]$

The vacuous extension simply expands F to cover additional variables with no additional information.

Definition 3.6. Given a MNF F, and a set of variables $\mathcal{Z}$, the **marginalization** of F to the variables from $\text{Var}(F) \cap \mathcal{Z}$, denoted by $\mathbf{marg}(F|\mathcal{Z})$, is defined by:

$\mathbf{marg}(F|\mathcal{Z})$ is a MNF that covers variables $\text{Var}(\mathbf{marg}(F|\mathcal{Z})) = \text{Var}(F) \cap \mathcal{Z}$.

For an arbitrary $V \in \text{Val}(\text{Var}(F) \cap \mathcal{Z})$, $\mathbf{marg}(F|\mathcal{Z})[V] = \sum_{V' \in \text{Val}(\text{Var}(F) \setminus \mathcal{Z})} F[\langle V, V' \rangle]$

Definition 3.7. Given a MNF F, and a set of variables $\mathcal{Z}$ with assignment $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$, the **conditioning** of F to the assignment $V_{\mathcal{Z}}$, denoted by $\textbf{cond}(F|V_{\mathcal{Z}})$, is defined by:

$\textbf{cond}(F|V_{\mathcal{Z}})$ is a MNF that covers variables $\text{Var}(\textbf{cond}(F|V_{\mathcal{Z}})) = \text{Var}(F) \setminus \mathcal{Z}$.

For an arbitrary $V \in \text{Val}(\text{Var}(F) \setminus \mathcal{Z})$, $\textbf{cond}(F|V_{\mathcal{Z}})[V] = F[\langle V, V_{\mathcal{Z}}[\text{Var}(F) \cap \mathcal{Z}]\rangle]$

Definition 3.8. Given a MNF F with at least 1 nonzero entry, $\text{norm}(F)$ will denote a normalized version of F. For each $V \in \text{Val}(\text{Var}(F))$, $\text{norm}(F)[V] = \frac{1}{K}F[V]$ where $K = \sum_{V' \in \text{Val}(\text{Var}(F))} F[V']$.

The multiplication of MNFs is defined below:

Definition 3.9. Given MNFs F_1 and F_2, their **product** $F = F_1 \times F_2$ is defined as follows: $\text{Var}(F) = \text{Var}(F_1) \cup \text{Var}(F_2)$ and for each $V \in \text{Val}(\text{Var}(F_1) \cup \text{Var}(F_2))$, $F[V] = F_1[V[\text{Var}(F_1)]] \cdot F_2[V[\text{Var}(F_2)]]$.

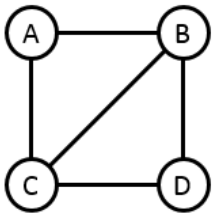
Definition 3.10. A **Markov network (MN)** is defined by a set of MNFs $\{F_1, F_2, \ldots F_k\}$. The probability distribution denoted by the Markov network is:

$$P_{MN} = \text{norm}(F_1 \times F_2 \times ... \times F_k)$$

A Markov network can also be envisioned as a simple graph where each node is uniquely labeled with a variable from $\text{Var}(P_{MN})$. Given two variables $x, y \in \text{Var}(P_{MN})$, an edge exists between x and y iff there exists F_i such that $\{x, y\} \subseteq \text{Var}(F_i)$. In other words, each F_i induces a complete clique over its variables.

An example Markov network is shown in figure 3.1. This Markov network is characterized by 3 factors which give rise to the simple graph depicted in figure 3.1. The resultant probability distribution is given in table 3.1.

The simple graph generated by an MN makes it simple to infer conditional independences between its variables. Two sets of variables are independent of each other iff there does not exist a path from one set to the other, in other words two sets of variables are independent



$\text{Val}(A) = \{a_1, a_2\}$
$\text{Val}(B) = \{b_1, b_2\}$
$\text{Val}(C) = \{c_1, c_2\}$
$\text{Val}(D) = \{d_1, d_2\}$

Factor F_1
$\text{Var}(F_1) = \{A, B\}$

A	B	$F_1[A, B]$
a_1	b_1	2
a_1	b_2	1
a_2	b_1	1
a_2	b_2	2

Factor F_2
$\text{Var}(F_2) = \{A, C\}$

A	C	$F_2[A, C]$
a_1	c_1	1
a_1	c_2	4
a_2	c_1	4
a_2	c_2	1

Factor F_3: $\text{Var}(F_3) = \{B, C, D\}$

B	$F_3[B, c_1, d_1]$	$F_3[B, c_1, d_2]$	$F_3[B, c_2, d_1]$	$F_3[B, c_2, d_2]$
b_1	1	0	0	1
b_2	0	1	1	0

Figure 3.1: An example Markov network with 3 MNFs: F_1, F_2, F_3. The resultant probability distribution is listed in table 3.1.

A	a_1				a_2			
B	b_1		b_2		b_1		b_2	
C	c_1	c_2	c_1	c_2	c_1	c_2	c_1	c_2
D								
d_1	1/15	0	0	2/15	2/15	0	0	1/15
d_2	0	4/15	1/30	0	0	1/30	4/15	0

Table 3.1: The probability distribution induced by the Markov network in figure 3.1.

of each other iff they are both on different path components. When variables are fixed to certain values (i.e. assigned evidence), the conditional MN is formed by removing all nodes associated with the conditioned variables, along with all edges connected to the removed nodes (see the left and middle networks of figure 3.2 for an example of conditioning a variable). When a variable is eliminated via marginalization, the corresponding node is removed and a new factor is formed that covers all variables that were neighbors of the eliminated variable. This new factor forms a clique between all of the former neighbors of the removed node (see the left and right networks of figure 3.2 for an example of eliminating a variable via marginalization).

The property of MNs where two sets of variables are rendered conditionally independent of each other if the conditioned variables form a "wall" between them in the MN is referred to as the Markov property. Specifically,

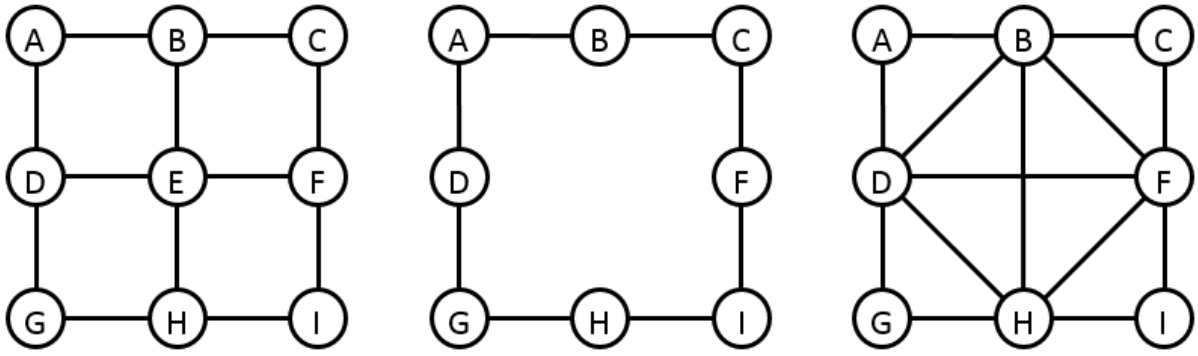


Figure 3.2: On the left is a Markov network where a two variable MNF corresponds to each edge. When variable E is assigned an evidence value, it is removed along with all edges yielding the network in the middle. When variable E is eliminated via marginalization, it is removed and clique of edges is formed among all neighbors of E. This clique denotes the 4-variable MNF that is the product of the 4 MNFs that corresponded to the edges incident with E with E subsequently summed out. This forms the Markov network on the right.

Theorem 3.11. ***The Markov property:*** *Given a Markov network, let $\mathcal{Y}_1$, $\mathcal{Y}_2$, and $\mathcal{Z}$ be disjoint sets of variables. Consider the simple graph generated by a Markov network. If there does not exist a path that connects $\mathcal{Y}_1$ and $\mathcal{Y}_2$ without passing through $\mathcal{Z}$, then $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are conditionally independent given $\mathcal{Z}$: $\mathcal{Y}_1 \perp \mathcal{Y}_2 | \mathcal{Z}$*

The Hammersley-Clifford theorem which is proven in [27, 12], proves that a probability model where the variables are nodes on an undirected graph satisfies the Markov property if *and only if*, it can be denoted by a normalized product of MNFs where each MNF corresponds to a clique in the graph. A product of MNFs, while referred to as a Markov network in this thesis, is traditionally referred to as a "Gibbs distribution" [40, pg. 108] or as a "Gibbsian ensemble" [27].

3.2.2 Bayesian networks

Given a set of random variables $\{x_1, x_2, \ldots, x_n\}$, it is valid to envision the variables being decided/instantiated in a sequential manner where x_i is decided after $x_1, x_2, \ldots, x_{i-1}$ has been decided. In reality, the conditional probability distribution for x_i may be dependent

only on a small subset of $\{x_1, x_2, \ldots, x_{i-1}\}$. This gives rise to the concept of a Bayesian network.

Definition 3.12. A **Bayesian network (BN)** consists of a directed acyclic graph (DAG) where each node corresponds to a random variable. For each variable x, $\text{Pa}(x)$ will denote the parents of x. In the context of the BN, x is only decided after its parents have all been decided. The probability distribution that decides x is extracted from a **conditional probability table (CPT)** where a separate probability distribution over x is indexed by each assignment to the parents of x.

For each variable x_i, the CPT associated with x_i will be denoted by $\text{CPT}(x_i)$. $\text{CPT}(x_i)$ is treated as an MNF over the variables $\{x_i\} \cup \text{Pa}(x_i)$. The total probability distribution denoted by the Bayesian network is:

$$P_{BN} = \text{CPT}(x_1) \times \text{CPT}(x_2) \times \cdots \times \text{CPT}(x_n)$$

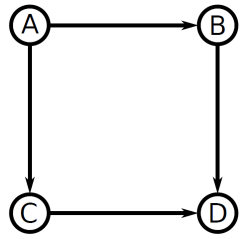
Note that no normalization is required.

For an arbitrary variable x from a BN, let $\mathcal{Y}$ denote all random variables that are not immediate parents of x or descendants of x (a descendant of x is a variable that can be reached via a directed path from x). It is then the case that $x \perp \mathcal{Y} | \text{Pa}(x)$ [40, pg. 62].

An example Bayesian network is shown in figure 3.3.

A	a_1				a_2			
B	b_1		b_2		b_1		b_2	
C	c_1	c_2	c_1	c_2	c_1	c_2	c_1	c_2
D								
d_1	0.0000	0.0000	0.0000	0.1340	0.0165	0.0000	0.0000	0.1485
d_2	0.0000	0.5360	0.0000	0.0000	0.0000	0.0165	0.1485	0.0000

Table 3.2: The probability distribution induced by the Bayesian network in figure 3.3.



$\text{Val}(A) = \{a_1, a_2\}$
$\text{Val}(B) = \{b_1, b_2\}$
$\text{Val}(C) = \{c_1, c_2\}$
$\text{Val}(D) = \{d_1, d_2\}$

$\Pr(A)$

$\Pr(a_1)$	$\Pr(a_2)$
0.67	0.33

$\Pr(B|A)$

A	$\Pr(b_1\|A)$	$\Pr(b_2\|A)$
a_1	0.8	0.2
a_2	0.1	0.9

$\Pr(C|A)$

A	$\Pr(c_1\|A)$	$\Pr(c_2\|A)$
a_1	0.0	1.0
a_2	0.5	0.5

$\Pr(D|B,C)$

C	c_1		c_2	
D	d_1	d_2	d_1	d_2
B	$\Pr(D\|B,c_1)$		$\Pr(D\|B,c_2)$	
b_1	1	0	0	1
b_2	0	1	1	0

Figure 3.3: An example Bayesian network with 4 variables: A, B, C, D. The resultant probability distribution is listed in table 3.2.

3.2.3 Variable Elimination

This section will review the well known approach to probabilistic inference on Markov networks known as variable elimination. Variable elimination (also referred to as bucket elimination or fusion) has been discussed in [38, 26, 39]. Here, for the purposes of clarity, we will give the algorithm for variable elimination. Variable elimination carries out inference by first applying the evidence $V_{\mathcal{Z}}$ to each DSF individually. Next, each non-query variable is eliminated by multiplying together all DSFs that contain the variable and then marginalizing out said variable. The DSFs that contained the eliminated variable are removed and replaced with the resultant DSF. Lastly, all remaining DSFs are multiplied together and the resultant single DSF is normalized to yield the final DS model that is restricted to the query variables and conditioned by the evidence. The variable elimination process is described in detail below:

Variable Elimination

Input: A MN as a set of factors $\{F_1, F_2, \ldots, F_k\}$; query variables $\mathcal{Y}$; evidence variables $\mathcal{Z}$ ($\mathcal{Y} \cap \mathcal{Z} = \emptyset$); and evidence $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$

for Each $i = 1, 2, \ldots, k$ **do**

// Condition each DSF individually.

$F_i = \mathbf{cond}\,(F_i|V_{\mathcal{Z}})$

end for

// Eliminate the remaining non-query variables.

$\mathcal{W} = \mathcal{X} \setminus (\mathcal{Y} \cup \mathcal{Z})$

$\mathcal{G} = \{F_1, F_2, \ldots, F_k\}$

for Each $x \in \mathcal{W}$ **do**

$F_{\text{prod}} = \mathbf{1}_{\emptyset}$

for Each $F \in \mathcal{G}$ **do**

if $x \in \text{Var}(F)$ **then**

$F_{\text{prod}} = F_{\text{prod}} \times F$

Remove F from $\mathcal{G}$

end if

end for

// Marginalize out x:

$F_{\text{prod}} = \mathbf{marg}\,(F_{\text{prod}}|\text{Var}(F_{\text{prod}}) \setminus x)$

Add F_{prod} to $\mathcal{G}$

end for

// Multiply together all remaining DSFs.

$F_{\text{total}} = \mathbf{1}_{\emptyset}$

for Each $F \in \mathcal{G}$ **do**

$F_{\text{total}} = F_{\text{total}} \times F$

end for

$D_{\text{total}} = \text{norm}(F_{\text{total}})$

return: D_{total}

3.3 Other Uncertainty Metrics

This section will describe various uncertainty metrics that will be utilized throughout this thesis. Complex applications of these uncertainty models such as marginalization, conditioning, and their use in graphical models such as in Markov networks or Bayesian networks, is discussed in chapter 5.

Figure 3.4 depicts a schematic of how the various uncertainty models that will be utilized in this thesis can be derived from basic probability distributions. The first branch, "fuzzification", chooses how uncertainty is incorporated into a probability distribution. Every non-probabilistic uncertainty metric considered in this thesis is based on probability distributions with uncertainty about the probability values themselves. Putting decisive hard restrictions on the probability values yields structures such as "credal sets" (see section 3.3.2), while softer restrictions on the probabilities yields "fuzzy probabilities" (see section 3.3.1). The section branch, "complexity", chooses the complexity of the model, with Dempster-Shafer models (see section 3.3.4) being more complicated than probability interval distributions (see section 3.3.3).

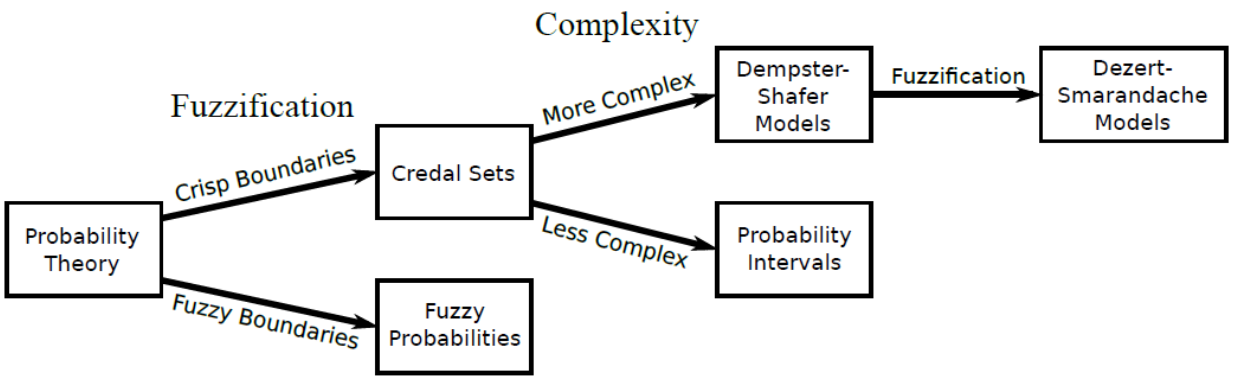


Figure 3.4: The non-probabilistic uncertainty metrics that will be used in this thesis can be derived from probability theory by incorporating different amounts of fuzziness and complexity.

3.3.1 Fuzzy Probability Distributions

When uncertainty is present in probability values, one alternative is to use "fuzzy probabilities" in place of the standard "point probabilities". The approach to fuzzy probabilities that will be used in this thesis is from [11, 53]. A fuzzy probability consists of a center value c that acts as a normal probability, and a lower and upper limit l and u that contains the center value: $l \leq c \leq u$. The interval formed by these limits is not subject to the same requirements as the probability intervals [17]. The interval formed by the lower and upper limit is not engineered to be tight; the lower bound may be less than 0, and the upper bound may be greater than 1. The relaxation of the tightness conditions on these bounds not only improves computational complexity, but is also important given the fuzzy numbers [53] which have a triangular "membership function", with a membership of 1 at the center value, that linearly decreases to 0 at both the lower and upper bounds. A lower bound of $-\infty$ creates a membership function that is 1 for all values less than the center value; and an upper bound of $+\infty$ creates a membership function that is 1 for all values greater than the center value. Lower and upper bounds that fall outside of $[0, 1]$ may not make sense from the perspective of probabilities, but they help shape the membership function of the fuzzy number inside of $[0, 1]$.

A fuzzy number is denoted by (l, c, u) where l is the lower limit, c is the center value, and u is the upper limit. The membership function $\psi : \mathbb{R} \to [0, 1]$ is defined by:

$$\psi(x) = \begin{cases} 0 & (x \leq l) \\ \frac{x-l}{c-l} & (l < x < c) \\ 1 & (x = c) \\ \frac{u-x}{u-c} & (c < x < u) \\ 0 & (u \leq x) \end{cases}$$

A non-fuzzy number (a point number) c is denoted by (c, c, c).

Definition 3.13. A **fuzzy probability distribution** S over the domain $\text{Val}(S) = \{1, 2, \ldots, M\}$ is a set of fuzzy numbers $(l_1, c_1, u_1), (l_2, c_2, u_2), ..., (l_M, c_M, u_M)$ where (l_j, c_j, u_j) is the fuzzy probability that outcome j will occur. It is required that $c_1, c_2, ..., c_M$ form a probability distribution: $\sum_{j=1}^{M} c_j = 1$.

3.3.2 Credal Sets

In the literature, [13, 15, 16, 32, 33, 34, 81], convex sets of probability distributions are referred to as "credal sets".

Definition 3.14. A **credal set**, is a convex set of probability distributions. All probability distributions in a credal set cover the same variables and have the same domain.

Specific "subtypes" of credal sets that will be the focus of investigation include "probability interval distributions" and "Dempster-Shafer models". In [33], credal sets are denoted by listing their "extreme points". The extreme points are points that belong to the credal set, but are not a convex combination of other points in the credal set [33]. Each subtype of credal set however has a more compact style of representation that comes with the restriction that there are some credal sets that cannot be represented by the current subtype.

A credal set subtype is considered to be "non-trivial" if and only if it denotes a set of probability distributions as opposed to a single probability distribution.

The following notation will be used with respect to credal sets:

- Given a single probability distribution Pr, the set {Pr} will be denoted using simply Pr.
- Given a credal set S,
  - The set of all probability distributions contained by S is denoted by simply S.
  - $\text{Var}(S)$ is the set of variables covered by each probability distribution from S.
  - $\text{Val}(S)$ denotes $\text{Val}(\text{Var}(S))$.

- Set(S) denotes Set(Var(S)).

- Given an arbitrary condition C,
  - $\Pr_L(C) = \min(\Pr(C))$ denotes the smallest probability that C is satisfied.
  - $\Pr_U(C) = \max(\Pr(C))$ denotes the largest probability that C is satisfied.

In the following two sections, two specific subtypes of credal sets will be described: "probability interval distributions" and "Dempster-Shafer" models.

3.3.3 Probability Interval Distributions

The use of probability intervals as opposed to point probabilities is discussed in [17, 25] and [68, section 4].

Definition 3.15. A **probability interval distribution** S over the domain $\text{Val}(S) = \{1, 2, \ldots, M\}$ is a set of closed intervals $[l_1, u_1], [l_2, u_2], \ldots, [l_M, u_M]$. A probability distribution $p_1, p_2, \ldots, p_M$ is contained by S if and only if $\forall j = 1, 2, \ldots, M : l_j \leq p_j \leq u_j$. In addition, the lower and upper bounds of the intervals must satisfy the following properties:

$$\begin{aligned}
\text{(The intervals must be subsets of } [0,1]) \quad & \forall j = 1, 2, \ldots, M : \ 0 \leq l_j \leq u_j \leq 1 \\
\text{(At least one probability distribution is contained)} \quad & \sum_{j=1}^{M} l_j \leq 1 \leq \sum_{j=1}^{M} u_j \\
\text{(All bounds are reachable)} \quad & \forall j' = 1, 2, \ldots, M : \ l_{j'} \geq 1 - \sum_{j : j \neq j'} u_j \\
& \forall j' = 1, 2, \ldots, M : \ u_{j'} \leq 1 - \sum_{j : j \neq j'} l_j
\end{aligned}$$

An important restriction on the bounds of the probability intervals, is that for any bound, the bound can be reached by at least one probability distribution contained by S. Let $p_1, p_2, \ldots, p_M$ be an arbitrary probability distribution contained by S. Consider $p_{j'}$. Aside from the lower bound of $l_{j'}$, $p_{j'}$ is also limited by the bounds placed on the other probabilities

since $p_{j'} = 1 - \sum_{j:j\neq j'} p_j$. Setting all other probabilities to their maximum values creates another lower bound for $p_{j'}$: $1 - \sum_{j:j\neq j'} u_j$. For $p_{j'}$ to attain the value $l_{j'}$, it must be the case that $l_{j'} \geq 1 - \sum_{j:j\neq j'} u_j$. A similar argument provides a restriction on the upper bound of $p_{j'}$.

3.3.4 Dempster-Shafer Models

A description of Dempster-Shafer theory can be found in [37, chapter 5] and [67, 71, 76, 74].

Definition 3.16. A **Dempster-Shafer model** S over the domain $\mathrm{Val}(S) = \{1, 2, \ldots, M\}$ is described by a "mass function" $m(|S) : \mathrm{Set}(S) \to [0, 1]$ (recall that $\mathrm{Set}(S) = 2^{\mathrm{Val}(S)} \setminus \{\emptyset\}$). It must be the case that:

$$\sum_{J\in\mathrm{Set}(S)} m(J|S) = 1$$

A probability distribution $p_1, p_2, \ldots, p_M$ is contained by S if and only if

$$\forall J' \subseteq \{1, 2, \ldots, M\} : \sum_{(J\subseteq J')\wedge(J\neq\emptyset)} m(J|S) \leq \sum_{j\in J'} p_j \leq \sum_{(J\cap J'\neq\emptyset)\wedge(J\neq\emptyset)} m(J|S)$$

In other words, the probability of the outcome j being a member of J' is bounded from below by the "belief":

$$\mathrm{Bel}(J'|S) = \sum_{J\subseteq J'\wedge J\neq\emptyset} m(J|S)$$

and from above by the "plausibility":

$$\mathrm{Pl}(J'|S) = \sum_{J\cap J'\neq\emptyset\wedge J\neq\emptyset} m(J|S)$$

The mass assigned to set J', $m(J'|S)$, can "slosh" around freely within set J'. The "belief" $\mathrm{Bel}(J'|S)$ is the mass that is confined to set J', while the "plausibility" $\mathrm{Pl}(J'|S)$ is the maximum mass that can accumulate in set J'.

Any probability distribution contained by S can be generated in the following manner:

For each $J \in \text{Set}(S)$, the weight contained by $m(J|S)$ is partitioned between the elements of J. Every and only the probability distributions contained by S can be formed from this process.

Dempster-Shafer models have a greater expressive power than probability intervals. Every probability interval distribution has an equivalent Dempster-Shafer model, but only a small fraction of Dempster-Shafer models have an equivalent probability interval distribution.

In a manner similar to the use of probability intervals, a Dempster-Shafer model can be completely characterized by the lower bound "belief function" $\text{Bel}(|S) : \text{Set}(S) \rightarrow [0, 1]$. The belief function must satisfy the following properties:

(The belief/lower bound must be contained by $[0, 1]$)

$\forall J \in \text{Set}(S) : 0 \leq \text{Bel}(J|S) \leq 1$

(The lower bounds must respect the union of disjoint sets)

$\forall J_1, J_2 \in \text{Set}(S) : J_1 \cap J_2 = \emptyset \implies \text{Bel}(J_1 \cup J_2|S) \geq \text{Bel}(J_1|S) + \text{Bel}(J_2|S)$

(The lower bound must be 1 for the entire domain)

$\text{Bel}(\text{Val}(S)|S) = 1$

A Dempster-Shafer model can also be completely characterized by the upper bound "plausibility function" $\text{Pl} : \text{Set}(S) \rightarrow [0, 1]$. The plausibility function must satisfy the

following properties:

(The plausibility/upper bound must be contained by $[0, 1]$)

$$\forall J \in \text{Set}(S) : 0 \leq \text{Pl}(J|S) \leq 1$$

(The upper bounds must respect the union of disjoint sets)

$$\forall J_1, J_2 \in \text{Set}(S) : J_1 \cap J_2 = \emptyset \implies \text{Pl}(J_1 \cup J_2|S) \leq \text{Pl}(J_1|S) + \text{Pl}(J_2|S)$$

(The upper bound must be 1 for the entire domain)

$$\text{Pl}(\text{Val}(S)|S) = 1$$

Given a valid belief/lower bound function or a valid plausibilty/upper bound function, the mass function can be computed via the inclusion/exclusion principle [77, pg. 4]:

$$\forall J' \in \text{Set}(S) : m(J'|S) = \sum_{J \subseteq J' \wedge J \neq \emptyset} (-1)^{|J'|+|J|} \text{Bel}(J|S)$$

$$\forall J' \in \text{Set}(S) : m(J'|S) = \sum_{J \supseteq (\text{Val}(S) \setminus J') \wedge J \neq \emptyset} (-1)^{1+|\text{Val}(S)|+|J'|+|J|} \text{Pl}(J|S)$$

It is also important to not that for most sets J from $\text{Set}(S)$, that the mass $m(J|S)$ assigned to S is 0. Only a small subset of sets from $\text{Set}(S)$ are assigned non-zero masses, and this subset is referred to as the set of "focal elements" and will be denoted by $\mathcal{E}(S)$. $m(J) > 0$ only if $J \in \mathcal{E}$. It may still be the case that a focal element is assigned a 0 mass however.

Given a specific Dempster-Shafer model S, model S can described by the following **notation**: given the set of focal elements $\mathcal{E}(S) = \{J_1, J_2, ..., J_k\}$, the model is denoted by the list of focal element/weight pairs: $\langle J_1, m(J_1|S)\rangle; \langle J_2, m(J_2|S)\rangle; ...\langle J_k, m(J_k|S)\rangle$

An important operation involving Dempster-Shafer models is Dempster's rule of combination [71]. Given two Dempster-Shafer models S_1 and S_2 over the same domain $\text{Val}(S_1) = \text{Val}(S_2) = \{1, 2, \ldots, M\}$, a third Dempster-Shafer model $S_{1,2}$ over the same domain $\text{Val}(S_{1,2}) =$

$\{1, 2, \ldots, M\}$ can be formed via Dempster's rule of combination.

The focal elements of $S_{1,2}$ are: $\mathcal{E}(S_{1,2}) = \{J_1 \cap J_2 | J_1 \in \mathcal{E}(S_1) \wedge J_2 \in \mathcal{E}(S_2) \wedge J_1 \cap J_2 \neq \emptyset\}$

The mass of focal element $J \in \mathcal{E}(S_{1,2})$ is

$$m(J|S_{1,2}) = \frac{1}{K} \sum_{J_1 \in \mathcal{E}(S_1) \wedge J_2 \in \mathcal{E}(S_2) \wedge J_1 \cap J_2 = J} m(J_1|S_1) m(J_2|S_2)$$

where $K = \sum_{J_1 \in \mathcal{E}(S_1) \wedge J_2 \in \mathcal{E}(S_2) \wedge J_1 \cap J_2 \neq \emptyset} m(J_1|S_1) m_2(J_2|S_2)$ is a normalization constant that ensures that the masses all sum to 1. $1 - K = \sum_{J_1 \in \mathcal{E}(S_1) \wedge J_2 \in \mathcal{E}(S_2) \wedge J_1 \cap J_2 = \emptyset} m_1(J_1|S_1) m_2(J_2|S_2)$ is referred to as the "conflict" between S_1 and S_2.

3.3.5 Dezert-Smarandache Models

Dezert-Smarandache (DSm) theory [19, 62] is a generalization of DS theory. DSm generalizes DS theory by using focal elements that are not simply sets of possible outcomes, but also intersections of possible outcomes. The realization of DSm in this thesis will continue to use Dempster's rule of combination without any proportional conflict redistribution, unlike [19, 62].

In essence, the DSm approach allows for overlap between what otherwise would be distinct outcomes. Each possible outcome acts as a "set". The set of all sets that can be built using the set operations of "union" ($\cup$) and "intersection" ($\cap$), starting with the outcomes is referred to as the "Dedekind lattice". Non-empty elements of the Dedekind lattice are what probability values are assigned to.

Definition 3.17. Given a domain $\text{Val}(S) = \{1, 2, ..., M\}$, the "**Dedekind lattice**" [19, 62] is a set $D^{\text{Val}(S)}$ of expressions that are formed in the following manner:

- Each element of $\text{Val}(S)$ "denotes a set" and is treated as an expression from $D^{\text{Val}(S)}$.
- Given any two expressions $A, B \in D^{\text{Val}(S)}$, then $A \cup B \in D^{\text{Val}(S)}$ and $A \cap B \in D^{\text{Val}(S)}$.

- Any two expressions from $D^{\mathrm{Val}(S)}$ that describe equivalent sets do not count as separate elements of $D^{\mathrm{Val}(S)}$.

- Any expression that cannot be formed from the first two requirements is not a member of $D^{\mathrm{Val}(S)}$.

In this thesis, the empty set $\emptyset$ is not a member of $D^{\mathrm{Val}(S)}$.

As an example of the Dedekind lattice, if $\mathrm{Val}(S) = \{1, 2, 3\}$, then $D^{\mathrm{Val}(S)} = \{1, 2, 3, 1 \cup 2, 1 \cup 3, 2 \cup 3, 1 \cap 2, 1 \cap 3, 2 \cap 3, 1 \cup (2 \cap 3), 2 \cup (1 \cap 3), 3 \cup (1 \cap 2), 1 \cap (2 \cup 3), 2 \cap (1 \cup 3), 3 \cap (1 \cup 2), 1 \cup 2 \cup 3, 1 \cap 2 \cap 3\}$

A Dezert-Smarandache model is a generalization of the Dempster-Shafer model from $\mathrm{Set}(S) = 2^{\mathrm{Val}(S)} \setminus \{\emptyset\}$ to $D^{\mathrm{Val}(S)}$. Any expression J from $D^{\mathrm{Val}(S)}$ that contains only the operator $\cup$ is the direct analog of a set from $\mathrm{Set}(S)$ that contains only the elements from $\mathrm{Val}(S)$ that appear in expression J. In essence, if $\mathrm{Val}(S) = \{1, 2, 3\}$, then $\mathrm{Set}(S) = \{1, 2, 3, 1 \cup 2, 1 \cup 3, 2 \cup 3, 1 \cup 2 \cup 3\}$.

Definition 3.18. A **Dezert-Smarandache model** S over the domain $\mathrm{Val}(S) = \{1, 2, \ldots, M\}$ is described by a "mass function" $m(|S) : D^{\mathrm{Val}(S)} \to [0, 1]$. It must be the case that:

$$\sum_{J \in D^{\mathrm{Val}(S)}} m(J) = 1$$

All notation associated with Dempster-Shafer models equally applies to Dezert-Smarandache models.

3.4 Risk Analysis

"Risk" is any quantity that quantifies an undesirable outcome, such as the probability of the undesirable outcome occurring, the expected cost of the undesirable outcome, a heuristic combination of the probability and cost, etc. Examples of risk values related to biometric

systems are the "false accept rate" (FAR), and the "false reject rate" (FRR). The FAR is the probability that the system gives a positive response given that the correct response is negative, and the FRR is the probability that the system gives a negative response given that the correct response is positive.

3.4.1 Risk fusion

When multiple risk values are obtained, a common practice is to "fuse" these risk values into a single quantity that a human operator, or decision support algorithm can use to make a decision. The process of "fusing" risks is a heuristic process that is described further in chapter 6. The basics of information fusion are provided below:

Let risk values R_1 and R_2 be real numbers, and let 0 be the "decision threshold", which is the limit between "high risk" and "low risk". Positive values are high risk, and negative values are low risk. Fusing R_1 and R_2 into a final risk value $R_{\text{final}}(R_1, R_2)$ must satisfy the following properties:

- $R_{\text{final}}(R_1, R_2)$ is monotone **increasing** with respect to R_1 and R_2.
- $R_1 \leq 0$ and $R_2 \leq 0$ implies that $R_{\text{final}}(R_1, R_2) \leq \min(R_1, R_2)$. When R_1 and R_2 are both low, they compound each other and the overall risk decreases.
- $R_1 \geq 0$ and $R_2 \geq 0$ implies that $R_{\text{final}}(R_1, R_2) \geq \max(R_1, R_2)$. When R_1 and R_2 are both high, they compound each other and the overall risk increases.

A very simple fusion operation is: $R_{\text{final}}(R_1, R_2) = R_1 + R_2$.

A more sophisticated fusion operator can be derived in the following manner: For each $p = 1, 2$, let $f_p(R_p) = \Pr(\text{"high-risk"}|R_p)$ (and $1 - f_p(R_p) = \Pr(\text{"low-risk"}|R_p)$). It is safe to assume that f_p is monotone increasing with respect to R_p. It is also reasonable to assume that $f_p(0) = 0.5$.

When $((R_1 > 0) \wedge (R_2 < 0)) \vee ((R_1 < 0) \wedge (R_2 > 0))$, the risks disagree with each other, and we wish to choose the risk that is least likely to be incorrect:

$$(R_1 > 0 > R_2) \wedge (1 - f_1(R_1) < f_2(R_2)) \Rightarrow (R_{\text{final}} > 0)$$
$$(R_1 > 0 > R_2) \wedge (1 - f_1(R_1) = f_2(R_2)) \Rightarrow (R_{\text{final}} = 0)$$
$$(R_1 > 0 > R_2) \wedge (1 - f_1(R_1) > f_2(R_2)) \Rightarrow (R_{\text{final}} < 0)$$
$$(R_1 < 0 < R_2) \wedge (f_1(R_1) < 1 - f_2(R_2)) \Rightarrow (R_{\text{final}} < 0)$$
$$(R_1 < 0 < R_2) \wedge (f_1(R_1) = 1 - f_2(R_2)) \Rightarrow (R_{\text{final}} = 0)$$
$$(R_1 < 0 < R_2) \wedge (f_1(R_1) > 1 - f_2(R_2)) \Rightarrow (R_{\text{final}} > 0)$$

If R_{final} obeys the above properties for each R_1 and R_2, then the risk that is more likely to be correct will be the risk that determines the sign of R_{final}.

Below, we will construct a fusion operation that obeys all of the aforementioned properties. To ease the description, let $R_3 = R_1 + R_2$ and $R_4 = R_2 - R_1$ be transformed risk values. R_3 is a measure of the "total risk value", while R_4 is a measure of the "risk difference". Note that $f_1(R_1) + f_2(R_2)$ is monotone increasing with respect to R_3 (while R_4 is held constant). Let g be a function such that if $R_3 = g(R_4)$, then $f_1(R_1) + f_2(R_2) = 1$. $R_3 = g(R_4)$ marks the boundary between low and high final risk. Since $f_1(0) = f_2(0) = 0.5$, it is the case that $g(0) = 0$.

$$R_{\text{final}}(R_1, R_2) =$$
$$\begin{cases} R_3 & ((R_3 \geq |R_4|) \vee (R_3 \leq -|R_4|)) \\ |R_4|\frac{R_3 - g(R_4)}{|R_4| - g(R_4)} & (g(R_4) \leq R_3 \leq |R_4|) \\ -|R_4|\frac{R_3 - g(R_4)}{-|R_4| - g(R_4)} & (-|R_4| \leq R_3 \leq g(R_4)) \end{cases}$$

Note that $R_3 \geq |R_4| \iff R_1, R_2 \geq 0$ and $R_3 \leq -|R_4| \iff R_1, R_2 \leq 0$. Figure 3.5

illustrates the relationship between (R_1, R_2) and (R_3, R_4), as well as the zones that form the different piecewise domains for $R_{\text{final}}(R_1, R_2)$.

This above fusion operator provides a means for fusing risk the risk values R_1 and R_2 to compute the combined risk value R_{final} .

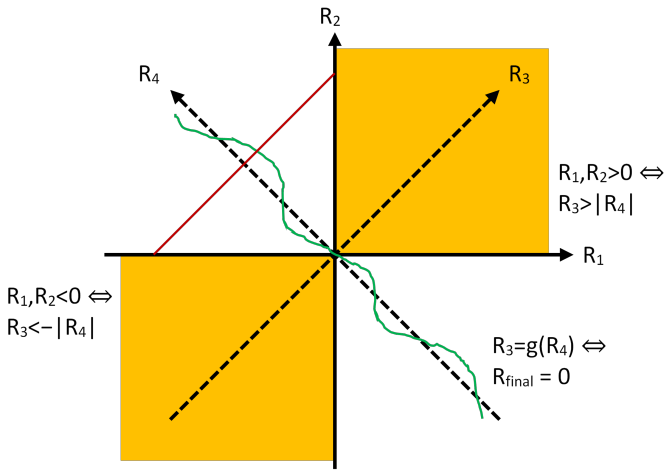


Figure 3.5: The relationship between (R_1, R_2) and (R_3, R_4). The shaded region is the domain where $R_{\text{final}}(R_1, R_2) = R_1 + R_2$. The wavy line is one possible curve for $R_3 = g(R_4)$ and is where $R_{\text{final}} = 0$. In the unshaded area, $R_{\text{final}}(R_1, R_2)$ is a linear interpolation between the wavy curve and the R_1 and R_2 axes. The line in the top-left quadrant is a line where R_4 is constant, and R_3 parameterizes a piecewise linear function from the negative R_1 axis to the wavy line; and from the wavy line to the positive R_2 axis.

As a final numerical example, imagine that $R_1 = -1$ and $R_2 = +3$. R_1 and R_2 disagree with each other, and so the final risk R_{final} is not simply the sum of R_1 and R_2. $R_3 = R_1 + R_2 = 2$ and $R_4 = R_2 - R_1 = 4$. Assume, as an example, that $g(4) = +1$. Since $g(4) > 0$, this indicates that the curve that separates high and low risk values is displaced in a direction that favors negative risk values. Since $R_3 > g(R_4)$, $R_{\text{final}}(-1, +3) = 4 \cdot \frac{2-1}{4-1} = +\frac{4}{3}$. In conclusion, the positive R_2 dominated the negative R_1. However, the positive $g(4)$ almost dragged the final risk factor down to being negative.

The fusion operator given above is only one of many different possible fusion operators that can be used to fuse risk values R_1 and R_2. Above, axioms were given that dictate how R_1, R_2, and R_{final} must behave and there are infinitely other fusion operators that satisfy

the axioms.

Different mechanisms for fusing strengths of belief have been researched for various applications [47, 66, 62, 73].

3.5 Information Theory

This section will give a review of material related to information theory that will be needed for chapter 7 when describing the computationally efficient approach to decomposing a probability distribution into layers. A review of information theory can be found in [40, pg. 1137–1143].

Given a set of random variables $\mathcal{Y} \subseteq \mathcal{X}$, the "Shannon entropy" of $\mathcal{Y}$ is:

$$\mathbf{H}(\mathcal{Y}) = -\sum_{V \in \text{Val}(\mathcal{Y})} \Pr(V) \log_2(\Pr(V))$$

It can proven via perturbative approaches that the Shannon entropy is maximized when for all $V \in \text{Val}(\mathcal{Y})$, that $\Pr(V) = \frac{1}{|\text{Val}(\mathcal{Y})|}$.

Given disjoint sets of random variables $\mathcal{Y}, \mathcal{Z} \subseteq \mathcal{X}$, the joint Shannon entropy over $\mathcal{Y} \cup \mathcal{Z}$ is:

$$\begin{aligned}
\mathbf{H}(\mathcal{Y} \cup \mathcal{Z}) &= -\sum_{V \in \text{Val}(\mathcal{Y} \cup \mathcal{Z})} \Pr(V) \log_2(\Pr(V)) \\
&= -\sum_{V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})} \sum_{V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})} \Pr(\langle V_{\mathcal{Y}}, V_{\mathcal{Z}} \rangle) \log_2(\Pr(\langle V_{\mathcal{Y}}, V_{\mathcal{Z}} \rangle)) \\
&= -\sum_{V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})} \Pr(V_{\mathcal{Z}}) \sum_{V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})} \Pr(V_{\mathcal{Y}} | V_{\mathcal{Z}}) (\log_2(\Pr(V_{\mathcal{Z}})) + \log_2(\Pr(V_{\mathcal{Y}} | V_{\mathcal{Z}}))) \\
&= -\sum_{V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})} \left(\Pr(V_{\mathcal{Z}}) \log_2(\Pr(V_{\mathcal{Z}})) + \Pr(V_{\mathcal{Z}}) \sum_{V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})} \Pr(V_{\mathcal{Y}} | V_{\mathcal{Z}}) \log_2(\Pr(V_{\mathcal{Y}} | V_{\mathcal{Z}})) \right) \\
&= \mathbf{H}(\mathcal{Z}) + \sum_{V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})} \Pr(V_{\mathcal{Z}}) \left(-\sum_{V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})} \Pr(V_{\mathcal{Y}} | V_{\mathcal{Z}}) \log_2(\Pr(V_{\mathcal{Y}} | V_{\mathcal{Z}})) \right)
\end{aligned}$$

The conditional entropy of $\mathcal{Y}$ when $\mathcal{Z} = V_{\mathcal{Z}}$, denoted by $\mathbf{H}(\mathcal{Y}|V_{\mathcal{Z}})$, is the entropy of $\mathcal{Y}$ after the variables from $\mathcal{Z}$ have been fixed to $V_{\mathcal{Z}}$:

$$\mathbf{H}(\mathcal{Y}|V_{\mathcal{Z}}) = -\sum_{V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})} \Pr(V_{\mathcal{Y}}|V_{\mathcal{Z}}) \log_2(\Pr(V_{\mathcal{Y}}|V_{\mathcal{Z}}))$$

hence:

$$\mathbf{H}(\mathcal{Y} \cup \mathcal{Z}) = \mathbf{H}(\mathcal{Z}) + \sum_{V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})} \Pr(V_{\mathcal{Z}}) \mathbf{H}(\mathcal{Y}|V_{\mathcal{Z}})$$

The conditional entropy of $\mathcal{Y}$ given $\mathcal{Z}$ without any assignment to the variables from $\mathcal{Z}$ is:

$$\mathbf{H}(\mathcal{Y}|\mathcal{Z}) = \sum_{V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})} \Pr(V_{\mathcal{Z}}) \mathbf{H}(\mathcal{Y}|V_{\mathcal{Z}})$$

so:

$$\mathbf{H}(\mathcal{Y} \cup \mathcal{Z}) = \mathbf{H}(\mathcal{Z}) + \mathbf{H}(\mathcal{Y}|\mathcal{Z})$$

If variable sets $\mathcal{Y}$ and $\mathcal{Z}$ are independent, then $\mathbf{H}(\mathcal{Y}|V_{\mathcal{Z}}) = \mathbf{H}(\mathcal{Y})$ for all $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$ so hence $\mathbf{H}(\mathcal{Y}|\mathcal{Z}) = \mathbf{H}(\mathcal{Y})$ and $\mathbf{H}(\mathcal{Y} \cup \mathcal{Z}) = \mathbf{H}(\mathcal{Y}) + \mathbf{H}(\mathcal{Z})$.

Lastly, it can proven via perturbative approaches that the joint entropy of $\mathcal{Y} \cup \mathcal{Z}$ is maximized when $\mathcal{Y}$ and $\mathcal{Z}$ are independent: $\mathbf{H}(\mathcal{Y} \cup \mathcal{Z}) \leq \mathbf{H}(\mathcal{Y}) + \mathbf{H}(\mathcal{Z})$. The difference in entropy, $\mathbf{I}(\mathcal{Y} : \mathcal{Z}) = \mathbf{H}(\mathcal{Y}) + \mathbf{H}(\mathcal{Z}) - \mathbf{H}(\mathcal{Y} \cup \mathcal{Z})$, is the "mutual information" that exists between $\mathcal{Y}$ and $\mathcal{Z}$. The mutual information is a measure of how closely bound two variable sets are, and will be used to decide whether $\mathcal{Y}$ and $\mathcal{Z}$ can be approximated as being independent or not.

An important metric that is commonly used to compare probability distributions is the Kullback-Liebler divergence (KL-divergence), also referred to as "relative entropy". Given two probability distributions $\Pr_1$ and $\Pr_2$ over variable set $\mathcal{Y}$, the KL-divergence between $\Pr_1$ and $\Pr_2$ is:

$$\text{KL}(\Pr_1, \Pr_2) = \sum_{V \in \text{Val}(\mathcal{Y})} \Pr_1(V) \log_2 \left(\frac{\Pr_1(V)}{\Pr_2(V)} \right)$$

It can be shown that $\mathrm{KL}(\mathrm{Pr}_1, \mathrm{Pr}_2) \geq 0$ where $\mathrm{KL}(\mathrm{Pr}_1, \mathrm{Pr}_2) = 0$ only if $\mathrm{Pr}_1 = \mathrm{Pr}_2$:

$$\begin{aligned}
\mathrm{KL}(\mathrm{Pr}_1, \mathrm{Pr}_2) &= \sum_{V \in \mathrm{Val}(\mathcal{Y})} \mathrm{Pr}_1(V) \log_2 \left(\frac{\mathrm{Pr}_1(V)}{\mathrm{Pr}_2(V)} \right) = \sum_{V \in \mathrm{Val}(\mathcal{Y})} \mathrm{Pr}_1(V) \left(-\log_2 \left(\frac{\mathrm{Pr}_2(V)}{\mathrm{Pr}_1(V)} \right) \right) \\
&\geq \sum_{V \in \mathrm{Val}(\mathcal{Y})} \mathrm{Pr}_1(V) \cdot \frac{1}{\ln(2)} \left(1 - \frac{\mathrm{Pr}_2(V)}{\mathrm{Pr}_1(V)} \right) = \frac{1}{\ln(2)} \sum_{V \in \mathrm{Val}(\mathcal{Y})} (\mathrm{Pr}_1(V) - \mathrm{Pr}_2(V)) \\
&= \frac{1}{\ln(2)} (1 - 1) = 0
\end{aligned}$$

In the above derivation, the inequality becomes equality only when $\frac{\mathrm{Pr}_2(V)}{\mathrm{Pr}_1(V)} = 1$ for all $V \in \mathrm{Val}(\mathcal{Y})$. In other words, equality holds only when $\mathrm{Pr}_2 = \mathrm{Pr}_1$.

It should also be noted that the KL-divergence is not symmetric: $\mathrm{KL}(\mathrm{Pr}_1, \mathrm{Pr}_2) \neq \mathrm{KL}(\mathrm{Pr}_2, \mathrm{Pr}_1)$ in most cases.

3.6 Conclusion

This chapter has presented the background material that is necessary to understand the content and contributions of the next four chapters. This background material consisted of a review of probability theory, graphical models of probability distributions, a survey of non-probabilistic uncertainty metrics, risk analysis, and a brief description of information theory. The next four chapters will present the primary contributions of this thesis.

Chapter 4

Modular Models

4.1 Introduction

This chapter will address the NP-hardness of probabilistic inference using Bayesian networks. The approach that will be described in this chapter will be to break a large Bayesian network into a library of more manageable "modules". The modules can be assembled into a larger network by connecting the modules through the use of interfaces that will be described in this chapter.

Other Bayesian network interfacing paradigms that have been used in other works are listed in Tables 4.1 and 4.2. A description of how the modularization approach given in this chapter differs from other approaches is also given.

4.2 NP-hardness of probabilistic inference

In [40, pg.288–290], it is shown that probabilistic inference using an arbitrary Bayesian network is NP-hard. NP means Non-deterministic Polynomial time complexity. Problems that are "NP-hard" are such that if a deterministic polynomial time algorithm exists that solves the problem, then the problem provides a means where every NP problem can be solved in polynomial time. While it is unknown whether NP-hard problems can be solved

Table 4.1: Bayesian network interfacing paradigms

SOURCE	DESCRIPTION	COMPARISON
[28]	Separate Bayesian networks are connected via "virtual links". A virtual link runs from a specific node in the source network to a specific node in the destination network. The posterior probability of the source node becomes the prior probability of the destination node.	In our proposed concept of linking Bayesian networks, the posterior probabilities stored in the intermediate nodes are used to compute the priors in the next network using a variable function instead of a direct substitution.
[35]	The output of multiple computational modules (which may not necessarily be Bayesian networks) are merged into a single output through the use of a relatively simple Bayesian network. The output of each module is a single value that is treated as an evidence value in the Bayesian network that is merging the outputs.	In our proposed approach, an entire probability distribution is returned from a network/module as opposed to a single estimated value.
[41, 55]	A large Bayesian network is simply subdivided into smaller networks. Directed edges that exist between nodes can run from one module to another.	In [40, pg.288–290], it is shown that inference using an arbitrary Bayesian network is NP-hard. This makes inference using large Bayesian networks computationally intractable. By breaking the network into sub-modules using the paradigm described in our approach, the size of each sub-network is limited and inference remains computationally tractable.

in polynomial time, if a problem is established as being NP-hard, then it is considered computationally intractable for large inputs. The NP-hardness of probabilistic inference using Bayesian networks means that using large Bayesian networks probabilistic inference is computationally intractable.

This computational intractability is addressed in this chapter by breaking the network into sub-modules. The size of each sub-network is limited and probabilistic inference becomes computationally tractable, at the expense of some accuracy of the model.

Table 4.2: Bayesian network interfacing paradigms continued

SOURCE	DESCRIPTION	COMPARISON
[8, 29, 50, 57]	"Clique trees" are used to provide structure to the process of variable elimination. Each node in the clique tree denotes a clique of nodes from the original Markov network. Each edge in the clique tree denotes the intersection between the two cliques of the connected clique tree nodes. "Messages" are passed along edges of the clique tree to enable the eventual computation of the marginal probability over each clique. This process is used in the "HUGIN" shell [8].	A clique tree has a tree structure, and if the Markov network has an intrinsically loopy structure, extra variables must be incorporated into each clique in the loop to "squish" the loop into a simple path of clique nodes. Clique trees provide structure to the variable elimination process, but they do not mitigate NP-hardness.
[42, 50, 52, 69]	A set of Markov network factors is denoted using a bipartite graph called a "factor graph", where the variables and factors form the different partitions. An iterative process referred to in [40] as "loopy belief propagation" is used to circulate "messages" between the variables and factors until convergence is achieved. Improvements to the convergence are proposed in [52]. A generalization of factor graphs that consist of multiple partitions with factors of varying complexity is used in [69].	For "loopy belief propagation", convergence is not guaranteed to occur, which implies the absence of an upper bound on the computational complexity. The modular models proposed in this chapter do not allow information to flow in closed loops which forces the rapid computation of a posterior probability distribution.

4.3 Bayesian Network Modules

A large Bayesian network will be broken into "modules" in order to restrain the computational complexity of probabilistic inference. The computational complexity is reduced for two reasons:

- The size of each module is limited, which limits the computational complexity of the probabilistic inference that occurs in each module.
- Information can only flow in one direction when two modules are linked. Moreover, information cannot return to a module via a circuit of directed links. Preventing feedback implies that the computational complexity has a linear upper bound with respect to the number of modules.

Definition 4.1. A **Bayesian network module** is a modified small-scale Bayesian network where some prior probabilities and conditional probabilities may be functions of posterior probability distributions computed in parent modules.

Figure 4.1 depicts an example of the interface/connections that exist between modules. The two example modules are named A and B. The connection between modules A and B is facilitated through the use of "intermediate nodes" that store posterior probability distributions that were computed in module A. These posterior probabilities in turn influence some of the prior probabilities and conditional probabilities that are present in module B. In this example, variables X, Y, and Z are variables from module A. The marginal posterior probability distribution of variable X is computed, as well as the posterior joint probability distribution of variables Y, Z. These posterior probability distributions are stored in the intermediate nodes. The posterior probability distribution of X is used to compute the prior probabilities of variable W in module B via some pre-specified function. The posterior joint probability distribution of variables Y, Z is used to compute the conditional probabilities of variable S in module B via another pre-specified function.

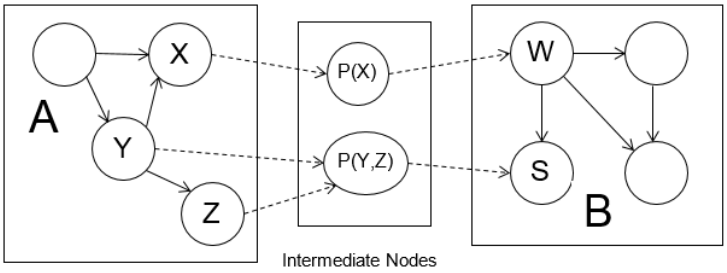


Figure 4.1: Example of a belief network interfacing between modeling modules A and B.

In addition to mitigating the computational complexity, another advantage to using modules is that modules can be shuffled around on an as needs basis to create new models, as opposed to training a new complex model from scratch.

Definition 4.2. A **Modular model** is a large-scale network of Bayesian network modules. The links between modules consist of **intermediate nodes** that store posterior probability distributions computed in the parent module to provide as input to the child module. The modular model forms a complex model of the scenario under consideration, and the Bayesian network modules are often chosen from a library of modules.

This chapter will illustrate the concept of Bayesian network modules by presenting an example where Bayesian network modules are used to model a scenario related to biometric enabled e-borders. A number of Bayesian network modules that are designed to assist in the modeling of various scenarios that may be encountered in the context of ABC (Automated Border Control [79]) machines. Each module denotes a specific aspect of an ABC machine, or an attack scenario. These modules can be linked together into a directed acyclic graph to form a larger network that can then be used to infer risks associated with a specific scenario or setup.

The set of random variables associated with each module may not necessarily be disjoint.

In Figure 4.2, the process of using a library of modules for modeling scenarios and performing probabilistic inference is depicted. Starting with a scenario that is desired to be modeled, modules are chosen from the library of modules and snapped together to form a

modular model. From this modular model, observed evidence related to a specific instance of the scenario is applied to the model in order to generate posterior probabilities related to the specific scenario instance.

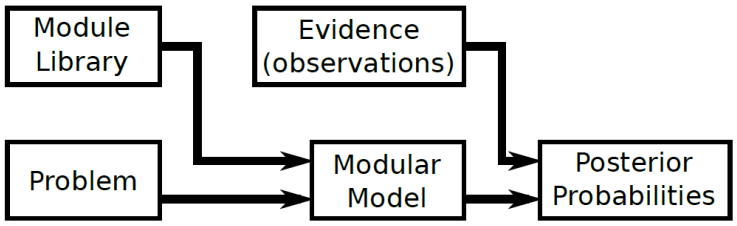


Figure 4.2: A high level depiction of using a library of modules for modeling scenarios and performing probabilistic inference.

4.4 The random variables

In the following sections, several Bayesian network modules will be presented as examples. Several modules may contain the same variables, so all random variables that are found throughout the modules are listed and described here for simplicity.

Below is a listing of the random variables used throughout the modules:

- Variable H (e-passport holder) is assigned the value h_1 (yes) if the traveler has an e-passport, and the value h_2 (no) if otherwise.

- Variable A (attack mode) is assigned the value a_1 (no) if no attack on the e-passport system is taking place; the value a_2 (passport lost) if the traveller has lost his/her e-passport; the value a_3 (stolen passport) if the traveller is using a stolen passport; the value a_4 (skimmed passport) if the traveller is using a fake passport created from skimmed data; and the value a_5 (other attack) if the attack mode is not one of the previous options.

- Variable L (e-passport reported as lost) is assigned the value l_1 (yes) if the passport's

true owner as reported to the border authorities that his/her passport has been lost (possibly stolen); and the value l_2 if otherwise.

- Variable F (fraudulently obtained passport) is assigned the value f_1 (yes) if the passport has been obtained fraudulently, and f_2 (no) if otherwise. F denotes whether or not the passport was truly obtained fraudulently, and not simply the opinion of the ABC machine.

- Variable W (e-passport watch list) is assigned the value w_1 (yes) if the traveler's passport is on a watch-list for being fraudulently obtained, and the value w_2 (no) if otherwise.

- Variable S (security features) is assigned the value s_1 (yes) if the security features (watermarks, checksums, etc.) are deemed to be valid, and is assigned the value s_2 (no) if otherwise.

- Variable C (crosscheck) is assigned the values c_1 (1^{st} attempt), c_2 (2^{nd} attempt), and c_3 (3^{rd} attempt) if the chip and optically scanned data match on the 1^{st}, 2^{nd}, and 3^{rd} attempt respectively. C is assigned c_4 (failed) if there is no match.

- Variable V (valid passport) is assigned the value v_1 (yes) if the passport is deemed to be valid by the authentication control system, and v_2 (no) if otherwise.

- Variable I (impostor) is assigned the value i_1 (yes) is the traveler is trying to impersonate another individual.

- The following variables relate to various biometric modalities:

  - Variable J (face scan) is assigned the values j_1 (1^{st} attempt), j_2 (2^{nd} attempt), and j_3 (3^{rd} attempt) if the face biometric scanner authenticates the traveler on the 1^{st}, 2^{nd}, and 3^{rd} attempt respectively. J is assigned j_4 (failed) if the face scan does not authenticate the traveler.

- Variable J' (iris scan) is assigned the values j'_1 (1^{st} attempt), j'_2 (2^{nd} attempt), and j'_3 (3^{rd} attempt) if the iris biometric scanner authenticates the traveler on the 1^{st}, 2^{nd}, and 3^{rd} attempt respectively. J' is assigned j'_4 (failed) if the iris scan does not authenticate the traveler.
- Variable J'' (fingerprint scan) is assigned the values j_1 (1^{st} attempt), j_2 (2^{nd} attempt), and j_3 (3^{rd} attempt) if the fingerprint biometric scanner authenticates the traveler on the 1^{st}, 2^{nd}, and 3^{rd} attempt respectively. J is assigned j_4 (failed) if the fingerprint scan does not authenticate the traveler.

- Variable M (manual check) is assigned the value m_1 (yes) is the traveler is redirected to a manual check by a border agent, and assigned the value m_2 (no) if otherwise.

- Variable A' (e-passport authentication) is assigned the values a'_1 (1^{st} attempt), a'_2 (2^{nd} attempt), and a'_3 (3^{rd} attempt) if authentication of the e-passport was successful on the 1^{st}, 2^{nd}, and 3^{rd} attempt respectively. A' is assigned a'_4 (failed) if authentication fails.

- Variable E (exit) is assigned the value e_1 (successful exit) if the traveler passes the border, and is assigned the value e_2 (blocked) if the traveller is stopped at the border.

- Variable W' (wait time) is assigned the value w'_1 if the traveler is takes less than 10 minutes to cross the border; the value w'_2 if the traveller takes more than 10 minutes to cross the border; and the value w'_3 if the traveler fails to cross the border.

At this point it is important to note that in the examples throughout this thesis, except for where explicitly stated otherwise, that all probabilities and belief values have been arbitrary generated for the sake of example, and are not derived from real world data. In practice, where statistical data and expert knowledge is available to the engineers of biometric enabled authentication systems, real world data will be used in these models.

4.5 E-passport Holder Network Modules

The primary objective of an E-Passport Holder network module is to model the risk/probabilities of the traveler not holding an e-passport, or of the e-passport being obtained fraudulently. Below, some example modules of varying complexity and purpose are given to illustrate a variety of choices that can be made for this particular module.

4.5.1 Simple E-passport Holder Network Module

The Simple E-passport Holder Network shown in this section is a simple network that is to be used when the E-passport holder network is of no particular importance. This network has nodes H, W, and F. The network is shown in Figure 4.3.

Example Scenarios:

- A traveller holds an e-passport, but his/her passport is also on an e-passport watch-list. Despite this, the traveller's e-passport was obtained legitimately.
- A traveller does not hold an e-passport. This may be because the traveller has never enrolled for an e-passport or has forgotten/lost his/her passport. In either case, the nonexistent passport is deemed to be not fraudulent.

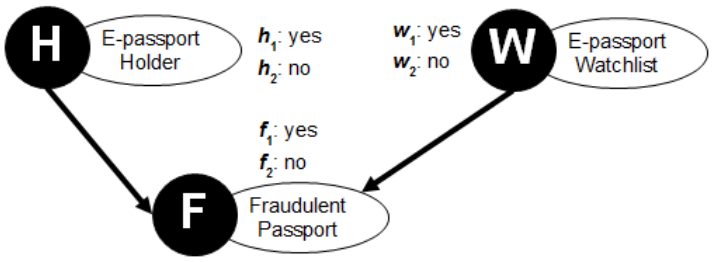


Figure 4.3: The Simple E-passport Holder Network

The probability distributions have been chosen with regard to the following assumptions:

- If the traveller does not hold an e-passport, then the (nonexistent) passport could not have been obtained fraudulently: $H = h_2 \implies F = f_2$

The probability tables for nodes H, W, and F are shown in tables 4.3, 4.4, and 4.5 respectively.

Table 4.3: The probability distribution $\Pr(H)$ associated with H in the Simple E-passport Holder Network.

$\Pr(H)$	
$\Pr(h_1)$	$\Pr(h_2)$
0.96	0.04

Table 4.4: The probability distribution $\Pr(W)$ associated with W in the Simple E-passport Holder Network.

$\Pr(W)$	
$\Pr(w_1)$	$\Pr(w_2)$
0.001	0.999

Table 4.5: The probability distributions $\Pr(F|H,W)$ associated with F in the Simple E-passport Holder Network.

$\Pr(F\|H,W)$		
H,W	$\Pr(f_1\|H,W)$	$\Pr(f_2\|H,W)$
h_1w_1	0.6	0.4
h_1w_2	0.001	0.999
h_2w_1	0	1
h_2w_2	0	1

4.5.2 E-passport Attack Network Module

The E-passport attack network module is used in place of the Simple E-passport Holder network module when a specific attack against the E-passport system is to be modeled. This network has nodes H, A, L, F, and W. The network is shown in Figure 4.4.

Example Scenarios:

- A traveller does not hold an e-passport due to it being lost. The traveller reports this loss to the authorities who then proceed to place the passport on a watchlist to keep a potential thief from using it.
- The traveller has stolen an e-passport and is trying to use it to illegally cross the border. The passport's true owner has reported the loss, placing the passport on a watchlist.

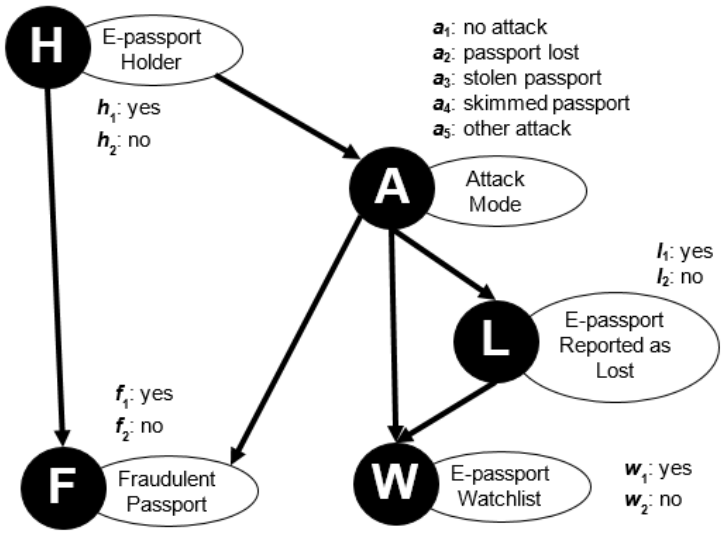


Figure 4.4: The E-passport Attack Network

The probability distributions have been chosen with regard to the following assumptions:

- If the traveller holds an e-passport, then the passport cannot be lost: $H = h_1 \implies A \neq a_2$
- If the traveller does not hold an e-passport, then the traveller is not using a stolen or forged e-passport: $H = h_2 \implies A \notin \{a_3, a_4\}$

- If the passport's true owner has not lost his/her passport, then the passport won't be reported as being lost: $A \in \{a_1, a_4\} \implies L = l_2$

- If the traveller holds an e-passport, then if no attack is taking place, or if the traveller has lost their e-passport, then the e-passport was not obtained fraudulently: $(H = h_1 \wedge A \in \{a_1, a_2\}) \implies F = f_2$

- If the traveller holds an e-passport, then if the passport is a stolen passport or a cloned passport, then the e-passport was obtained fraudulently: $(H = h_1 \wedge A \in \{a_3, a_4\}) \implies F = f_1$

- If the traveller does not hold an e-passport, then the (nonexistent) passport could not have been obtained fraudulently: $H = h_2 \implies F = f_2$

- If the passport was reported as being lost, then the passport is placed on a watchlist: $L = l_1 \implies W = w_1$

The probability values for nodes H, A, L, F, and W are shown in tables 4.6, 4.7, 4.8, 4.9, and 4.10 respectively.

Table 4.6: The probability distribution $\Pr(H)$ associated with H in the E-passport Attack Network.

$\Pr(H)$

$\Pr(h_1)$	$\Pr(h_2)$
0.96	0.04

Table 4.7: The probability distributions $\Pr(A|H)$ associated with A in the E-passport Attack Network.

$\Pr(A|H)$

H	$\Pr(a_1\|H)$	$\Pr(a_2\|H)$	$\Pr(a_3\|H)$
h_1	0.85	0	0.05
h_2	0.45	0.5	0

H	$\Pr(a_4\|H)$	$\Pr(a_5\|H)$
h_1	0.05	0.05
h_2	0	0.05

Table 4.8: The probability distributions $\Pr(L|A)$ associated with L in the E-passport Attack Network.

$\Pr(L|A)$

A	$\Pr(l_1\|A)$	$\Pr(l_2\|A)$
a_1	0	1
a_2	0.8	0.2
a_3	0.9	0.1
a_4	0	1
a_5	0.5	0.5

Table 4.9: The probability distributions $\Pr(F|H, A)$ associated with F in the E-passport Attack Network.

$\Pr(F|H, A)$

H, A	$\Pr(f_1\|H, A)$	$\Pr(f_2\|H, A)$
h_1a_1	0	1
h_1a_2	0	1
h_1a_3	1	0
h_1a_4	1	0
h_1a_5	0.5	0.5
h_2a_1	0	1
h_2a_2	0	1
h_2a_3	0	1
h_2a_4	0	1
h_2a_5	0	1

Table 4.10: The probability distributions $\Pr(W|A, L)$ associated with W in the E-passport Attack Network.

$\Pr(W|A, L)$

A, L	$P(w_1\|A, L)$	$P(w_2\|A, L)$
a_1l_1	1	0
a_1l_2	0.001	0.999
a_2l_1	1	0
a_2l_2	0.001	0.999
a_3l_1	1	0
a_3l_2	0.01	0.99
a_4l_1	1	0
a_4l_2	0.05	0.95
a_5l_1	1	0
a_5l_2	0.5	0.5

4.6 E-passport Scan Network Modules

The primary objective of the E-passport Scan Networks is to compute the risk/probability of a passport being accepted as valid. There are two variations of this network that can be chosen, depending on the network that is chosen as the parent. The E-passport Scan Network does not receive any information about the attack mode (variable A), while the the E-passport Scan Network 2 receives the posterior distribution of the attack mode.

4.6.1 E-passport Scan Network Module 1

The primary objective of the E-passport Scan network module is to compute the risk/probability of a passport being accepted as valid without any information about variable A. Inference cannot proceed until a joint posterior distribution for variables H and F is received from a parent network. This network has nodes H, S, C, and V. The E-passport Scan Network is shown in Figure 4.5.

Example Scenarios:

- A traveller's e-passport is passing the security features check, but due to damage is failing the chip-optical data crosscheck. Consequently, the e-passport is not authenticated by the automated system.
- A traveller does not hold an e-passport, so the nonexistent passport is not authenticated.

The probability distributions have been chosen with regard to the following assumptions:

- If the traveler does not hold an e-passport, then the security features check will fail by default: $H = h_2 \implies S = s_2$

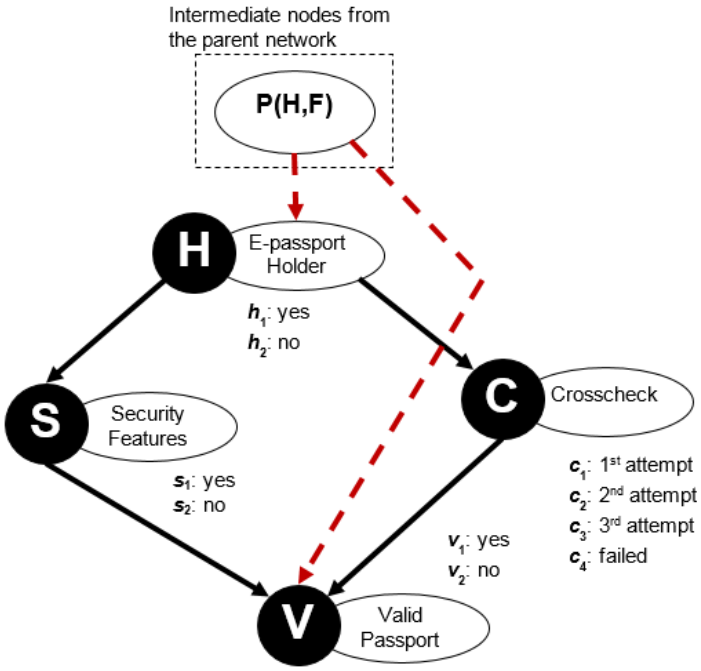


Figure 4.5: The E-passport Scan Network

- If the traveler does not hold an e-passport, then the chip-optical data crosscheck will fail be default: $H = h_2 \implies C = c_4$

- If the security features check, or the chip-optical data crosscheck fail, then the passport will not be deemed valid: $(S = s_2 \vee C = c_4) \implies V = v_2$

- The probability of the passport being found valid ($\Pr(v_1)$) increases as a function of the probability of that the passport is not fraudulent ($\Pr(f_2)$).

The node H (e-passport holder) is a copy of the node H from the parent network. The probability distribution of node H is the marginal posterior distribution for H received from the parent network. The existence of this node is to ensure that the posterior probability values of H only once influences the posterior of the upcoming node V.

The probability values for nodes S, C, and V are shown in tables 4.11, 4.12, and 4.13 respectively.

Table 4.11: The probability distributions $\Pr(S|H)$ associated with S in the E-passport Scan Network.

$\Pr(S|H)$

H	$\Pr(s_1\|H)$	$\Pr(s_2\|H)$
h_1	0.99	0.01
h_2	0	1

Table 4.12: The probability distributions $\Pr(C|H)$ associated with C in the E-passport Scan Network.

$P(C|H)$

H	$\Pr(c_1\|H)$	$\Pr(c_2\|H)$
h_1	0.85	0.10
h_2	0	0
H	$\Pr(c_3\|H)$	$\Pr(c_4\|H)$
h_1	0.04	0.01
h_2	0	1

Table 4.13: The probability distributions $P(V|S,C)$ associated with V in the E-passport Scan Network.

$\Pr(V|S,C)$

S,C	$\Pr(v_1\|S,C)$	$\Pr(v_2\|S,C)$
s_1c_1	$0.2 \cdot \Pr(f_1)$	$1 - 0.2 \cdot \Pr(f_1)$
	$+0.99 \cdot \Pr(f_2)$	$-0.99 \cdot \Pr(f_2)$
s_1c_2	$0.1 \cdot \Pr(f_1)$	$1 - 0.1 \cdot \Pr(f_1)$
	$+0.85 \cdot \Pr(f_2)$	$-0.85 \cdot \Pr(f_2)$
s_1c_3	$0.05 \cdot \Pr(f_1)$	$1 - 0.05 \cdot \Pr(f_1)$
	$+0.70 \cdot \Pr(f_2)$	$-0.70 \cdot \Pr(f_2)$
s_1c_4	0	1
s_2c_1	0	1
s_2c_2	0	1
s_2c_3	0	1
s_2c_4	0	1

4.6.2 E-passport Scan Network Module 2

A variation of the E-passport Scan network module can be used if the E-passport Attack Network is chosen as the parent network, and it is required that the posterior distribution of the attack mode (variable A) affects the probability of the security features check failing and the probability of the chip-optical data crosscheck failing. Inference cannot proceed until the marginal posterior distributions for variables H, A, and F are received from the parent

network. This network has nodes H, S, C, and V. The E-passport Scan Network 2 is shown in Figure 4.6.

Example Scenarios:

- A traveller is using an e-passport cloned from skimmed data. Since their e-passport is a homemade forgery, it is more likely to fail the security features check and the chip-optical data crosscheck. The passport then fails both checks and is not authenticated.

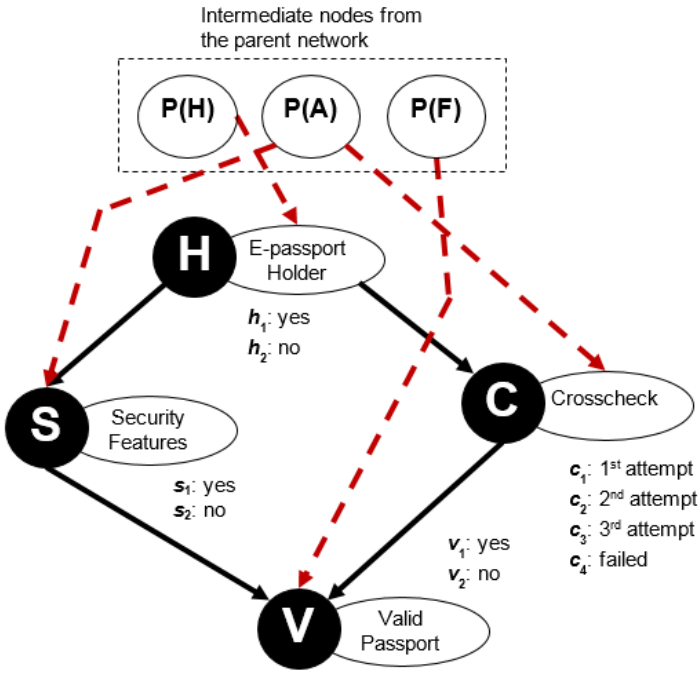


Figure 4.6: The E-passport Scan Network 2

The probability distributions in the E-passport Scan Network 2 are identical to the distributions in the E-passport Scan Network with the exception of nodes S and C. The probability distributions assigned to these nodes have the additional dependence on the posterior distribution of A. Since $A = a_4$ is the only scenario (aside from the possibility of $A = a_5$, "other attack") where a fake passport is being used, only $\Pr(a_4)$ will affect

the probabilities of S and C. These new distributions are shown in tables 4.14 and 4.15 respectively.

Table 4.14: The probability distributions $\Pr(S|H)$ associated with S in the E-passport Scan Network 2.

$\Pr(S|H)$

H	$\Pr(s_1\|H)$	$\Pr(s_2\|H)$
h_1	$0.99 - 0.03 \cdot \Pr(a_4)$	$0.01 + 0.03 \cdot \Pr(a_4)$
h_2	0	1

Table 4.15: The probability distributions $\Pr(C|H)$ associated with C in the E-passport Scan Network 2.

$\Pr(C|H)$

H	$\Pr(c_1\|H)$	$\Pr(c_2\|H)$
h_1	$0.85 - 0.01 \cdot \Pr(a_4)$	$0.10 - 0.01 \cdot \Pr(a_4)$
h_2	0	0
H	$\Pr(c_3\|H)$	$\Pr(c_4\|H)$
h_1	$0.04 - 0.01 \cdot \Pr(a_4)$	$0.01 + 0.03 \cdot \Pr(a_4)$
h_2	0	1

4.7 Facial Verification Network Module

The primary purpose of the facial verification network module is to incorporate the risk/probability of a traveler attempting to use the travel documents of another person. Inference cannot proceed until marginal posterior distributions for variables W, F, and V are received from the parent network(s). This network has nodes I, J, and M. The Facial Verification Network is shown in Figure 4.7.

Example Scenarios:

- A traveler is holding an e-passport that was deemed valid but is trying to impersonate another individual. Since the false accept rate for face verification is low [63], authentication fails and the traveler is directed to manual control.

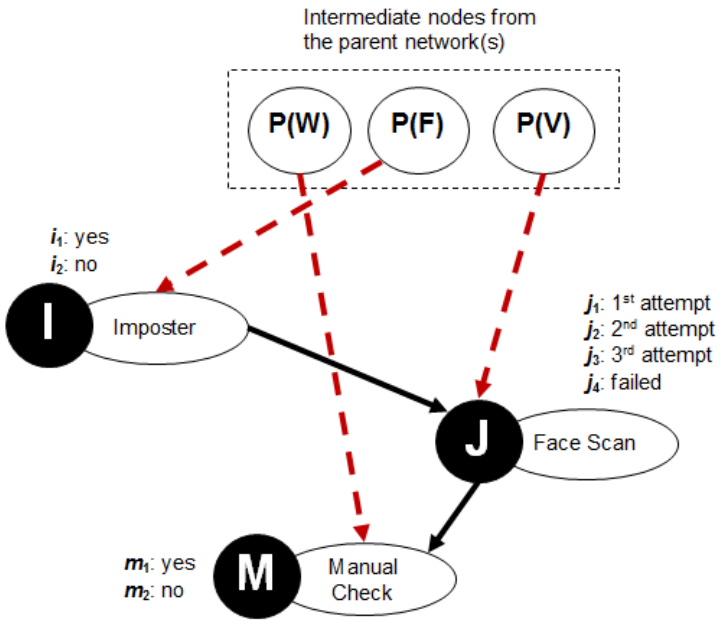


Figure 4.7: The Facial Verification Network

The probability distributions have been chosen with regard to the following assumptions:

- The probability that the traveler is an impostor, $\Pr(i_1)$, increases with respect to $\Pr(f_1)$, the probability that the passport was obtained fraudulently.
- If the traveller does not have a valid passport, the face scan will fail by default since the e-passport stores the biometric templates: $\Pr(v_2) = 1 \implies J = j_4$
- If the traveller is on a watchlist they will be directed to a manual check: $\Pr(w_1) = 1 \implies M = m_1$

The probability values for nodes I, J, and M are shown in tables 4.16, 4.17, and 4.18 respectively. For node J, the chosen probabilities are motivated by the FAR (false accept rate) and the FRR (false reject rate) rates listed in the conclusion of [63].

Table 4.16: The probability distributions $\Pr(I)$ associated with I in the Facial Verification Network.

$\Pr(I)$

	$\Pr(i_1)$	$\Pr(i_2)$
$\Pr(f_1) < 0.5$	$1.6 \cdot \Pr(f_1)$	$1 - 1.6 \cdot \Pr(f_1)$
$\Pr(f_1) \geq 0.5$	0.80	0.20

Table 4.17: The probability distributions $\Pr(J|I)$ associated with J in the Facial Verification Network.

$\Pr(J|I)$

I	$\Pr(j_1\|I)$	$\Pr(j_2\|I)$
i_1	$0.0002 \cdot \Pr(v_1)$	$0.0003 \cdot \Pr(v_1)$
i_2	$0.9 \cdot \Pr(v_1)$	$0.07 \cdot \Pr(v_1)$
I	$\Pr(j_3\|I)$	$\Pr(j_4\|I)$
i_1	$0.0005 \cdot \Pr(v_1)$	$1 - 0.001 \cdot \Pr(v_1)$
i_2	$0.005 \cdot \Pr(v_1)$	$1 - 0.975 \cdot \Pr(v_1)$

Table 4.18: The probability distributions $\Pr(M|J)$ associated with M in the Facial Verification Network.

$\Pr(M|J)$

J	$\Pr(m_1\|J)$	$\Pr(m_2\|J)$
j_1	$1 - 0.99 \cdot \Pr(w_2)$	$0.99 \cdot \Pr(w_2)$
j_2	$1 - 0.85 \cdot \Pr(w_2)$	$0.85 \cdot \Pr(w_2)$
j_3	$1 - 0.70 \cdot \Pr(w_2)$	$0.70 \cdot \Pr(w_2)$
j_4	1	0

4.8 Biometric Modality Fusion Network Module

This network is motivated by the work in [35]. Similar to the Bayesian network described in [35], the purpose of this network is to draw a single conclusion regarding whether the traveller is an impostor given the measurements of a number of biometric modalities. Inference cannot proceed until marginal posterior distributions for variable F and V are received from the parent network. This network has nodes I, J, J', and J''. The Biometric Fusion Network is shown in Figure 4.8.

Example Scenarios:

- A traveller holds an e-passport that was not fraudulently obtained and that has

been deemed to be valid. The traveller then passes the face and iris scan, but fails the fingerprint scan. The posterior probability of the traveller being an impostor is subsequently used to redirect the traveller to manual control or otherwise.

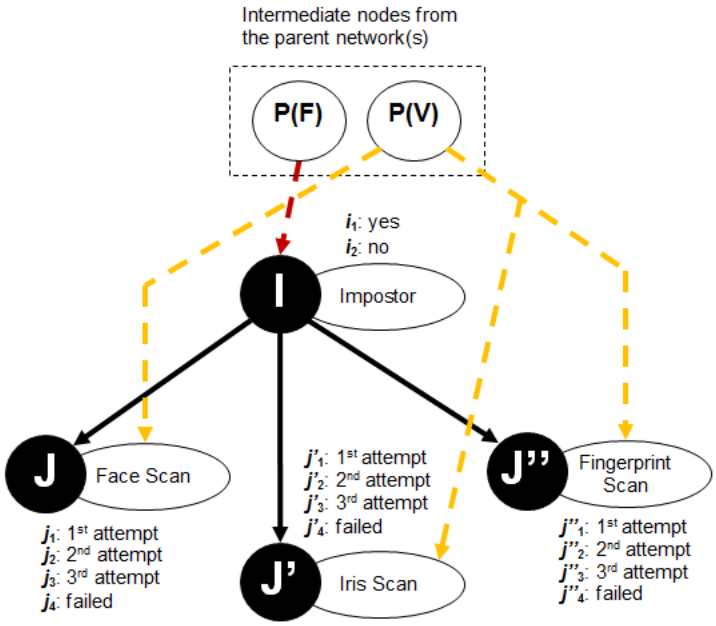


Figure 4.8: The Biometric Fusion Network Module

The probability distributions associated with nodes I and J are identical to the probability distributions associated with nodes I and J from the Facial Verification Network. For nodes J' and J'', like how the probabilities for node J were motivated by the values listed in [63], the probabilities for nodes J' and J'' are motivated by values listed in [7]. The same assumptions made for node J are also made for nodes J' and J'':

- If the traveller does not have a valid passport, the iris and fingerprint scan will fail by default since the e-passport stores the biometric templates: $\Pr(v_2) = 1 \implies (J' = j'_4 \wedge J'' = j''_4)$

The probability distributions for J' and J'' are listed in tables 4.19 and 4.20 respectively.

Table 4.19: The probability distributions $\Pr(J'|I)$ associated with J' in the Biometric Fusion Network.

$\Pr(J'|I)$

I	$\Pr(j'_1\|I)$	$\Pr(j'_2\|I)$
i_1	$0.001 \cdot \Pr(v_1)$	$0.002 \cdot \Pr(v_1)$
i_2	$0.9 \cdot \Pr(v_1)$	$0.03 \cdot \Pr(v_1)$
I	$\Pr(j'_3\|i)$	$\Pr(j'_4\|i)$
i_1	$0.002 \cdot \Pr(v_1)$	$1 - 0.005 \cdot \Pr(v_1)$
i_2	$0.03 \cdot \Pr(v_1)$	$1 - 0.96 \cdot \Pr(v_1)$

Table 4.20: The probability distributions $\Pr(J''|I)$ associated with J'' in the Biometric Fusion Network.

$\Pr(J''|I)$

I	$\Pr(j''_1\|I)$	$\Pr(j''_2\|I)$
i_1	$0.01 \cdot \Pr(v_1)$	$0.01 \cdot \Pr(v_1)$
i_2	$0.7 \cdot \Pr(v_1)$	$0.1 \cdot \Pr(v_1)$
I	$\Pr(j''_3\|I)$	$\Pr(j''_4\|I)$
i_1	$0.02 \cdot \Pr(v_1)$	$1 - 0.04 \cdot \Pr(v_1)$
i_2	$0.02 \cdot \Pr(v_1)$	$1 - 0.82 \cdot \Pr(v_1)$

4.9 "Mantrap" Network Module

The primary objective of the mantrap network is to model the waiting time risks of travelers using the mantrap system. This network has nodes H, A', M, E, and W'. The network is shown in Figure 4.9.

Example Scenarios:

- A traveller holds an e-passport that passes authentication. The traveller is not directed to a manual check and crosses the border in under 10 minutes.
- A traveller holds an e-passport, but it fails authentication. The traveller is then directed to a manual check. The traveller ultimately passes the border, but has been made to wait for longer than 10 minutes.
- A traveller does not hold an e-passport and is directed to a manual check. At the manual check, the traveller is found to be unauthorized and is denied access to the

border.

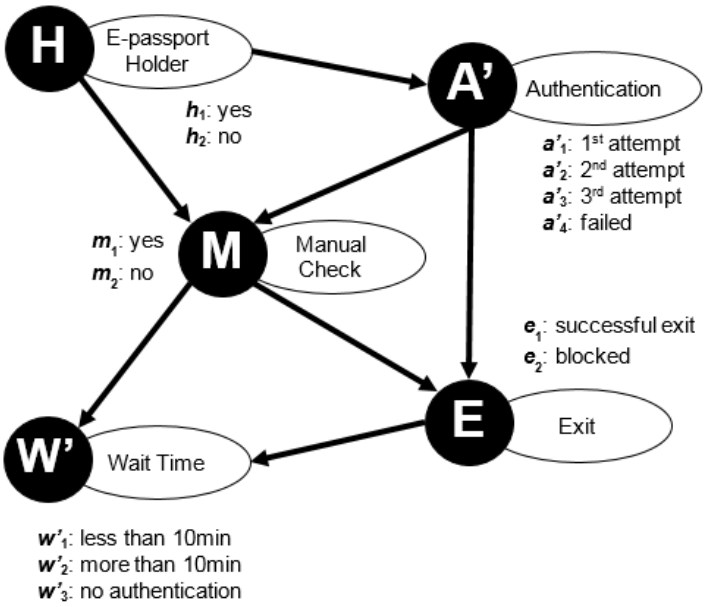


Figure 4.9: The Mantrap Network

The probability distributions have been chosen with regard to the following assumptions:

- If the traveller does not hold an e-passport, then e-passport authentication will fail by default: $H = h_2 \implies A' = a'_4$
- If the traveller does not hold an e-passport, then the traveller is directed to manual check: $H = h_2 \implies M = m_1$
- If the e-passport authentication fails, then the traveller is directed to manual check: $A' = a'_4 \implies M = m_1$
- If the traveller is not directed to manual check, then the traveller exits the border crossing successfully: $M = m_2 \implies E = e_1$
- If the traveller fails to exit the border crossing successfully, his/her wait time is theoretically infinite: $E = e_2 \implies W' = w'_3$

The probability distributions for nodes H, A', M, E, and W' are shown in tables 4.21, 4.22, 4.23, 4.24, and 4.25 respectively.

Table 4.21: The probability distribution $\Pr(H)$ associated with H in the Mantrap Network.

$\Pr(H)$	
$\Pr(h_1)$	$\Pr(h_2)$
0.98	0.02

Table 4.22: The probability distributions $\Pr(A'|H)$ associated with A' in the Mantrap Network.

$\Pr(A'\|H)$		
H	$\Pr(a'_1\|H)$	$\Pr(a'_2\|H)$
h_1	0.70	0.15
h_2	0	0
H	$\Pr(a'_3\|H)$	$\Pr(a'_4\|H)$
h_1	0.05	0.10
h_2	0	1

Table 4.23: The probability distributions $\Pr(M|H, A')$ associated with M in the Mantrap Network.

$\Pr(M\|H, A')$		
H, A'	$\Pr(m_1\|H, A')$	$\Pr(m_2\|H, A')$
$h_1a'_1$	0.01	0.99
$h_1a'_2$	0.02	0.98
$h_1a'_3$	0.03	0.97
$h_1a'_4$	1	0
$h_2a'_1$	1	0
$h_2a'_2$	1	0
$h_2a'_3$	1	0
$h_2a'_4$	1	0

4.10 Example modular model

In figure 4.10, an example modular model is shown. This modular model consists of the "simple e-passport holder module" (section 4.5.1, figure 4.3); the "e-passport scan network module 1" (section 4.6.1, figure 4.5); and the "facial verification network module" (section

Table 4.24: The probability distributions $\Pr(E|A', M)$ associated with E in the Mantrap Network.

$\Pr(E|A', M)$

A', M	$\Pr(e_1\|A', M)$	$\Pr(e_2\|A', M)$
$a'_1 m_1$	0.999	0.001
$a'_1 m_2$	1	0
$a'_2 m_1$	0.998	0.002
$a'_2 m_2$	1	0
$a'_3 m_1$	0.997	0.003
$a'_3 m_2$	1	0
$a'_4 m_1$	0.6	0.4
$a'_4 m_2$	1	0

Table 4.25: The probability distributions $\Pr(W'|M, E)$ associated with W' in the Mantrap Network.

$\Pr(W'|M, E)$

M, E	$\Pr(w'_1\|M, E)$	$\Pr(w'_2\|M, E)$	$\Pr(w'_3\|M, E)$
$m_1 e_1$	0.4	0.6	0
$m_1 e_2$	0	0	1
$m_2 e_1$	0.9	0.1	0
$m_2 e_2$	0	0	1

4.7, figure 4.7). This modular model describes the scenario of authenticating a biometric-enabled passport that uses the facial biometric. The "simple e-passport holder module"; "e-passport scan network module 1"; and "facial verification network module" are connected in series to reflect the sequential nature of the authentication process. The "simple e-passport holder module" describes the conditions leading to the authentication process. The "e-passport scan network module 1" describes the first stage of the authentication process of validating the e-passport itself. Lastly the "facial verification network module" verifies the holder's identity using the face biometric.

4.11 Conclusion

It is proven in [40, pg.288–290] that probabilistic inference using Bayesian networks is NP-hard. This makes probabilistic inference using large-scale Bayesian networks computationally intractable. To address this, the computational complexity of probabilistic inference using

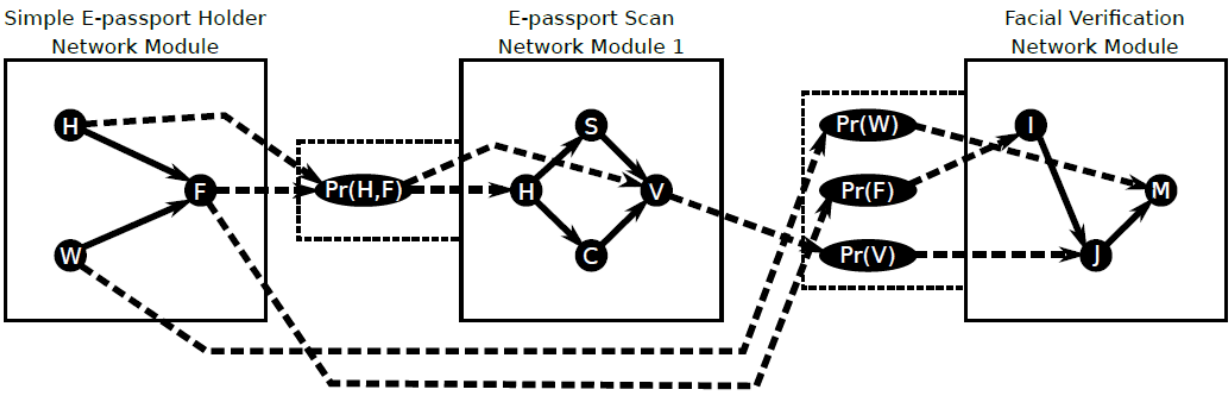


Figure 4.10: An example modular model.

modular models is linear with respect to the number of modules assuming that the size of each module is bounded from above.

In this chapter was presented an example library of Bayesian network modules. Each module represented a specific aspect of automated border control. To model a specific scenario from automated border control, modules are chosen and linked via appropriate interface nodes. From the observation that these modules provide an efficient way of breaking down a complicated scenario, it is concluded that these modules will provide an effective framework for analyzing a large complicated system without the need for complex Bayesian networks that require large amounts of statistical data to build, and for which probabilistic inference is computationally intractable.

Chapter 5

Graphical Models using Generalized Uncertainty Metrics

5.1 Introduction

This chapter will focus on the application of non-probabilistic uncertainty metrics to graphical paradigms such as causal networks. The primary contribution of this chapter is a novel compilation of how graphical models that utilize non-probabilistic metrics, such as Dempster-Shafer models, can be established and the various operations/algorithms involved. The application of different uncertainty metrics depending on the availability of statistical data and requirements on the output information, referred to as a "multi-metric inference engine" or "unified inference engine", is proposed as a novel contribution. With regards to Dempster-Shafer theory, the theory that is introduced in [22] is compiled and expanded upon. Emphasis is placed on the generalization of conditional probability tables to Dempster-Shafer theory, and their use in a Dempster-Shafer analog to Bayesian networks. Table 5.1 compiles a short list of existing forays into the theory of graphical models of uncertainty that utilize non-probabilistic metrics.

In Figure 5.1, the process of uncertainty inference with a choice of uncertainty metrics

Table 5.1: A description of existing work related to alternative uncertainty metrics and their associated graphical uncertainty models.

Reference(s)	Contribution
[11, 53]	The formulation of fuzzy probabilities as a replacement for probability distributions.
[17, 25, 68]	The formulation of probability intervals as a replacement for probability distributions.
[18]	The transferable belief model (a generalization of Bayes' rule), ballooning extensions and a two variable implementation of the Dempster-Shafer analogs to conditional probability tables which we will refer to as "conditional Dempster-Shafer tables".
[30, 31]	Undirected (non-causal) and directed (causal) graphical models that describe Dempster-Shafer models over a relatively large number of random variables. The rule of combination that is used is not analogous to Dempster's rule of combination however.
[38, 26, 39]	Establishes a Dempster-Shafer analog to Markov networks (factors and belief functions are referred to as "valuations"). An analysis of Dempster-Shafer analogs to the well known probabilistic inference algorithms: variable elimination and belief propagation is given.

is shown. Information such as the causal network, available data, limitations on the input data, and requirements on the output data are all considered to choose an uncertainty metric: probabilities; fuzzy probabilities; probability intervals; Dempster-Shafer models; or Dezert-Smarandache models. The resultant model, alongside observations related to the specific scenario, are processed by a "unified inference engine" that produces a final posterior uncertainty model that uses the chosen uncertainty metric.

Definition 5.1. The **multi-metric inference engine** or **unified inference engine** refers to the process depicted in figure 5.1 where an uncertainty metric is chosen on the basis of the available statistical data and output data requirements, and the appropriate inference algorithm based on the choice of uncertainty metric is used to compute a posterior uncertainty model that uses the chosen uncertainty metric. Tables 5.10 and 5.11 qualify the conditions for choosing an uncertainty metric.

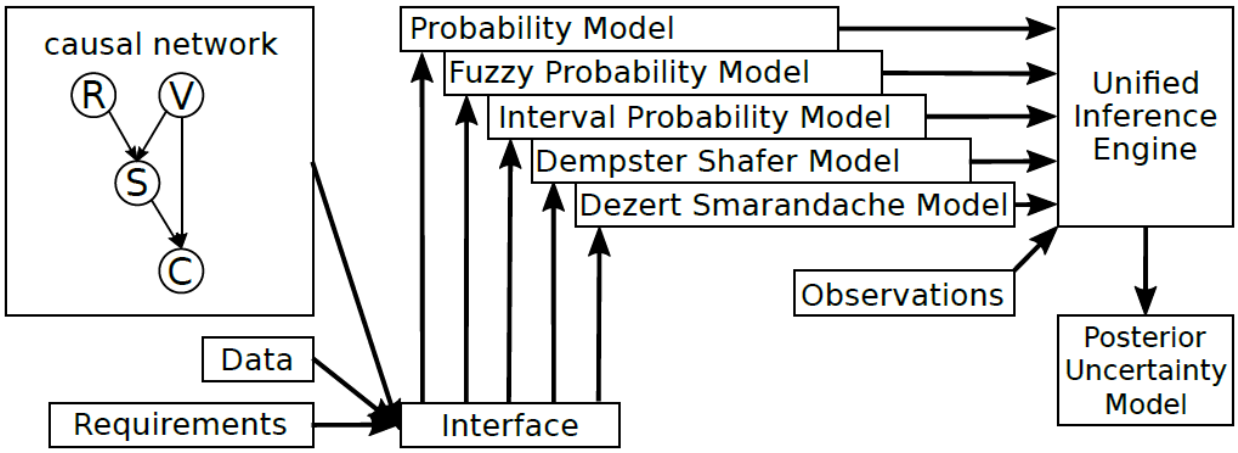


Figure 5.1: A high level depiction of uncertainty inference with a choice of uncertainty metrics.

5.2 Marginal and Conditioned Fuzzy Probability Distributions

Generalizing probability theory to fuzzy probabilities is a matter of directly substituting fuzzy numbers (fuzzy probabilities) in place of ordinary numbers (point probabilities). In order to properly carry out operations such as marginalization, conditioning, and computing joint fuzzy probability distributions from conditional fuzzy probability tables, arithmetic involving fuzzy numbers must be established using theory from [53]:

- Given two fuzzy numbers (l_1, c_1, u_1) and (l_2, c_2, u_2), the sum is defined by $(l_1, c_1, u_1) + (l_2, c_2, u_2) = (l_1 + l_2, c_1 + c_2, u_1 + u_2)$.
- Given a fuzzy number (l, c, u), the negative is defined by $-(l, c, u) = (-u, -c, -l)$.
- Given two fuzzy numbers (l_1, c_1, u_1) and (l_2, c_2, u_2), the product is defined by $(l_1, c_1, u_1) \cdot (l_2, c_2, u_2) = (\min(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2), c_1 c_2, \max(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2))$.
- Given a fuzzy number (l, c, u) where $0 \notin (l, u)$, the reciprocal is defined by $(l, c, u)^{-1} = (1/u, 1/c, 1/l)$.

With these operations, graphical models that utilize fuzzy probabilities are directly analogous to graphical models that utilize point probabilities.

5.3 Marginal and Conditioned Probability Interval Distributions

Consider a probability interval distribution over the variables $\mathcal{X}$. Given an arbitrary assignment $V \in \text{Val}(\mathcal{X})$, the probability interval $\Pr_I(V) = [\Pr_L(V), \Pr_U(V)]$ associated with V contains the true probability $\Pr(V)$.

At least one point probability distribution should be allowed:

$$\sum_{V \in \text{Val}(\mathcal{X})} \Pr_L(V) \leq 1 \leq \sum_{V \in \text{Val}(\mathcal{X})} \Pr_U(V)$$

Each bound should be reachable:

$$\forall V \in \text{Val}(\mathcal{X}) : \Pr_L(V) \geq 1 - \sum_{V' \in \text{Val}(\mathcal{X}) \wedge V' \neq V} \Pr_U(V')$$

and

$$\forall V \in \text{Val}(\mathcal{X}) : \Pr_U(V) \leq 1 - \sum_{V' \in \text{Val}(\mathcal{X}) \wedge V' \neq V} \Pr_L(V')$$

The processes of marginalization and conditioning probability interval distributions using theory from [17] will now be given:

5.3.1 Marginalization

The process of marginalizing probability interval distributions is adapted from [17].

Given a subset $\mathcal{Y}$ of $\mathcal{X}$, $\mathcal{Y} \subseteq \mathcal{X}$, marginalizing a probability interval distribution over the variables in $\mathcal{X}$ to the variables in $\mathcal{Y}$ is done by the following process:

Given an arbitrary assignment $V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y})$, the marginal probability $\Pr(V_{\mathcal{Y}}) = \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] = V_{\mathcal{Y}}} \Pr(V')$ is bounded from below by $\sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] = V_{\mathcal{Y}}} \Pr_L(V')$ and from above by $\sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] = V_{\mathcal{Y}}} \Pr_U(V')$. However, these bounds are not necessarily reachable, and can be made tighter.

Consider all assignments V' from $\mathrm{Val}(\mathcal{X})$ that are not consistent with $V_{\mathcal{Y}}$: $V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] \neq V_{\mathcal{Y}}$. If all probabilities $\Pr(V')$ associated with these assignments are increased to their maximum possible value $\Pr_U(V')$, the sum $\Pr(V_{\mathcal{Y}}) = \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] = V_{\mathcal{Y}}} \Pr(V')$ may not be able to decrease to the lower bound $\sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] = V_{\mathcal{Y}}} \Pr_L(V')$. This establishes a second lower bound for $\Pr(V_{\mathcal{Y}})$: $1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] \neq V_{\mathcal{Y}}} \Pr_U(V')$. Conversely, there is a second upper bound for $\Pr(V_{\mathcal{Y}})$: $1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] \neq V_{\mathcal{Y}}} \Pr_L(V')$.

The lower and upper bounds for $\Pr(V_{\mathcal{Y}})$ are:

$$\Pr_L(V_{\mathcal{Y}}) = \max \left(\sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] = V_{\mathcal{Y}}} \Pr_L(V'), 1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] \neq V_{\mathcal{Y}}} \Pr_U(V') \right)$$

and

$$\Pr_U(V_{\mathcal{Y}}) = \min \left(\sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] = V_{\mathcal{Y}}} \Pr_U(V'), 1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Y}] \neq V_{\mathcal{Y}}} \Pr_L(V') \right)$$

respectively.

5.3.2 Conditioning

The process of conditioning probability interval distributions is adapted from [17].

Given a subset $\mathcal{Z}$ of $\mathcal{X}$, $\mathcal{Z} \subseteq \mathcal{X}$, as well as an assignment $V_{\mathcal{Z}} \in \mathrm{Val}(\mathcal{Z})$, conditioning the probability interval distribution using the evidence $V_{\mathcal{Z}}$ proceeds as follows:

Let $V \in \mathrm{Val}(\mathcal{X} \setminus \mathcal{Z})$ be an arbitrary assignment to the variables that remain when the variables from $\mathcal{Z}$ are removed from $\mathcal{X}$. The conditional probability of V is $\Pr(V|V_{\mathcal{Z}}) = \frac{\Pr(\langle V, V_{\mathcal{Z}} \rangle)}{\Pr(V_{\mathcal{Z}})}$. In order to compute $\Pr_L(V|V_{\mathcal{Z}})$, the probability $\Pr(V_{\mathcal{Z}})$ needs to be maximized while $\Pr(\langle V, V_{\mathcal{Z}} \rangle) = \Pr_L(\langle V, V_{\mathcal{Z}} \rangle)$. With the restriction that $\Pr(\langle V, V_{\mathcal{Z}} \rangle) = \Pr_L(\langle V, V_{\mathcal{Z}} \rangle)$, an

upper bound on $\Pr(V_{\mathcal{Z}})$ is $\Pr_L(\langle V, V_{\mathcal{Z}}\rangle) + \sum_{V' \in \mathrm{Val}(\mathcal{X}\setminus\mathcal{Z}) \wedge V' \neq V} \Pr_U(\langle V', V_{\mathcal{Z}}\rangle)$. Another upper bound can be formed by setting all probabilities for assignments $V' \in \mathrm{Val}(\mathcal{X})$ that do not match $V_{\mathcal{Z}}$ ($V'[\mathcal{Z}] \neq V_{\mathcal{Z}}$) to their minimum $\Pr_L(V')$. The 2nd upper bound formed is $1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Z}] \neq V_{\mathcal{Z}}} \Pr_L(V')$. The maximum value for $\Pr(V_{\mathcal{Z}})$ is
$\min\Big(\Pr_L(\langle V, V_{\mathcal{Z}}\rangle) + \sum_{V' \in \mathrm{Val}(\mathcal{X}\setminus\mathcal{Z}) \wedge V' \neq V} \Pr_U(\langle V', V_{\mathcal{Z}}\rangle), 1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Z}] \neq V_{\mathcal{Z}}} \Pr_L(V')\Big)$.

Therefore:

$$\Pr_L(V|V_{\mathcal{Z}}) = \frac{\Pr_L(\langle V, V_{\mathcal{Z}}\rangle)}{\min\Big(\Pr_L(\langle V, V_{\mathcal{Z}}\rangle) + \sum_{V' \in \mathrm{Val}(\mathcal{X}\setminus\mathcal{Z}) \wedge V' \neq V} \Pr_U(\langle V', V_{\mathcal{Z}}\rangle), 1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Z}] \neq V_{\mathcal{Z}}} \Pr_L(V')\Big)}$$

and by similar reasoning,

$$\Pr_U(V|V_{\mathcal{Z}}) = \frac{\Pr_U(\langle V, V_{\mathcal{Z}}\rangle)}{\max\Big(\Pr_U(\langle V, V_{\mathcal{Z}}\rangle) + \sum_{V' \in \mathrm{Val}(\mathcal{X}\setminus\mathcal{Z}) \wedge V' \neq V} \Pr_L(\langle V', V_{\mathcal{Z}}\rangle), 1 - \sum_{V' \in \mathrm{Val}(\mathcal{X}) \wedge V'[\mathcal{Z}] \neq V_{\mathcal{Z}}} \Pr_U(V')\Big)}$$

5.3.3 Forming Joint Probability Interval Distributions

The process of forming joint probability interval distributions is well known.

Given two disjoint sets of variables $\mathcal{Y}$ and $\mathcal{Z}$ ($\mathcal{Y} \subseteq \mathcal{X}$, $\mathcal{Z} \subseteq \mathcal{X}$, and $\mathcal{Y} \cap \mathcal{Z} = \emptyset$) that are independent ($\mathcal{Y} \perp \mathcal{Z}$), the joint probability interval distribution over the variables $\mathcal{Y} \cup \mathcal{Z}$ can be formed from the marginal probability interval distributions for $\mathcal{Y}$ and $\mathcal{Z}$:

Given an arbitrary assignment $V \in \mathrm{Val}(\mathcal{Y} \cup \mathcal{Z})$, the lower and upper bounds on $\Pr(V)$ are $\Pr_L(V) = \Pr_L(V[\mathcal{Y}])\Pr_L(V[\mathcal{Z}])$ and $\Pr_U(V) = \Pr_U(V[\mathcal{Y}])\Pr_U(V[\mathcal{Z}])$.

It is important to note that the information that variable sets $\mathcal{Y}$ and $\mathcal{Z}$ are independent is lost when the joint probability interval distribution is formed. Variable independence, or the lack thereof, cannot be inferred from joint probability interval distributions.

5.4 Marginal and Conditioned Dempster-Shafer (DS) models

Descriptions of Dempster-Shafer theory can be found in [37, chapter 5], and [67, 71, 76, 74].

A description of Dempster-Shafer (DS) models can be found in section: 3.3.4.

Given a DS model D, the notation $\text{Var}(D)$ denotes the set of variables covered by D.

The DS model that stores no information consists of a single focal element that is the set of all possible assignments. The assigned probability is 1. This DS model is referred to as the "vacuous" model [20] and is denoted by $\mathbf{1}_{\mathcal{Y}}$, where $\mathcal{Y}$ is the set of variables covered by the DS model: $\text{Var}(\mathbf{1}_{\mathcal{Y}}) = \mathcal{Y}$. The set of focal elements is $\mathcal{E}(\mathbf{1}_{\mathcal{Y}}) = \{\text{Val}(\mathcal{Y})\}$, and $m(\text{Val}(\mathcal{Y})|\mathbf{1}_{\mathcal{Y}}) = 1$.

5.4.1 Marginalization

Given a subset $\mathcal{Y}$ of $\mathcal{X}$, $\mathcal{Y} \subseteq \mathcal{X}$, marginalizing a Dempster-Shafer model D that covers the variables in $\mathcal{X}$ to a Dempster-Shafer model $\mathbf{marg}\,(D|\mathcal{Y})$ that covers the variables in $\mathcal{Y}$ is done by the following process. The process of marginalizing Dempster-Shafer models to cover a smaller set of variables is referred to as "coarsening" in [18, 56].

The focal elements, which are the sets from the power set $\text{Set}(\mathcal{Y})$ that may be assigned a non-zero probability mass, are the cylindrical projections of the focal elements from D onto the variables from $\mathcal{Y}$: $\mathcal{E}(\mathbf{marg}\,(D|\mathcal{Y})) = \{J[\mathcal{Y}]|J \in \mathcal{E}(D)\}$. See the introductory chapter on notation for a description of the cylindrical projection of a set $J \in \text{Set}(\mathcal{X})$ onto the variable set $\mathcal{Y}$, which is denoted by $J[\mathcal{Y}]$.

Given a focal element J from $\mathcal{E}(\mathbf{marg}\,(D|\mathcal{Y}))$, the probability mass assigned to J is

$$m(J|\mathbf{marg}\,(D|\mathcal{Y})) = \sum_{J' \in \mathcal{E}(D) \wedge J'[\mathcal{Y}] = J} m(J'|D)$$

5.4.2 Conditioning

Let $\mathcal{Z}$ be a subset of $\mathcal{X}$, $\mathcal{Z} \subseteq \mathcal{X}$, and let $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$ be an assignment to the variables from $\mathcal{Z}$. Conditioning a Dempster-Shafer model D that covers the variables in $\mathcal{X}$ to a Dempster-Shafer model $\mathbf{cond}\,(D|V_{\mathcal{Z}})$ that covers variables $\mathcal{X} \setminus \mathcal{Z}$ using the evidence $V_{\mathcal{Z}}$ proceeds as follows [18]:

The focal elements are $\mathcal{E}(\mathbf{cond}\,(D|V_{\mathcal{Z}})) = \{(J \cap \{V_{\mathcal{Z}}\})[\mathcal{X} \setminus \mathcal{Z}] | J \in \mathcal{E}(D)\} \setminus \{\emptyset\}$ (recall from the notation chapter that when the intersection of sets of variable assignments is performed, both sets of variables are cylindrically extrapolated to cover the same set of variables. $J \cap \{V_{\mathcal{Z}}\}$ is set J with all assignments that do not agree with $V_{\mathcal{Z}}$ removed. $(J \cap \{V_{\mathcal{Z}}\})[\mathcal{X} \setminus \mathcal{Z}]$ is set $J \cap \{V_{\mathcal{Z}}\}$ with the assignments to the variables in $\mathcal{Z}$ removed from each assignment.)

Given a focal element J from $\mathcal{E}(\mathbf{cond}\,(D|V_{\mathcal{Z}}))$, the probability mass assigned to J is

$$m(J|\mathbf{cond}\,(D|V_{\mathcal{Z}})) = \frac{1}{K} \sum_{J' \in \mathcal{E}(D) \wedge (J' \cap \{V_{\mathcal{Z}}\})[\mathcal{X} \setminus \mathcal{Z}] = J} m(J'|D)$$

where $K = \sum_{J \in \mathcal{E}(D) \wedge J \cap \{V_{\mathcal{Z}}\} \neq \emptyset} m(J|D)$ is a normalization constant so that all probability masses sum to 1.

5.4.3 Forming joint DS models

Given two disjoint sets of variables $\mathcal{Y}$ and $\mathcal{Z}$ ($\mathcal{Y} \subseteq \mathcal{X}$, $\mathcal{Z} \subseteq \mathcal{X}$, and $\mathcal{Y} \cap \mathcal{Z} = \emptyset$) that are independent ($\mathcal{Y} \perp \mathcal{Z}$), the joint Dempster-Shafer models over the variables $\mathcal{Y} \cup \mathcal{Z}$ can be formed from the marginal Dempster-Shafer models for $\mathcal{Y}$ and $\mathcal{Z}$ using an analogue to "conjunctive combination" [18]:

Let $D_{\mathcal{Y}}$ and $D_{\mathcal{Z}}$ denote the marginal DS models that cover the variables $\mathcal{Y}$ and $\mathcal{Z}$ respectively.

The focal elements of $D_{\mathcal{Y}} \times D_{\mathcal{Z}}$ are $\mathcal{E}(D_{\mathcal{Y}} \times D_{\mathcal{Z}}) = \{J_{\mathcal{Y}} \cap J_{\mathcal{Z}} | J_{\mathcal{Y}} \in \mathcal{E}(D_{\mathcal{Y}}) \wedge J_{\mathcal{Z}} \in \mathcal{E}(D_{\mathcal{Z}})\}$. (Recall that sets $J_{\mathcal{Y}}$ and $J_{\mathcal{Z}}$ are cylindrically extended to cover $\mathcal{Y} \cup \mathcal{Z}$ before the intersection is performed.)

Given a focal element J from $\mathcal{E}(D_{\mathcal{Y}} \times D_{\mathcal{Z}})$, the probability mass assigned to J is

$$m(J|D_{\mathcal{Y}} \times D_{\mathcal{Z}}) = m(J[\mathcal{Y}]|D_{\mathcal{Y}})m(J[\mathcal{Z}]|D_{\mathcal{Z}})$$

Similar to the situation with probability interval distributions, it is important to note that the information that variable sets $\mathcal{Y}$ and $\mathcal{Z}$ are independent is lost when the joint Dempster-Shafer model is formed. Variable independence, or the lack thereof, cannot be inferred from Dempster-Shafer models. If fact, simple probability distributions are the only uncertainty model from which variable independence can be established with certainty.

5.4.4 About DSm models

Dezert-Smarandache (DSm) models are almost exactly the same as Dempster-Shafer models, except for the fact that the focal elements are sets built from the operators $\cup$ and $\cap$, where the primitive expressions are the individual values of the variable's domain. Marginalization, conditioning, and the forming of joint DSm models proceeds in an identical manner for DSm models as they do for DS models.

5.5 Graphical Uncertainty Models

This section will detail analogs to Markov and Bayesian networks using non-probabilistic metrics, such as fuzzy probabilities, probability interval distributions, Dempster-Shafer models, and Dezert-Smarandache models.

5.5.1 Causal networks

This section will detail a generalization of Bayesian networks to other uncertainty metrics. A Bayesian network that has been stripped of all probability tables is a "causal network":

Definition 5.2. Given the collection of random variables $\mathcal{X}$, a **causal network** is graphical

uncertainty model where each random variable is depicted as a node is a directed acyclic graph (DAC). Given two random variables/nodes x_1 and x_2, a directed edge extends from x_1 to x_2 if and only if the value attained by x_1 has a direct causal influence on the value attained by x_2. It is also the case that the value attained by x_2 is determined at a later point in time than the value attained by x_1.

Each node in the causal network is assigned a "conditional uncertainty table" (CUT), which quantifies how a node/variable depends on its parents. In a Bayesian network conditional probability tables quantify how nodes depend on their parents. The generalization of conditional probability tables to general uncertainty metrics are "conditional uncertainty tables":

Definition 5.3. Assume a specific uncertainty metric. Consider a variable x and a set of variables $\mathcal{Y}$ where $x \notin \mathcal{Y}$. A "conditional uncertainty table (CUT) over x, dependent on $\mathcal{Y}$" is a structure where an uncertainty model over x that uses the specific uncertainty metric is assigned to each possible assignment to the variables in $\mathcal{Y}$. Each uncertainty model over x is a "row" in the conditional uncertainty table over x.

- When probabilities are the chosen uncertainty metric, the causal network becomes a "Bayesian network" (BN), and the CUTs are "Conditional Probability Tables" (**CPT**s).
- When fuzzy probabilities are the chosen uncertainty metric, the causal network becomes a "fuzzy Bayesian network" (FBN), and the CUTs are "Conditional Fuzzy Probability Tables" (**CFPT**s).
- When probability intervals are the chosen uncertainty metric, the causal network becomes a "probability interval Bayesian network (PIBN)", and the CUTs are "Conditional Probability Interval Tables" (**CPIT**s).

- When Dempster-Shafer (DS) models are the chosen uncertainty metric, the causal network becomes a "Dempster-Shafer Bayesian network (DSBN)", and the CUTs are "Conditional DS Tables" (**CDST**s).

- When Dezert-Smarandache (DSm) models are the chosen uncertainty metric, the causal network becomes a "Dezert-Smarandache Bayesian network (DSmBN)", and the CUTs are "Conditional DSm Tables" (**CDSmT**s).

5.5.2 Graphical Models using Fuzzy Probabilities

Generalizing Markov and Bayesian networks to fuzzy probabilities is as simple as replacing the factor entries with fuzzy numbers in the case of Markov networks, or probabilities with fuzzy probabilities in the case of Bayesian networks. Algorithms for probabilistic inference (computing marginal and conditional probabilities) remain unchanged. Normalization however, should be done last to avoid numerical instabilities.

5.5.3 Graphical Models using Probability Intervals

Generalizing Markov and Bayesian networks to probability intervals is done by replacing the factor entries with intervals of non-negative numbers in the case of Markov networks, or probabilities with probability intervals in the case of Bayesian networks. Unlike with fuzzy probabilities, care needs to be taken during probabilistic inference in order to ensure that the interval bounds remain tight. Probabilistic inference proceeds by first computing the total joint probability interval distribution, and then eliminating the non-query, non-evidence variables via marginalization using the process given for marginalization, and lastly conditioning the evidence variables via the process given for conditioning.

5.5.4 Graphical Models using Dempster-Shafer models

Dempster-Shafer Markov Networks (DS-MNs)

In [38, 26, 39], graphical uncertainty structures referred to as "valuation networks" are formulated. DS analogs to Markov networks are encompassed by this valuation network framework. This section however, will present a mathematically rigorous formulation of the Dempster-Shafer variant of the "valuation networks", as opposed to addressing valuation networks in their most general form.

In probability theory, a Markov network, described in section 3.2.1, is a paradigm for describing a probability distribution over a large number of random variables as the product of belief functions over a small number of random variables. Each belief function (here, "belief" does not refer to the same "belief" associated with DS models discussed in section 3.3.4) is called a "factor". The total probability distribution is the product of all factors (belief functions) times a constant that normalized the distribution.

To establish the DS analog to Markov networks proposed by this paper, we will first describe the concept of a Dempster-Shafer "factor" (DSF). DSFs are referred to as "valuations" in [38, 26, 39].

Definition 5.4. A **Dempster-Shafer factor** (DSF) F is an DS model over a relatively small subset of variables denoted by $\text{Var}(F)$. Unlike DS models, the weights assigned to the focal elements ***do not*** have to be normalized to sum to 1. $D = \text{norm}(F)$ denotes the DS model derived from F via normalizing the focal element weights.

With the Markov-Networks from classical probability theory, factors are combined by multiplying together corresponding belief values. Here, we will describe the operation of multiplying (also referred to as combining) DSFs to form a DSF that encodes the information from both multiplicands. The operation of multiplication performed on DSFs, also known as "conjunctive combination" [18], is similar to Dempster's rule of combination [71] (see section 3.3.4) except for the fact that the product factor is **not normalized**.

Before we describe how DSFs can be multiplied, we first need to discuss "vacuous extensions" [18], which is the process of extending DSFs to cover additional variables without assuming any new information about the new variables.

Definition 5.5. Vacuous Extension [18]: Given DSF F, and a set of variables $\mathcal{Z} \supseteq \text{Var}(F)$, the **vacuous extension** of F to the variables from $\mathcal{Z}$, denoted by $F[\mathcal{Z}]$, is defined by:

$\text{Var}(F[\mathcal{Z}]) = \mathcal{Z}$

The focal elements of $F[\mathcal{Z}]$ are: $\mathcal{E}(F[\mathcal{Z}]) = \{J[\mathcal{Z}]|J \in \mathcal{E}(F)\}$

Given a focal element J from $\mathcal{E}(F[\mathcal{Z}])$, the probability mass assigned to J is $m(J|F[\mathcal{Z}]) = m(J[\text{Var}(F)]|F)$

The marginalization and conditioning of DSFs is done in a similar manner to the marginalization and conditioning of DS models, only no normalization is performed afterwards.

The multiplication of DSFs is similar to Dempster's rule of combination. The multiplication of DSFs is defined as follows:

Definition 5.6. DSF multiplication: DSFs F_1 and F_2 are multiplied together to get DSF $F_1 \times F_2$ via the following:

$\text{Var}(F_1 \times F_2) = \text{Var}(F_1) \cup \text{Var}(F_2)$

The focal elements of $F_1 \times F_2$ are: $\mathcal{E}(F_1 \times F_2) = \{J_1 \cap J_2 | J_1 \in \mathcal{E}(F_1) \wedge J_2 \in \mathcal{E}(F_2)\}$ (Recall that J_1 and J_2 are both cylindrically extended to cover variables $\text{Var}(F_1) \cup \text{Var}(F_2)$ before the intersection operator is applied.)

Given a focal element J from $\mathcal{E}(F_1 \times F_2)$, the probability mass assigned to J is $m(J|F_1 \times F_2) = \sum_{J_1 \in \mathcal{E}(F_1) \wedge J_2 \in \mathcal{E}(F_2) \wedge J = J_1 \cap J_2} m(J_1|F_1) m(J_2|F_2)$

As an example of DSF multiplication, consider two DSFs F_1 and F_2.

$\text{Var}(F_1) = \{X, Y\}$ and $\text{Var}(F_2) = \{Y, Z\}$.

Let $\text{Val}(X) = \{x_1, x_2\}$; $\text{Val}(Y) = \{y_1, y_2\}$; and $\text{Val}(Z) = \{z_1, z_2\}$.

Let $F_1 = \langle\{x_1y_1, x_1y_2\}, 0.5\rangle; \langle\{x_2y_1, x_2y_2\}, 0.5\rangle$ and $F_2 = \langle\{y_1z_1, y_2z_1\}, 0.4\rangle; \langle\{y_1z_2, y_2z_2\}, 0.6\rangle$.

It is then the case that $F_1 \times F_2 = \langle\{x_1y_1z_1, x_1y_2z_1\}, 0.2\rangle; \langle\{x_1y_1z_2, x_1y_2z_2\}, 0.3\rangle; \langle\{x_2y_1z_1, x_2y_2z_1\}, 0.2\rangle; \langle\{x_2y_1z_2, x_2y_2z_2\}, 0.3\rangle$.

With DSF multiplication established, we now define Dempster-Shafer Markov Networks (DS-MNs).

Definition 5.7. A DS-MN over $\mathcal{X}$ is characterized by a set of DSFs $\{F_1, F_2, \ldots, F_k\}$. Graphically the variables in $\mathcal{X}$ are visualized as nodes in a simple undirected graph. Given variables $x, y \in \mathcal{X}$, an edge exists between x and y iff there exists F_i such that $x, y \in \text{Var}(F_i)$. The total DS model denoted by the DS-MN is the product $D = \text{norm}(F_1 \times F_2 \times ... \times F_k)$.

The marginalization of a DSF F to variables from $\mathcal{Y} \subseteq \text{Var}(F)$, denoted by $\textbf{marg}\,(F|\mathcal{Y})$, is performed in the exact same manner as the marginalization of a Dempster-Shafer model.

As an example of DSF marginalization, consider DSF F:

$\text{Var}(F) = \{X, Y\}$ and $\text{Val}(X) = \{x_1, x_2\}$ and $\text{Val}(Y) = \{y_1, y_2\}$.

Let $F = \langle\{x_1y_1, x_1y_2\}, 0.1\rangle; \langle\{x_2y_1, x_2y_2\}, 0.2\rangle; \langle\{x_1y_1, x_2y_2\}, 0.4\rangle; \langle\{x_1y_2, x_2y_1\}, 0.8\rangle$.

It is then the case that $\textbf{marg}\,(F|X) = \langle\{x_1\}, 0.1\rangle; \langle\{x_2\}, 0.2\rangle; \langle\{x_1, x_2\}, 1.2\rangle$.

The conditioning of a DSF F using the evidence $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$ where $\mathcal{Z} \subseteq \text{Var}(F)$, denoted by $\textbf{cond}\,(F|V_{\mathcal{Z}})$, is performed in a similar manner to the conditioning of a Dempster-Shafer model. The primary difference is that no normalization of probability masses is performed afterwards.

As an example of DSF conditioning, consider DSF F:

$\text{Var}(F) = \{X, Y\}$ and $\text{Val}(X) = \{x_1, x_2\}$ and $\text{Val}(Y) = \{y_1, y_2\}$.

Let $F = \langle\{x_1y_1, x_1y_2\}, 0.2\rangle; \langle\{x_2y_1, x_2y_2\}, 0.3\rangle; \langle\{x_1y_1, x_2y_2\}, 0.4\rangle$.

It is then the case that $\mathbf{cond}\,(F|x_1) = \langle\{y_1, y_2\}, 0.2\rangle; \langle\{y_1\}, 0.4\rangle$.

Dempster-Shafer Bayesian Networks (DS-BNs)

In a traditional BN, the total probability distribution is computed by treating each conditional probability table as a Markov network factor, and then taking the product of all factors. For the case of a DS-BN, the conditional DS tables cannot be treated as DSFs in a trivial manner. For each variable $x \in \mathcal{X}$, the CDST assigned to x will denote a set of $|\text{Val}(\text{Pa}(x))|$ DSFs as opposed to a single Markov network factor in the classical probability case. For each $V_{\text{Pa}(x)} \in \text{Val}(\text{Pa}(x))$, let $D_{x|V_{\text{Pa}(x)}}$ denote the DS model over x that is indexed by $V_{\text{Pa}(x)}$. We will extend this DS model over x to a DSF over $\{x\} \cup \text{Pa}(x)$ denoted by $F_{x|V_{\text{Pa}(x)}}$ using the "Ballooning Extension" [18]:

Definition 5.8. The Ballooning Extension for sets [18]: Consider an arbitrary set J of assignments to the set of variables $\text{Var}(J)$, and an arbitrary assignment $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$ where $\text{Var}(J) \cap \mathcal{Z} = \emptyset$. The **ballooning extension** of J to cover the variables from $\text{Var}(J) \cup \mathcal{Z}$, denoted by $\mathbf{balloon}\,(J|V_{\mathcal{Z}})$, is defined by:

$\mathbf{balloon}\,(J|V_{\mathcal{Z}}) = (J \times \{V_{\mathcal{Z}}\}) \cup (\text{Val}(\text{Var}(J)) \times (\text{Val}(\mathcal{Z}) \setminus \{V_{\mathcal{Z}}\}))$

In essence, J is extended so that $V_{\mathcal{Z}}$ is appended to each existing assignment, and all assignments that do not agree with $V_{\mathcal{Z}}$ are also included.

Definition 5.9. The Ballooning Extension for DSFs [18]: Consider a DSF F, a set of variables $\mathcal{Z}$ such that $\mathcal{Z} \cap \text{Var}(F) = \emptyset$, and an arbitrary assignment $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$. The **ballooning extension** of F to cover the variables from $\text{Var}(F) \cup \mathcal{Z}$, denoted by $\mathbf{balloon}\,(F|V_{\mathcal{Z}})$, is defined by:

$\text{Var}(\mathbf{balloon}\,(F|V_{\mathcal{Z}})) = \text{Var}(F) \cup \mathcal{Z}$

The focal elements of $\mathbf{balloon}\,(F|V_{\mathcal{Z}})$ are $\mathcal{E}(\mathbf{balloon}\,(F|V_{\mathcal{Z}})) = \{\mathbf{balloon}\,(J|V_{\mathcal{Z}})\,|J \in \mathcal{E}(F)\}$

Given a focal element J from $\mathcal{E}(\mathbf{balloon}\,(F|V_{\mathcal{Z}}))$, the probability mass assigned to J is $m((J \cap \{V_{\mathcal{Z}}\})[\mathrm{Var}(F)]|F)$

In essence each focal element from $\mathcal{E}(F)$ is extended so that $V_{\mathcal{Z}}$ is appended to each existing assignment, and all assignments that do not agree with $V_{\mathcal{Z}}$ are also included. The weights of each focal element are unchanged.

As an example of the Ballooning Extension, consider the DSF F:

$\mathrm{Var}(F) = \{X\}$ and $\mathrm{Val}(X) = \{x_1, x_2\}$ and $\mathrm{Val}(Y) = \{y_1, y_2\}$.

Let $F = \langle\{x_1\}, 0.3\rangle; \langle\{x_1, x_2\}, 0.7\rangle$.

Extending F to cover the additional variable Y with the assignment y_1, yields the ballooning extension: $\mathbf{balloon}\,(F|y_1) = \langle\{x_1y_1, x_1y_2, x_2y_2\}, 0.3\rangle; \langle\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}, 0.7\rangle$.

The total DSF associated with variable $x \in \mathcal{X}$ is the product:

$F_x = \prod_{V_{\mathrm{Pa}(x)} \in \mathrm{Val}(\mathrm{Pa}(x))} F_{x|V_{\mathrm{Pa}(x)}} = \prod_{V_{\mathrm{Pa}(x)} \in \mathrm{Val}(\mathrm{Pa}(x))} \mathbf{balloon}\left(D_{x|V_{\mathrm{Pa}(x)}} \middle| V_{\mathrm{Pa}(x)}\right)$

This is the interpretation of a conditional DS tables given in [18]. An important property that F_x has is that if x is marginalized out, then the resultant DSF is the vacuous DS model over $\mathrm{Pa}(x)$. This means that F_x holds no information about the parents of x. We will prove this in theorem 5.10. The total DS model associated with the DS-BN is simply the product

$F = \prod_{x \in \mathcal{X}} F_x = \prod_{x \in \mathcal{X}} \prod_{V_{\mathrm{Pa}(x)} \in \mathrm{Val}(\mathrm{Pa}(x))} F_{x|V_{\mathrm{Pa}(x)}} = \prod_{x \in \mathcal{X}} \prod_{V_{\mathrm{Pa}(x)} \in \mathrm{Val}(\mathrm{Pa}(x))} \mathbf{balloon}\left(D_{x|V_{\mathrm{Pa}(x)}} \middle| V_{\mathrm{Pa}(x)}\right)$

Just as with a traditional probabilistic BN, no normalization of the weights is needed to produce the total DS model after all the factors are multiplied together. We will prove this in theorem 5.11.

Both theorems below use the following observation: Given DS models $D_1, D_2, \ldots, D_N$, the weights of the product $D_1 \times D_2 \times \cdots \times D_N$ do not need to be normalized to form a DS model if there does not exist focal elements $J_1 \in \mathcal{E}(D_1)$, $J_2 \in \mathcal{E}(D_2)$, ..., $J_N \in \mathcal{E}(D_N)$ from each D_i such that $J_1 \cap J_2 \cap \cdots \cap J_N = \emptyset$.

Theorem 5.10. *Consider an arbitrary conditional DS table over x, dependent on $\mathcal{Y}$. Let $F_x = \prod_{V_\mathcal{Y} \in \text{Val}(\mathcal{Y})} F_{x|V_\mathcal{Y}} = \prod_{V_\mathcal{Y} \in \text{Val}(\mathcal{Y})} \textbf{\textit{balloon}}\left(D_{x|V_\mathcal{Y}} \middle| V_\mathcal{Y}\right)$ be the total DSF associated with x that is generated from the conditional DS table. If F_x is marginalized to eliminate x, then the resultant DSF is simply the vacuous DS model over $\mathcal{Y}$: $\textbf{\textit{marg}}(F_x|\mathcal{Y}) = \mathbf{1}_\mathcal{Y}$.*

The proof is given in Appendix A.

Theorem 5.11. *Given an arbitrary DS-BN, no normalization of focal element weights is needed to produce the final DS model after all factors are multiplied together.*

The proof is given in Appendix A.

Inference using DS-MNs

When performing inference using DS-MNs, the technique of variable elimination can be applied. For variable elimination to work correctly, the following theorems must hold:

Theorem 5.12. *For a set of DSFs $\{F_1, F_2, \ldots, F_k\}$; evidence variables $\mathcal{Z}$ and evidence $V_\mathcal{Z}$,*

$$\textbf{\textit{cond}}\left(\prod_{i=1}^{k} F_i \middle| V_\mathcal{Z}\right) = \prod_{i=1}^{k} \textbf{\textit{cond}}(F_i|V_\mathcal{Z})$$

The proof is given in the Appendix. Theorem 5.12 implies that conditioning can be done by conditioning each DSF individually.

Theorem 5.13. *For a variable $x \in \mathcal{X}$; a DSF F_1 for which $x \in \text{Var}(F_1)$; and another DSF F_2 for which $x \notin \text{Var}(F_2)$, it is the case that:*

$$\textbf{\textit{marg}}(F_1 \times F_2|\text{Var}(F_1 \times F_2) \setminus x) = \textbf{\textit{marg}}(F_1|\text{Var}(F_1) \setminus x) \times F_2$$

This theorem is mentioned as a lemma in [38]. The proof is given in the Appendix. Theorem 5.13 implies that a variable can be marginalized out using only the factors that contain said variable.

5.5.5 About DSm models

The application of Dezert-Smarandache (DSm) models to causal networks replaces the CPTs with conditional DSm tables (CDSmTs). For probabilistic inference, each DSm model will first be reduced to a DS model from which inference on Dempster-Shafer Bayesian networks proceeds. Given a DSm model S, S is reduced to an unnormalized DS model S' via the following process. To understand this process, it should be understood that an expression J is in "conjunctive normal form" if and a only if it as the form $(L_{1,1} \cup L_{1,2} \cup ... \cup L_{1,k_1}) \cap (L_{2,1} \cup L_{2,2} \cup ... \cup L_{2,k_2}) \cap ... \cap (L_{p,1} \cup L_{p,2} \cup ... \cup L_{p,k_p})$ where each "literal" $L_{i,j}$ is a element of Val(S). Each subexpression $L_{i,1} \cup L_{i,2} \cup ... \cup L_{i,k_i}$ is a "clause" of J and corresponds to the set $\{L_{i,1}, L_{i,2}, ..., L_{i,k_i}\}$.

Start by assigning $m(J'|S') \leftarrow 0$ for each $J' \in \text{Set}(S)$.
for each $J \in \mathcal{E}(S)$ **do**
  First express J using "conjunctive normal form".
  for each clause J_c from J **do**
    Generate the subset J' of Val(S) from the literals from J_c.
    $m(J'|S') \leftarrow m(J'|S') + m(J|S)$
  end for
end for

The resultant DS model S' may *not* be normalized, but this is not significant as renormalization occurs during uncertainty inference.

5.6 Example causal network and inference using different metrics

In this section, we propose an example causal network that models an Evidence Accumulation and Risk Assessment (EA&RA) scenario, and apply different uncertainty metrics to the model.

To choose an uncertainty metric and provide the conditional uncertainty tables, the following must be considered: 1) The choice of uncertainty metric, 2) The data available to create the input data structures (CUTs), 3) The requirements on the output data structure (the uncertainty model), and 4) the heuristics driving the uncertainty inference. No one uncertainty metric can perfectly satisfy these requirements. In this thesis, we use the probabilistic metric, probabilistic intervals, the DS belief metric and its extension DSm, as well as the fuzzy metric.

The idea of the proposed approach is the uniform modeling platform which includes two components: (a) a Graphical representation of a given scenario in the form of a causal network and (b) a Mechanism of uncertainty inference in different uncertainty metrics.

Given a scenario, the approach to modeling and uncertainty inference for decision support includes the following steps:

Step 1: Represent the scenario by a causal network.

Step 2: Assign the nodes CUTs. Chose an appropriate metric (data structure) for representing uncertainty; in this chapter, we use probability models, probability interval models, DS models, DSm models, and fuzzy probability models.

Step 3: Apply the observed evidence to the causal network, and then utilize the algorithm appropriate to the chosen uncertainty metric to compute the posterior uncertainty model.

Step 4: Make a decision based on a heuristic analysis of the posterior uncertainty model. For example, if probability theory is the chosen approach to uncertainty, then the most probable outcome is the outcome with the highest posterior probability.

5.6.1 Example causal network 1

This section will present an example EA&RA scenario, modeled by a causal network, and example observations to demonstrate the modeling approach using the different metrics of uncertainty. Due to the fuzziness of the variables involved and the lack of statistical data, the numerical values (probabilities and belief values) that populate the model are chosen arbitrarily for the sake of example and are not generated from real data.

The causal network, which models the scenario of traveler ID verification/authentication at airports and other border crossing scenarios, is shown in figure 5.2. Each variable is described in table 5.2.

Table 5.2: A description of each variable in causal network 1.

Variable	Description
R	The node "ID source reliability" ($R \in \{r_1, r_2, r_3\}$) denotes the three level (r_1 ="high", r_2 ="medium", r_3 ="low") reliability of the e-passport/ID authentication, which depends on many factors such as: country of issue, number of defense levels in the document, life cycle history, type of the chip, type of biometric modality, type of encryption, and the type of RFID mechanism.
V	The node "Valid ID" ($V \in \{v_1, v_2\}$) denotes whether the e-passport ID should pass the validation procedure (valid v_1) or not (invalid v_2). The "valid" or "invalid" state reflects the true state of the e-passport and not simply the opinion of the authentication machine.
S	The node "ID scan" ($S \in \{s_1, s_2, s_3, s_4\}$) denotes the outcome of the authentication of the e-passport. The scan is subject to various unwanted effects such as the traveler's mistakes in the use of the scanning device, scanner errors, as well as hidden reasons related to errors in the use of the database, conflicts of comparisons, and communication errors or delays. These effects are encoded in the form of the number of attempts at scanning the traveler document; three attempts are allowed (s_1, s_2, s_3), after which the traveler is directed to manual control (s_4). Ideally, if the traveler's e-passport is invalid, they should always be directed to manual control.
C	The node "ID Information credibility" ($C \in \{c_1, c_2, c_3\}$) describes the three level (c_1 ="high", c_2 ="medium", c_3 ="low") credibility of the outcome of the validation process. If the credibility of the validation process is known priori, it can be used to compute posterior beliefs related to the validity of the traveler document (node V).

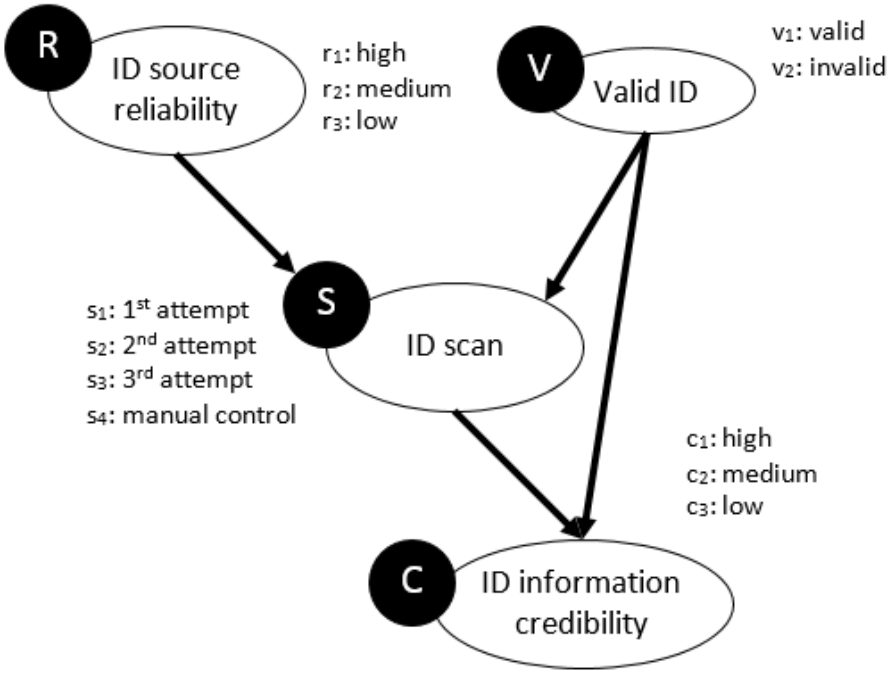


Figure 5.2: Causal network of the traveler ID validation scenario.

5.6.2 Probability measures

When probabilities are the chosen uncertainty metric, conditional probability tables (CPTs) are assigned to each node. The CPTs are given in Table 5.3. (*The probabilities and belief values in this example have been arbitrarily chosen and are not generated from real statistics.*).

The causal network in Fig. 5.2 along with the CPTs from Table 5.3 provides a tool for probabilistic inference given various scenarios of traveler ID validation.

As an example of probabilistic inference using the Bayesian realization of the causal network, consider the following scenario: `IF the reliability of the ID source is known to be 'low' and the credibility of the result to be 'high':` $R = r_3$, `and` $C = c_1$, `THEN what is the posterior probability that the ID is valid:` $\Pr(V = v_1 | R = r_3, C = c_1)$.

This scenario models a situation of conflict where an unreliable ID scanner produces a credible outcome.

Output: The calculations are performed using the software package "DS-BN_v2.exe" [5]. The final result is $\Pr(V = v_1|R = r_3, C = c_1) \approx 0.9893$. It is very likely that the ID was valid.

A measure of "conflict" present in the scenario is the prior probability of the evidence: $\Pr(R = r_3, C = c_1)$. A similar means for quantifying conflict is given in [66]. The higher the prior probability of the evidence, the less "conflict" is present. In this case: $\Pr(R = r_3, C = c_1) \approx 0.0931$. This probability is small, implying a high degree of conflict. This is what is expected since the evidence which involves a low ID source reliability, and a credible outcome, seems to be contradictory.

Table 5.3: The CPTs corresponding to the nodes of the Bayesian realization of the causal network shown in Fig. 5.2.

Node R	r_1	r_2	r_3
$Pr(R)$	0.5	0.3	0.2

Node V	v_1	v_2
$\Pr(V)$	0.99	0.01

Node S	s_1	s_2	s_3	s_4
$\Pr(S\|R = r_1, V = v_1)$	1	0	0	0
$\Pr(S\|R = r_1, V = v_2)$	0	0	0	1
$\Pr(S\|R = r_2, V = v_1)$	0.65	0.05	0.05	0.25
$\Pr(S\|R = r_2, V = v_2)$	0.05	0.05	0.15	0.75
$\Pr(S\|R = r_3, V = v_1)$	0.4	0.05	0.05	0.5
$\Pr(S\|R = r_3, V = v_2)$	0.4	0.05	0.05	0.5

Node C	c_1	c_2	c_3
$\Pr(C\|V = v_1, S = s_1)$	1	0	0
$\Pr(C\|V = v_1, S = s_2)$	0.8	0.2	0
$\Pr(C\|V = v_1, S = s_3)$	0.5	0.5	0
$\Pr(C\|V = v_1, S = s_4)$	0	0	1
$\Pr(C\|V = v_2, S = s_1)$	0	0	1
$\Pr(C\|V = v_2, S = s_2)$	0	0.2	0.8
$\Pr(C\|V = v_2, S = s_3)$	0	0.5	0.5
$\Pr(C\|V = v_2, S = s_4)$	1	0	0

5.6.3 Fuzzy probability measures

The first generalization of probability theory that will be used in this chapter are "fuzzy probabilities" [11, 53]. A fuzzy probability consists of a center value that acts as a normal probability, and a lower and upper limit that contains the center value. The interval formed

by these limits is not subject to the same requirements as the probability intervals [17]. The interval formed by the lower and upper limit is not engineered to be tight; the lower bound may be less than 0, and the upper bound may be greater than 1. The relaxation of the tightness conditions on these bounds not only improves computational complexity, but is also important given the fuzzy numbers [53] which have a triangular "membership function", with a membership of 1 at the center value, that linearly decreases to 0 at both the lower and upper bounds. A lower bound of $-\infty$ creates a membership function that is 1 for all values less than the center value; and an upper bound of $+\infty$ creates a membership function that is 1 for all values greater than the center value. Lower and upper bounds that fall outside of $[0, 1]$ may not make sense from the perspective of probabilities, but they help shape the membership function of the fuzzy number inside of $[0, 1]$.

An uncertainty radius of 0.1 is extended around each probability from Table 5.3 except to where the probability is decisively 0 or 1. This results in the fuzzy probability tables (Table 5.4). The uncertainty radius, in a manner similar to error margins, quantifies the uncertainty about the value of the point probabilities. Larger radii correspond to greater degrees of uncertainty. A fuzzy probability is denoted by (l, c, u) where c is the center value, and l and u are the lower and upper limits of the membership function, respectively.

As an example of probabilistic inference using fuzzy probabilities, the same scenario will be considered: `IF the reliability of the ID source is known to be 'low' and the credibility of the result to be 'high':` $R = r_3$, `and` $C = c_1$, `THEN what is the posterior probability that the ID is valid:` $\Pr(V = v_1 | R = r_3, C = c_1)$.

Output: The calculations are performed using theory that is described in [53]. The final result is: $\Pr(V = v_1 | R = r_3, C = c_1) = (0.1125, 0.9893, 8.1461)$. This fuzzy probability implies that within the flexibility allowed by the fuzzy probabilities in Table 5.4, the posterior point probability can be as low as 0.1125. Despite an upper bound that exceeds 1, the true posterior point probability still cannot exceed 1. The upper bound of 8.1461 simply shapes the membership function of the fuzzy posterior probability inside of the interval [0, 1].

Table 5.4: The CFPTs corresponding to the nodes of the fuzzy Bayesian network shown in Fig. 5.2.

Node R	r_1	r_2	r_3
Node $Pr(R)$	(0.4, 0.5, 0.6)	(0.2, 0.3, 0.4)	(0.1, 0.2, 0.3)

Node V	v_1	v_2
Pr(V)	(0.89, 0.99, 1.00)	(0.00, 0.01, 0.11)

Node S	s_1	s_2	s_3	s_4
$\Pr(S\|R=r_1,V=v_1)$	(1,1,1)	(0,0,0)	(0,0,0)	(0,0,0)
$\Pr(S\|R=r_1,V=v_2)$	(0,0,0)	(0,0,0)	(0,0,0)	(1,1,1)
$\Pr(S\|R=r_2,V=v_1)$	(0.55,0.65,0.75)	(0.00,0.05,0.15)	(0.00,0.05,0.15)	(0.15,0.25,0.35)
$\Pr(S\|R=r_2,V=v_2)$	(0.00,0.05,0.15)	(0.00,0.05,0.15)	(0.05,0.15,0.25)	(0.65,0.75,0.85)
$\Pr(S\|R=r_3,V=v_1)$	(0.3,0.4,0.5)	(0.00,0.05,0.15)	(0.00,0.05,0.15)	(0.4,0.5,0.6)
$\Pr(S\|R=r_3,V=v_2)$	(0.3,0.4,0.5)	(0.00,0.05,0.15)	(0.00,0.05,0.15)	(0.4,0.5,0.6)

Node C	c_1	c_2	c_3
$\Pr(C\|V=v_1,S=s_1)$	(1,1,1)	(0,0,0)	(0,0,0)
$\Pr(C\|V=v_1,S=s_2)$	(0.7,0.8,0.9)	(0.1,0.2,0.3)	(0,0,0)
$\Pr(C\|V=v_1,S=s_3)$	(0.4,0.5,0.6)	(0.4,0.5,0.6)	(0,0,0)
$\Pr(C\|V=v_1,S=s_4)$	(0,0,0)	(0,0,0)	(1,1,1)
$\Pr(C\|V=v_2,S=s_1)$	(0,0,0)	(0,0,0)	(1,1,1)
$\Pr(C\|V=v_2,S=s_2)$	(0,0,0)	(0.1,0.2,0.3)	(0.7,0.8,0.9)
$\Pr(C\|V=v_2,S=s_3)$	(0,0,0)	(0.4,0.5,0.6)	(0.4,0.5,0.6)
$\Pr(C\|V=v_2,S=s_4)$	(1,1,1)	(0,0,0)	(0,0,0)

It should also be noted that the range of values covered by the posterior fuzzy probability will always contain the probability interval produced by the probability interval Bayesian network (which in this case was [0.5742, 1]); which in turn contains the probability value produced by the Bayesian network (which in this case was 0.9893). This relationship can be expressed as $0.9893 \in [0.5742, 1] \subseteq [0.1125, 8.1461]$. These intervals describe various accuracies for the probability $\Pr(V = v_1|R = r_3, C = c_1)$.

The prior probability of the evidence, $\Pr(R = r_3, C = c_1)$, is $(0.0267, 0.0931, 0.2373)$. Note that the upper bound is significantly larger than the point probability of $\Pr(R = r_3, C = c_1)$, which is 0.0931. This implies that the conflict in the observed evidence for this scenario is "lessened" when fuzzy probabilities are used.

5.6.4 Probability interval measures

Now to the nodes in the causal network in Fig. 5.2, assign CUTs that utilize probability intervals instead of point probabilities. A "probability interval" is an interval that contains the true probability. Generally, when any quantity is denoted using an interval, a "confidence" level is assigned to that interval. The "confidence" is typically a probability that the true value of the quantity lies in the given interval. Here however, we will not consider confidence levels and instead assume that the bounds on the probability intervals are hard, as confidence levels are not used for computed point probabilities, even though there is certainly a possibility of error. Without assuming a confidence of 1 for each probability interval, the total confidence drops to 0 as the number of probability intervals increases.

We use the theory of probability intervals from [17]. The probability intervals are subsets of $[0, 1]$; the bounds of each interval are assumed to be tight enough so that for each bound, there exists a probability distribution that achieves said bound without violating any of the other bounds. A major shortcoming with the use of probability intervals is that independence and conditional independence cannot be enforced in joint probability interval distributions. Independence and conditional independence is ubiquitous in causal networks.

For the current example, an uncertainty radius of 0.1 will be extended around each probability value from Table 5.3, except to where the probability is decisively 0 or 1. This results in the CPITs from Table 5.5. The uncertainty radius, in a manner similar to error margins, quantifies the uncertainty about the value of the point probabilities. Larger radii correspond to greater degrees of uncertainty.

As an example of probabilistic inference using probability intervals, the same scenario will be considered: `IF the reliability of the ID source is known to be 'low' and the credibility of the result to be 'high':` $R = r_3$, `and` $C = c_1$, `THEN what is the posterior probability that the ID is valid:` $\Pr(V = v_1 | R = r_3, C = c_1)$.

Output: The calculations are performed using theory that is described in [17]. The final result is: $\Pr(V = v_1 | R = r_3, C = c_1) = [0.5741, 1]$. This interval implies that within the

Table 5.5: The CPITs corresponding to the nodes of the probability interval Bayesian network shown in Fig. 5.2.

Node R	r_1	r_2	r_3
Node $Pr(R)$	[0.4, 0.6]	[0.2, 0.4]	[0.1, 0.3]

Node V	v_1	v_2
$\Pr(V)$	[0.89, 1.00]	[0.00, 0.11]

Node S	s_1	s_2	s_3	s_4
$\Pr(S\|R=r_1,V=v_1)$	[1,1]	[0,0]	[0,0]	[0,0]
$\Pr(S\|R=r_1,V=v_2)$	[0,0]	[0,0]	[0,0]	[1,1]
$\Pr(S\|R=r_2,V=v_1)$	[0.55,0.75]	[0.00,0.15]	[0.00,0.15]	[0.15,0.35]
$\Pr(S\|R=r_2,V=v_2)$	[0.00,0.15]	[0.00,0.15]	[0.05,0.25]	[0.65,0.85]
$\Pr(S\|R=r_3,V=v_1)$	[0.3,0.5]	[0.00,0.15]	[0.00,0.15]	[0.4,0.6]
$\Pr(S\|R=r_3,V=v_2)$	[0.3,0.5]	[0.00,0.15]	[0.00,0.15]	[0.4,0.6]

Node C	c_1	c_2	c_3
$\Pr(C\|V=v_1,S=s_1)$	[1,1]	[0,0]	[0,0]
$\Pr(C\|V=v_1,S=s_2)$	[0.7,0.9]	[0.1,0.3]	[0,0]
$\Pr(C\|V=v_1,S=s_3)$	[0.4,0.6]	[0.4,0.6]	[0,0]
$\Pr(C\|V=v_1,S=s_4)$	[0,0]	[0,0]	[1,1]
$\Pr(C\|V=v_2,S=s_1)$	[0,0]	[0,0]	[1,1]
$\Pr(C\|V=v_2,S=s_2)$	[0,0]	[0.1,0.3]	[0.7,0.9]
$\Pr(C\|V=v_2,S=s_3)$	[0,0]	[0.4,0.6]	[0.4,0.6]
$\Pr(C\|V=v_2,S=s_4)$	[1,1]	[0,0]	[0,0]

flexibility allowed by the probability intervals in Table 5.5, the posterior probability of the ID being valid can be low as 0.5742.

It should also be noted that the posterior probability interval will always contain the probability value produced by the Bayesian network (which in this case was 0.9893).

The prior probability of the evidence, $\Pr(R = r_3, C = c_1)$, is $[0.0267, 0.2373]$. Like with point probabilities, this prior quantifies the conflict that is present in the evidence. Note that the upper bound is significantly larger than the point probability of $\Pr(R = r_3, C = c_1)$, which is 0.0931. This implies that the conflict in the observed evidence for this scenario is "lessened" when probability intervals are used.

5.6.5 DS belief measures

Consider the use of DS theory [56, 71]. The procedure will be the same except for the CUTs. The approach here uses theory from [18, 22].

The defining feature of DS theory is that probabilities/belief weights are heuristically assigned to sets of possible outcomes as opposed to single outcomes; no probability is assigned to the empty set. Any set that is assigned a non-zero probability is called a "focal element". In [18, 22], a conditional Dempster-Shafer table (CDST) is used to assign a DS model to a variable, depending on the current assignment to the variable's parents. In [58], variables themselves can be assigned not just single values, but also sets of values from the variable's original domain. This power set minus the empty set becomes the variable's new domain.

For the current example, like with fuzzy probabilities and probability intervals, an "uncertainty" of ~ 0.1 will be introduced into each probability value from Table 5.3, except to where the probability is decisively 0 or 1. 0.1 is subtracted from each probability value (which is now the weight of a singleton focal element that contains the corresponding outcome) and the total deducted weight is accounted for in a focal element that contains all possible outcomes. If a probability is already ≤ 0.1, then the singleton focal element that corresponds to that outcome is simply removed. The CDSTs are shown in Table 5.6.

As an example of probabilistic inference using the DS realization of the causal network, the same scenario is used: `IF the reliability of the ID source is known to be 'low' and the credibility of the result to be 'high':` $R = r_3$, `and` $C = c_1$, `THEN what is the posterior belief and plausibility that the ID is valid:` $\text{Bel}(V = v_1 | R = r_3, C = c_1)$, $\text{Pl}(V = v_1 | R = r_3, C = c_1)$

Output: The calculations are performed using the software package "DS-BN_v2" [5], which uses theory that is described in [22]. The final result is: $\text{Bel}(V = v_1 | R = r_3, C = c_1) \approx 0.8779$ and $\text{Pl}(V = v_1 | R = r_3, C = c_1) \approx 0.9512$. Together, these two values form the interval of probability values $[0.8779, 0.9512]$. This interval implies that it is very likely that the ID was valid. Like the probability interval Bayesian network, the DS network produces a range of probability values. Unlike probability intervals however, this range is heuristic and may not denote an objectively quantifiable value.

It should also be noted that the range of probability values produced by the DS network

Table 5.6: The CDSTs corresponding to the nodes of the DS network shown in Fig. 5.2. Pairs of focal elements and weights are denoted by $\langle B, m(B)\rangle$, where B is the focal element and $m(B)$ is the weight.

DST model over R
$\langle\{r_1\}, 0.4\rangle$; $\langle\{r_2\}, 0.2\rangle$; $\langle\{r_3\}, 0.1\rangle$; $\langle\{r_1, r_2, r_3\}, 0.3\rangle$

DST model over V
$\langle\{v_1\}, 0.89\rangle$; $\langle\{v_1, v_2\}, 0.11\rangle$

R	V	DST model over S
r_1	v_1	$\langle\{s_1, 1\}\rangle$
r_1	v_2	$\langle\{s_4, 1\}\rangle$
r_2	v_1	$\langle\{s_1\}, 0.55\rangle$; $\langle\{s_4\}, 0.15\rangle$; $\langle\{s_1, s_2, s_3, s_4\}, 0.3\rangle$
r_2	v_2	$\langle\{s_3\}, 0.05\rangle$; $\langle\{s_4\}, 0.65\rangle$; $\langle\{s_1, s_2, s_3, s_4\}, 0.3\rangle$
r_3	v_1	$\langle\{s_1\}, 0.3\rangle$; $\langle\{s_4\}, 0.4\rangle$; $\langle\{s_1, s_2, s_3, s_4\}, 0.3\rangle$
r_3	v_2	$\langle\{s_1\}, 0.3\rangle$; $\langle\{s_4\}, 0.4\rangle$; $\langle\{s_1, s_2, s_3, s_4\}, 0.3\rangle$

V	S	DST model over C
v_1	s_1	$\langle\{c_1\}, 1\rangle$
v_1	s_2	$\langle\{c_1\}, 0.7\rangle$; $\langle\{c_2\}, 0.1\rangle$; $\langle\{c_1, c_2, c_3\}, 0.2\rangle$
v_1	s_3	$\langle\{c_1\}, 0.4\rangle$; $\langle\{c_2\}, 0.4\rangle$; $\langle\{c_1, c_2, c_3\}, 0.2\rangle$
v_1	s_4	$\langle\{c_3\}, 1\rangle$
v_2	s_1	$\langle\{c_3\}, 1\rangle$
v_2	s_2	$\langle\{c_2\}, 0.1\rangle$; $\langle\{c_3\}, 0.7\rangle$; $\langle\{c_1, c_2, c_3\}, 0.2\rangle$
v_2	s_3	$\langle\{c_2\}, 0.4\rangle$; $\langle\{c_3\}, 0.4\rangle$; $\langle\{c_1, c_2, c_3\}, 0.2\rangle$
v_2	s_4	$\langle\{c_1\}, 1\rangle$

(which in this case is $[0.8779, 0.9512]$) does not necessarily contain the probability value produced by the Bayesian realization (which in this case was 0.9893).

Lastly, the prior belief and plausibility of the evidence, $\text{Bel}(R = r_3, C = c_1)$ and $\text{Pl}(R = r_3, C = c_1)$, is 0.0280 and 0.2523 respectively. This forms the interval $[0.0280, 0.2523]$. The upper bound implies a slightly smaller amount of conflict than when probability intervals are used.

Consider another scenario: `IF there is no prior information`, `THEN what is the belief and plausibility that the credibility is 'high'`: $\text{Bel}(C = c_1)$, $\text{Pl}(C = c_1)$

The final result is: $\text{Bel}(C = c_1) \approx 0.579257$ and $\text{Pl}(C = c_1) \approx 0.936249$ This example is to provide contrast with the results of DSm (Dezert-Smarandache) theory, described in section 5.6.6.

5.6.6 DSm belief measures

Dezert-Smarandache (DSm) theory [62] is a generalization of DS theory. DSm generalizes DS theory by using focal elements that are not simply sets of possible outcomes, but also intersections of possible outcomes. The realization of DSm in this paper will continue to use Dempster's rule of combination without any proportional conflict redistribution, unlike [62].

In essence, the DSm approach allows for overlap between what otherwise would be distinct outcomes. Each possible outcome acts as a "set". The set of all sets that can be built using the set operations of "union" ($\cup$) and "intersection" ($\cap$), starting with the outcomes is referred to as the "Dedekind lattice". Non-empty elements of the Dedekind lattice are what probability values are assigned to.

For example consider variable R which denotes the reliability of the ID scanner. The domain of R is r_1 : "high"; r_2 : "medium"; and r_3 : "low". The quantities "high", "medium", and "low" are not rigorously defined quantities and there may exist overlap between these categories. The following overlaps may be possible: $r_1 \cap r_2$ and $r_2 \cap r_3$. The same situation exists with variable C which denotes the credibility of the outcome of the scanning process. The domain of C is c_1 : "high"; c_2 : "medium"; and c_3 : "low". Again, the following overlaps may be possible: $c_1 \cap c_2$ and $c_2 \cap c_3$.

Since variables R and C are the only variables that are receiving overlap categories, Table 5.7 only lists the "conditional Dezert-Smarandache tables" (CDSmTs) that correspond to variables R and C. For the prior DSm model for R, the prior DS model will be used, but a mass of 0.1 will be diffused between r_1 and r_2, as well as between r_2 and r_3. For the the conditional DSm table for C, a mass of 0.1 will be diffused between c_1 and c_2, as well as between c_2 and c_3, except for where the focal element weights are 0 or 1.

As an example of probabilistic inference using the DSm realization of the belief network, the same scenario is used: `IF the reliability of the ID source is known to be 'low' and the credibility of the result to be 'high':` $R = r_3$, `and` $C = c_1$, `THEN what is the posterior belief and plausibility that the ID is valid:` $\text{Bel}(V =$

Table 5.7: The CDSmTs corresponding to the nodes of the DSm network shown in Fig. 5.2. Pairs of focal elements and weights are denoted by $\langle B, m(B)\rangle$, where B is the focal element and $m(B)$ is the weight.

DSmT model over R
$\langle r_1, 0.35\rangle$; $\langle r_1 \cap r_2, 0.1\rangle$; $\langle r_2, 0.1\rangle$; $\langle r_2 \cap r_3, 0.1\rangle$; $\langle r_3, 0.05\rangle$; $\langle r_1 \cup r_2 \cup r_3, 0.3\rangle$

V	S	DSmT model over C
v_1	s_1	$\langle c_1, 1\rangle$
v_1	s_2	$\langle c_1, 0.65\rangle$; $\langle c_1 \cap c_2, 0.1\rangle$; $\langle c_2, 0.05\rangle$; $\langle c_1 \cup c_2 \cup c_3, 0.2\rangle$
v_1	s_3	$\langle c_1, 0.35\rangle$; $\langle c_1 \cap c_2, 0.1\rangle$; $\langle c_2, 0.35\rangle$; $\langle c_1 \cup c_2 \cup c_3, 0.2\rangle$
v_1	s_4	$\langle c_3, 1\rangle$
v_2	s_1	$\langle c_3, 1\rangle$
v_2	s_2	$\langle c_2, 0.05\rangle$; $\langle c_2 \cap c_3, 0.1\rangle$; $\langle c_3, 0.65\rangle$; $\langle c_1 \cup c_2 \cup c_3, 0.2\rangle$
v_2	s_3	$\langle c_2, 0.35\rangle$; $\langle c_2 \cap c_3, 0.1\rangle$; $\langle c_3, 0.35\rangle$; $\langle c_1 \cup c_2 \cup c_3, 0.2\rangle$
v_2	s_4	$\langle c_1, 1\rangle$

$v_1|R = r_3, C = c_1)$, $\text{Pl}(V = v_1|R = r_3, C = c_1)$

Output: As described in section 5.5.5, the probability mass of the intersection between sets will be added to the probability mass of each of the sets that form the intersection. For example, the prior DSm model for R which is: $\langle r_1, 0.35\rangle$; $\langle r_1 \cap r_2, 0.1\rangle$; $\langle r_2, 0.1\rangle$; $\langle r_2 \cap r_3, 0.1\rangle$; $\langle r_3, 0.05\rangle$; $\langle r_1 \cup r_2 \cup r_3, 0.3\rangle$, will become: $\langle r_1, 0.45\rangle$; $\langle r_2, 0.3\rangle$; $\langle r_3, 0.15\rangle$; $\langle r_1 \cup r_2 \cup r_3, 0.3\rangle$, to which DS theory can be applied.

Note however, that the probability masses of this proxy DS model do not sum to 1. The calculations are performed using the software package "DS-BN_v2" [5] which uses theory described in [22]. The final result is: $\text{Bel}(V = v_1|R = r_3, C = c_1) \approx 0.8779$ and $\text{Pl}(V = v_1|R = r_3, C = c_1) \approx 0.9512$. Together, these two values form the interval of probability values $[0.8779, 0.9512]$. This interval allows the conclusion that in the context of this example, the fuzziness of the values of the domain of R and the domain of C do not impact V.

Lastly, the prior belief and plausibility of the evidence, $\text{Bel}(R = r_3, C = c_1)$ and $\text{Pl}(R = r_3, C = c_1)$, is 0.03503 and 0.2366 respectively. This forms the interval $[0.03503, 0.2366]$. The upper bound implies a slightly greater amount of conflict than when probability intervals are used.

Scenario 2 used in section 5.6.5 is used here for the purposes of comparing the results of

DS analysis with the results of DSm analysis: `IF there is no prior information`, `THEN what is the belief and plausibility that the credibility is 'high'`: Bel($C = c_1$), Pl($C = c_1$)

The final result is: Bel($C = c_1$) ≈ 0.5801 and Pl($C = c_1$) ≈ 0.9203. The interval [0.5801, 0.9203] formed by DSm theory happens to be a subset of the interval formed by DS theory which is [0.5793, 0.9362]. This interval allows the conclusion that in the context of this example, the fuzziness of the values of the domain of R and the domain of C result in a more tighter range of belief values for the outcome of $C = c_1$.

5.6.7 Results Summary

Table 5.8 and Fig. 5.3 summarize the posterior uncertainty quantities over V given the evidence $R = r_3$ and $C = c_1$. In Fig. 5.3, the vertical axis denotes the probability values from 0 to 1. The left bar for each uncertainty metric denotes the range of probability values for $V = v_1$. The right bar for each uncertainty metric denotes the range of probability values for $V = v_2$. Fuzzy probabilities however, do not denote crisp ranges; they are denoted by tapered bars that begin at the lower bound, are thickest at the center value, and end at the upper bound. The upper bound of the fuzzy probability $\Pr(V = v_1|R = r_3, C = c_1)$ is greater than 1 and is hence not shown.

Table 5.8: The posterior uncertainty quantities over V given the evidence $R = r_3$ and $C = c_1$.

Uncertainty Metric	$\Pr(v_1\|r_3, c_1)$	$\Pr(v_2\|r_3, c_1)$
Probabilities	0.9893	0.0107
Probability Intervals	[0.5742, 1.0000]	[0.0000, 0.4258]
DS Belief-Plausibility	[0.8779, 0.9512]	[0.0488, 0.1221]
DSm Belief-Plausibility	[0.8779, 0.9512]	[0.0488, 0.1221]
Fuzzy Probabilities	(0.1125,0.9893,8.1461)	(0, 0.0107, 0.7416)

From Fig. 5.3, the following conclusions can derived regarding the probability of the e-passport being valid, $\Pr(V = v_1|R = r_3, C = c_1)$:

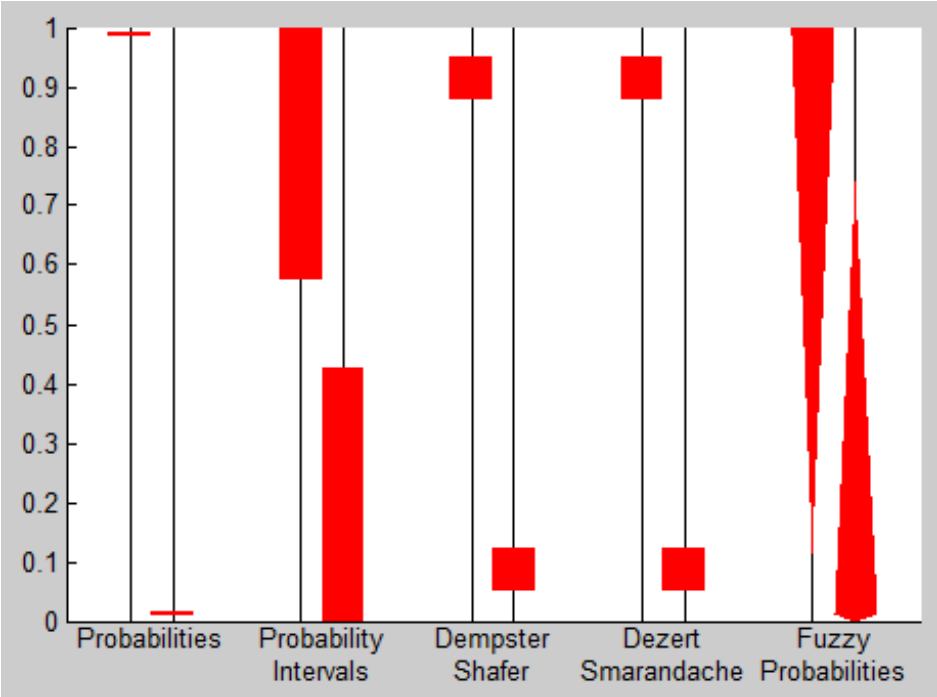


Figure 5.3: The posterior uncertainty quantities over V given the evidence $R = r_3$ and $C = c_1$. For each metric, the left column corresponds to $V = v_1$ and the right column corresponds to $V = v_2$.

- Probabilities: Accounting for no uncertainty, the probability is ~ 0.9893.
- Probability Intervals: Accounting for uncertainty in the form of rigid probability intervals, the range of probabilities is $[0.5742, 1]$.
- DS Belief-Plausibility: Allowing for Dempster-Shafer uncertainty, the range of probabilities is $[0.8779, 0.9512]$. This range lends credence to the interval $[0.5742, 1]$, but not to the point probability 0.9893.
- DSm Belief-Plausibility: Allowing for Dempster-Shafer uncertainty, as well as allowing overlap of reliability R and credibility C levels, the range of probabilities is $[0.8779, 0.9512]$. This range indicates that the fuzziness of variables R and C has no effect on the Dempster-Shafer uncertainty for this specific scenario.
- Fuzzy Probabilities: Allowing for uncertainty in the form of fuzzy probabilities, as well

as a relaxation of tightness requirements, gives a fuzzy probability of $(0.1125, 0.9893, 8.1461)$. This fuzzy probability lends credence to higher probability values.

Table 5.9 displays the result of uncertainty inference with different scenarios related to the causal network from figure 5.2. This table was computed using the software "Causal Networks and Uncertainty Metrics" [4], described in appendix D. From this table, it can be seen that the various uncertainty metrics give similar posterior values for each scenario, which helps to demonstrate the robustness of the multi-metric inference engine.

Table 5.9: The posterior uncertainty quantities computed for various scenarios related to the causal network from figure 5.2 using various uncertainty metrics.

uncertainty metric	probability	fuzzy probability	probability interval
scenario 1: $\Pr(V = v_1 \mid R = r_3, C = c_1)$	0.989255	(0.112516, 0.989255, 8.146067)	[0.574194, 1.000000]
scenario 2: $\Pr(S = s_2 \mid V = v_1, C = c_1)$	0.024768	(0.000000, 0.024768, 0.196629)	[0.000000, 0.164319]
scenario 3: $\Pr(R = r_2 \mid V = v_1, C = c_1)$	0.265635	(0.081077, 0.265635, 0.811486)	[0.106948, 0.504724]
scenario 4: $\Pr(S = s_1 \mid R = r_2, V = v_1)$	0.650000	(0.167637, 0.650000, 2.407705)	[0.256350, 0.918274]
scenario 5: $\Pr(C = c_2 \mid R = r_2, V = v_1)$	0.035000	(0.000000, 0.035000, 0.433387)	[0.000000, 0.302352]
uncertainty metric	DS: [belief, plausibility]	DSm: [belief, plausibility]	
scenario 1: $\Pr(V = v_1 \mid R = r_3, C = c_1)$	[0.877933, 0.951173]	[0.877933, 0.951173]	
scenario 2: $\Pr(S = s_2 \mid V = v_1, C = c_1)$	[0.000000, 0.235162]	[0.000000, 0.227148]	
scenario 3: $\Pr(R = r_2 \mid V = v_1, C = c_1)$	[0.182796, 0.456989]	[0.232877, 0.465754]	
scenario 4: $\Pr(S = s_1 \mid R = r_2, V = v_1)$	[0.550000, 0.850000]	[0.550000, 0.850000]	
scenario 5: $\Pr(C = c_2 \mid R = r_2, V = v_1)$	[0.000000, 0.216000]	[0.000000, 0.216322]	

A comparison of the approaches to uncertainty used in the inference engine is given in table 5.10 and table 5.11. Aspects of each approach that are described and compared include the input and output data structures, as well as the requirements for the input data, and the limitations on the output data. The key value of such comparisons for researchers is introduced via recommendations and comments.

The choice of uncertainty model is heavily dependent on the data that is available to create the CUTs, as well as the information that is expected to be given by the posterior uncertainty model. For instance, if statistical data is in abundance, probability theory will be the most suitable choice of uncertainty model, and will give the most informative results. If statistical data is lacking for certain variables, probability intervals can account for the uncertainty in those probabilities for which there is insufficient data. When bounds of the

probability intervals are 'soft', fuzzy probabilities may be used instead.

If statistical data is almost completely lacking, DS theory may be appropriate and an expert can guess at DS weights to populate the CDSTs. The key advantage of DSm theory is its ability to handle variable domains that do not contain crisp values. Values such as 'high'; 'medium'; and 'low' are not rigorously defined, and are hence 'fuzzy' and suitable for DSm theory. If there is a lack of statistical data, but point probabilities are still required for the output posterior model, a 'hybrid approach' involving point probabilities and probability intervals can be used.

5.7 Sensitivity and Technology Gap Analysis

Technology gap (TG) analysis is the process of computing goals required to achieve a desired posterior probability. Given a causal network that represents a certain scenario,

Definition 5.14. The Technology Gap posteriors (TG Posteriors) is defined as the required posterior probabilities/belief values for certain target scenarios. In the example covered in this chapter, the technology gap posteriors refers to the required maximum false match and false non-match rates of a biometric enabled e-border crossing.

Definition 5.15. The Technology Gap priors (TG Priors) is defined as the goals for the accuracy of various components (such as biomteric scanners) required to achieve the technology gap posteriors. In the example covered in this chapter, the technology gap priors refers to the target correctness rates of various elements of a biometric enabled service required to satisfy upper bounds on the false match and false non-match rates.

Definition 5.16. The TG navigator of a given scenario is defined as the process (metric, algorithm, etc.) of specifying the TG posteriors and determining the TG priors that must be met to address the scenario.

Definition 5.17. Bridging the TG gap is defined as the act of upgrading the existing technology to achieve the TG priors.

Table 5.10: Inference engine in the parallel-pipeline model: A comparison of different metrics for the causal network model.

Metric	Input data structure	Output data structure	Comments, and recommendations
Probabilities	A point probability is assigned to each possible value, variable, and assignment to the variable's parents. **Data Requirements:** Sufficient statistical data to compute every point probability with a negligible degree of error.	The posterior probability that the hypothesis of interest is true. **Limitations:** Point probabilities, out of all uncertainty structures, convey the most information.	This is the classical approach to inference using uncertainty.
Fuzzy probabilities [11, 53]	A fuzzy probability is assigned to each possible value, variable, and assignment to the variable's parents. **Data Requirements:** Similar to probability intervals, though the lower and upper bounds may be "soft".	The posterior fuzzy probability that the hypothesis of interest is true. **Limitations:** Similar to probability intervals. The presence of a center value for each fuzzy probability provides a usable point probability.	If the input fuzzy probabilities are derived by extending a lower and upper bound from input point probabilities, then the lower and upper bounds of the posterior fuzzy probabilities 1) always contain their respective point probabilities, and 2) it will also contain their respective posterior interval probabilities.
Probability intervals [17].	An interval of probabilities is assigned to each possible value, variable, and assignment to the variable's parents. **Data Requirements:** Sufficient statistical data to compute most point probabilities with a small margin of error.	An interval of posterior probabilities that the hypothesis of interest is true. **Limitations:** An interval of probability values does not indicate a specific point probability. The larger the interval, the less information is conveyed.	The intervals denote ranges within which the true probability may be found. Unlike DS theory, if the input probability intervals were derived by creating ranges around point probabilities, then posterior probability intervals always contain their respective posterior point probabilities.

5.7.1 Example causal network 2

This section will use a causal network to model a scenario where a traveler/subject is to be compared against a watchlist. Multiple images or biometric templates (referred to as "probes") are acquired from the traveler, and these templates are compared against a watchlist of suspect/high-risk individuals. The comparisons yield matching scores, which alongside a decision making strategy renders the decision to allow the traveler to pass or be intercepted.

Table 5.11: Table 5.10 continued.

Dempster-Shafer (DS) Models [56, 58].	Probability values are assigned to sets of possible values for each variable and assignment to the variable's parents. **Data Requirements:** Sufficient statistical data or expert knowledge to assign 'probabilities/belief weights' to specific sets of outcomes for each variable and assignment to the variable's parents.	An assignment of posterior belief values to sets of possible hypotheses. **Limitations:** The belief values are heuristic quantities, and DS's rule of combination is a heuristic process, and hence the range formed by the belief and plausibility may not denote an objectively quantifiable value.	Unlike probability intervals, if the belief weights were derived by introducing uncertainty to point probabilities, then the intervals formed by the posterior beliefs and plausibilities may fail to include their respective point probabilities.
Dezert-Smarandache (DSm) Models [62].	Probability values are assigned to sets built using the operations of $\cup$ and $\cap$. **Data Requirements:** Similar to DS theory. However, the domain of each variable does not have to contain crisp values.	An assignment of posterior belief values to sets of possible hypotheses. **Limitations:** Similar to DS theory. However, the fact that the variable domains may contain fuzzy values adds an extra layer to the heuristic aspect.	In addition to DS theory, DSm theory has the ability to utilize variables that do not take on discrete or well-defined values, *e.g.,* in our case, as indicated with variables R and C.

In Fig. 5.4, the TG navigator scenario is represented by a causal network where the variables are listed in table 5.12.

The initial data that populates the CUTs is real or near real. In particular, we used the performance statistic of automated border control reported in [48], as well as approximations caused by technical solutions [24, 43]. The data used for the experiment is the FRGC 2.0 database which contains 568 subjects with a total of 39,328 images. Face matching was performed using the advanced neuronet Verilook software package.

When probability distributions are used, the conditional uncertainty tables become conditional probability tables. The CPTs that are assigned to each node are listed in tables 5.13, 5.14, and 5.15:

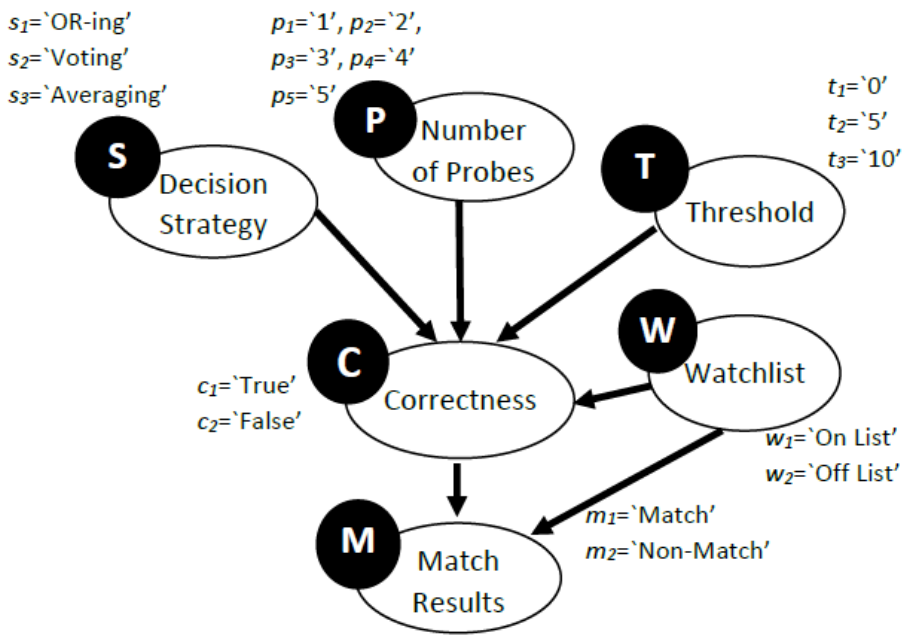


Figure 5.4: The TG navigator scenario: Specifying conditions for improving traveler risk assessment using biometric-enabled watchlist screening. The TG factors are identified in the causal network for traveler risk assessment using biometric-enabled watchlist and e-ID validation.

5.7.2 The TG formalization in terms of probabilities

Biometric-enabled watchlists are not commonly used in rapid traveler risk assessment using automated border control, except for some pilot projects. The main reason is that contemporary biometric-based profiling technologies significantly decrease the performance of automated gates. Detailed analyses of such scenarios are reported, in particular, in [43]. Specifically, technology challenges address the False Match Rate (FMR) and False Non-Match Rate (FNMR). The FMR is the probability that a match is invalid: $\Pr(w_2|m_1)$, which is the proportion of invalid matches (the traveler is not on the watchlist $W = w_2$) among all matches (the traveler is matched on the watchlist, either correctly or incorrectly $M = m_1$). The FNMR is the probability that a non-match is invalid: $\Pr(w_1|m_2)$, which is the proportion of invalid non-matches (the traveler is on the watchlist $W = w_1$) among all non-matches (the traveler is not matched to the watchlist, either correctly or incorrectly

Table 5.12: A description of each variable in causal network 2.

Variable	Description
W	The **watchlist** node denotes whether or not the traveler is **actually** on the watchlist (state w_1 ='On List') or not (state w_2 ='Off List').
T	The **threshold** node denotes a number that is assigned to a recognition system to determine whether a photo is accepted (positive) or rejected (negative). Higher thresholds yield less false acceptances but more false rejections. There are 3 thresholds that are considered by this example: t_1 denotes a threshold of '0'; t_2 denotes a threshold of '5'; and t_3 denotes a threshold of '10'.
S	The **decision strategy** node represents the methods of combining the results of matching for multiple probe images. In the s_1 =OR strategy, each probe result is returned as positive if its score is greater than the threshold, else negative. In the s_2 ='Vote' strategy, each probe casts one positive vote if its matching score is greater than the threshold, otherwise it casts a negative vote. The s_3 ='Average' strategy sums the scores of the entire probe set, and then divides it by the number of probes yielding an average score among the probe set. If the average score is greater than the threshold then the entire set is treated as positive.
P	The **number of probes** equates to the simulated number (one to five p_1, p_2, p_3, p_4, p_5) of snapshots of the user.
C	The **correctness** of the recognition system: c_1 ='True' defines the condition where the system correctly identifies a person on the watchlist (true acceptance) and correctly rejects people that are not on the watchlist (true rejection). c_2 ='False' defines the condition where the system identifies people who are not on the watchlist as being on it (false acceptance), and the wanted people as not being on the watchlist (false rejection).
M	The **matching** node represents system authentication in relation to the watchlist. m_1 denotes the scenario where a user is identified as being on the watchlist either correctly or incorrectly. m_2 denotes the scenario where a user is not identified as being on the watchlist either correctly or incorrectly.

Table 5.13: The CPTs corresponding to the node W, node T, node S, and node P in the Bayesian network in Fig. 5.4

W	Watchlist	
	'On List'	'Off List'
Pr(W)	0.0028	0.9972

T	Threshold		
	'0'	'5'	'10'
Pr(T)	$0.\overline{3}$	$0.\overline{3}$	$0.\overline{3}$

S	Decision Strategies		
	'Or'	'Vote'	'Average'
Pr(S)	$0.\overline{3}$	$0.\overline{3}$	$0.\overline{3}$

P	Number of Probes				
	'P1'	'P2'	'P3'	'P4'	'P5'
Pr(P)	0.2	0.2	0.2	0.2	0.2

$M = m_2$).

Both the FMR and FNMR affect security and privacy: in the case of a false match, an innocent person will be mistakenly directed to manual control; and a false non-match

Table 5.14: The CPT corresponding to node C of the BN in Fig. 5.4

			$\Pr(C\|W,T,S,P)$			
			W_1 = On List		W_2 = Off List	
T	S	P	'True'	'False'	'True'	'False'
'0'	'Or'	'P1'	1	0	0	1
'0'	'Or'	'P2'	1	0	0	1
'0'	'Or'	'P3'	1	0	0	1
'0'	'Or'	'P4'	1	0	0	1
'0'	'Or'	'P5'	1	0	0	1
'0'	'Vote'	'P1'	1	0	0	1
'0'	'Vote'	'P2'	1	0	0	1
'0'	'Vote'	'P3'	1	0	0.0943	0.9057
'0'	'Vote'	'P4'	1	0	0.0306	0.9694
'0'	'Vote'	'P5'	1	0	0.061	0.9390
'0'	'Average'	'P1'	1	0	0	1
'0'	'Average'	'P2'	1	0	0	1
'0'	'Average'	'P3'	1	0	0	1
'0'	'Average'	'P4'	1	0	0	1
'0'	'Average'	'P5'	1	0	0	1
'5'	'Or'	'P1'	0.9758	0.0242	0.8812	0.1188
'5'	'Or'	'P2'	0.9910	0.0090	0.8235	0.1765
'5'	'Or'	'P3'	0.9966	0.0034	0.7709	0.2291
'5'	'Or'	'P4'	0.9985	0.0015	0.7276	0.2724
'5'	'Or'	'P5'	0.9993	0.0007	0.6872	0.3128
'5'	'Vote'	'P1'	0.9758	0.0242	0.8812	0.1188
'5'	'Vote'	'P2'	0.9624	0.0376	0.8235	0.1765
'5'	'Vote'	'P3'	0.9947	0.0053	0.9441	0.0559
'5'	'Vote'	'P4'	0.9909	0.0091	0.9146	0.0854
'5'	'Vote'	'P5'	0.9973	0.0027	0.9581	0.0419
'5'	'Average'	'P1'	0.9758	0.0242	0.8812	0.1188
'5'	'Average'	'P2'	0.9831	0.0169	0.9135	0.0865
'5'	'Average'	'P3'	0.9894	0.0106	0.9294	0.0706
'5'	'Average'	'P4'	0.9931	0.0069	0.9384	0.0616
'5'	'Average'	'P5'	0.9957	0.0043	0.9429	0.0571
'10'	'Or'	'P1'	0.9545	0.0455	0.9529	0.0471
'10'	'Or'	'P2'	0.9820	0.0180	0.9273	0.0727
'10'	'Or'	'P3'	0.9920	0.0080	0.9027	0.0973
'10'	'Or'	'P4'	0.9964	0.0036	0.8805	0.1195
'10'	'Or'	'P5'	0.9980	0.0020	0.8607	0.1393
'10'	'Vote'	'P1'	0.9545	0.0455	0.9529	0.0471
'10'	'Vote'	'P2'	0.9309	0.0691	0.9273	0.0727
'10'	'Vote'	'P3'	0.9863	0.0137	0.9827	0.0173
'10'	'Vote'	'P4'	0.9785	0.0215	0.9723	0.0277
'10'	'Vote'	'P5'	0.9920	0.0080	0.9883	0.0117
'10'	'Average'	'P1'	0.9545	0.0455	0.9529	0.0471
'10'	'Average'	'P2'	0.9602	0.0398	0.9789	0.0211
'10'	'Average'	'P3'	0.9710	0.0290	0.9854	0.0146
'10'	'Average'	'P4'	0.9796	0.0204	0.9881	0.0119
'10'	'Average'	'P5'	0.9853	0.0147	0.9898	0.0102

results in entry to a country being granted to a person of interest without manual control. Conceptually, the solution to this problem is known as an improvement to the quality of biometric traits [43]. Hence, the FMR and FNMR are indicators of the TG. The problem of bridging the TG can be formulated in terms of the TG navigation over related parameters

Table 5.15: The CPT corresponding to node M of the BN in Fig. 5.4

Pr(M|W, C)		Matching	
W	C	'Match'	'Non-match'
On List	True	1	0
On List	False	0	1
Off List	True	0	1
Off List	False	1	0

and indicators.

The TG conditions specification: The threshold T, decision strategy S, and the number of probes P will be assumed to be arbitrarily fixed.

The TG posteriors: We require that the FMR ($\Pr(w_2|T, S, P, m_1)$) and the FNMR ($\Pr(w_1|T, S, P, m_2)$) be at most 10%.

The TG priors: We will be interested in:

1. the correctness rate for watchlist travelers $x_1 = \Pr(c_1|w_1, T, S, P)$, and

2. the correctness rate for non-watchlist travelers $x_2 = \Pr(c_1|w_2, T, S, P)$

that will achieve a desired FMR $y_1(x_1, x_2) = \Pr(w_2|T, S, P, m_1)$ of at most 10%, and a FNMR $y_2(x_1, x_2) = \Pr(w_1|T, S, P, m_2)$ of at most 10%. These correctness rates x_1 and x_2 establish the TG that must be cleared to achieve the desired FMR and FNMR.

The uncertainty inference is performed using a software package [4] that is available upon request. The tables below contain the necessary data to calculate y_1, y_2 from x_1, x_2:

(a)

	Watchlist	
W	'On List'	'Off List'
Pr(W)	0.0028	0.9972

(b)

	Correctness	
C	'True'	'False'
$\Pr(C\|w_1, T, S, P)$	x_1	$1 - x_1$
$\Pr(C\|w_2, T, S, P)$	x_2	$1 - x_2$

(c)

$Pr(M\|W, C)$		Matching	
W	C	'Match'	'Non-match'
'On List'	'True'	1	0
'On List'	'False'	0	1
'Off List'	'True'	0	1
'Off List'	'False'	1	0

The (x_1, x_2) pairs of interest are those such that $y_1(x_1, x_2) \leq 0.1$ and $y_2(x_1, x_2) \leq 0.1$. The set/region of (x_1, x_2) pairs where $y_1(x_1, x_2) \leq 0.1$ will be referred to as R_1, and the set/region of (x_1, x_2) pairs where $y_2(x_1, x_2) \leq 0.1$ will be referred to as R_2. To determine an optimal x_1 and x_2 a recursive binary search will be used. While it is known that regions R_1 and R_2 can be exactly computed by solving a linear system of equations (such as in [14, 44]), the binary search presented here can and will be generalized to other non-probabilistic uncertainty models. The following table lists $y_1 = \Pr(w_2|T, S, P, m_1)$ and $y_2 = \Pr(w_1|T, S, P, m_2)$ given various values for $x_1 = \Pr(c_1|w_1, T, S, P)$ and $x_2 = \Pr(c_1|w_2, T, S, P)$. Initially, x_1 and x_2 will be restricted to the set of values $\{0.0, 0.5, 1.0\}$.

x_1	x_2	y_1	y_2
0.00	0.00	1.000000	1.000000
0.00	0.50	1.000000	0.005584
0.00	1.00	0/0	0.002800
0.50	0.00	0.998598	1.000000
0.50	0.50	0.997200	0.002800
0.50	1.00	0.000000	0.001402
1.00	0.00	0.997200	0/0
1.00	0.50	0.994416	0.000000
1.00	1.00	0.000000	0.000000

$y_1(x_1, x_2)$ and $y_2(x_1, x_2)$ are monotone decreasing with respect to x_1 and x_2. Any range $[x_{1,L}, x_{1,U}] \times [x_{2,L}, x_{2,U}]$ such that $y_1(x_{1,U}, x_{2,U}) \leq 0.1$ will be partially contained by R_1, and if $y_1(x_{1,L}, x_{2,L}) \leq 0.1$, will be completely contained by R_1. The same is true with y_2 and R_2.

Decision: We observe that

1. the ranges $[0.00, 0.50] \times [0.50, 1.00]$ and $[0.50, 1.00] \times [0.50, 1.00]$ are partially contained by R_1.

2. the ranges $[0.00, 0.50] \times [0.50, 1.00]$ and $[0.50, 1.00] \times [0.50, 1.00]$ are completely contained by R_2, and $[0.50, 1.00] \times [0.00, 0.50]$ is partially contained by R_2.

After a range of interest has been identified as the location of the TG priors x_1 and x_2, the range can be further divided into 4 subregions that can be explored recursively. If TG goals x_1 and x_2 are chosen from a square that is

1. completely contained by R_2, then the FNMR is guaranteed to be at most 0.1.

2. partially contained by R_2, then the FNMR may be at most 0.1 but this not guaranteed.

After the TG priors x_1 (the desired correctness for travelers on the watchlist) and x_2 (the desired correctness for travelers not on the watchlist) has been chosen to improve the FMR and FNMR to at most 10%, the next problem is improving the existing technology to span the chosen TG. Such improvements may include adjustments to the deep learning approach to train the feature extractor, and hardware and infrastructure improvements such as better lighting, the use of infrared cameras etc.

5.7.3 The TG formalization in terms of fuzzy probabilities

Using the same TG example, the TG priors, $x_1 = \Pr(c_1|w_1, T, S, P)$ and $x_2 = \Pr(c_1|w_2, T, S, P)$, and the TG posteriors FMR and FNMR, $y_1(x_1, x_2) = \Pr(w_2|T, S, P, m_1)$ and $y_2(x_1, x_2) = \Pr(w_1|T, S, P, m_2)$ respectively, are now triangular fuzzy numbers. The tables below contain the necessary data to calculate y_1, y_2 from x_1, x_2:

(a)

W	Watchlist	
	'On List'	'Off List'
$\Pr(W)$	(0,0.0028,0.1028)	(0.8972,0.9972,1)

(b)

C	Correctness	
	'True'	'False'
$\Pr(C\|w_1, T, S, P)$	x_1	$(1,1,1) - x_1$
$\Pr(C\|w_2, T, S, P)$	x_2	$(1,1,1) - x_2$

(c)

$Pr(M\|W, C)$		Matching	
W	C	'Match'	'Non-match'
'On List'	'True'	(1,1,1)	(0,0,0)
'On List'	'False'	(0,0,0)	(1,1,1)
'Off List'	'True'	(0,0,0)	(1,1,1)
'Off List'	'False'	(1,1,1)	(0,0,0)

Decision:

Table 5.16: The TG in terms of fuzzy posteriors.

x_1	x_2	y_1	y_2
(0.00,0.00,0.50)	(0.00,0.00,0.50)	$(0.43, 1.00, 2.23)$	$(0.00, 1.00, +\infty)$
(0.00,0.00,0.50)	(0.00,0.50,1.00)	$(0.00, 1.00, +\infty)$	$(0.00, 0.01, +\infty)$
(0.00,0.00,0.50)	(0.50,1.00,1.00)	$(0.00, 0/0, +\infty)$	$(0.00, 0.00, 0.23)$
(0.00,0.50,1.00)	(0.00,0.00,0.50)	$(0.41, 1.00, 2.23)$	$(0.00, 1.00, +\infty)$
(0.00,0.50,1.00)	(0.00,0.50,1.00)	$(0.00, 1.00, +\infty)$	$(0.00, 0.00, +\infty)$
(0.00,0.50,1.00)	(0.50,1.00,1.00)	$(0.00, 0.00, +\infty)$	$(0.00, 0.00, 0.23)$
(0.50,1.00,1.00)	(0.00,0.00,0.50)	$(0.41, 1.00, 2.23)$	$(0.00, 0/0, +\infty)$
(0.50,1.00,1.00)	(0.00,0.50,1.00)	$(0.00, 0.99, +\infty)$	$(0.00, 0.00, +\infty)$
(0.50,1.00,1.00)	(0.50,1.00,1.00)	$(0.00, 0.00, +\infty)$	$(0.00, 0.00, 0.12)$

1. The (x_1, x_2) pairs of interest are those that yield a pair (y_1, y_2) that satisfies conditions $f_1(y_1)$ and $f_2(y_2)$.

2. $f_1(y_1)$ is an unknown condition which holds when y_1 is "small". $f_2(y_2)$ is an unknown condition which holds when y_2 is "small".

3. The set/region of (x_1, x_2) pairs where $f_1(y_1(x_1, x_2))$ holds will be referred to as R_1, and the set/region of (x_1, x_2) pairs where $f_2(y_2(x_1, x_2))$ holds will be referred to as R_2.

Search techniques similar to those used in the case of point probabilities can be used. Table 5.16 lists $y_1 = \Pr(w_2|T, S, P, m_1)$ and $y_2 = \Pr(w_1|T, S, P, m_2)$ given various values for $x_1 = \Pr(c_1|w_1, T, S, P)$ and $x_2 = \Pr(c_1|w_2, T, S, P)$. x_1 and x_2 are restricted to the set of values $\{(0.0, 0.0, 0.5), (0.0, 0.5, 1.0), (0.5, 1.0, 1.0)\}$. These 3 values are fuzzy versions of the values $\{0.0, 0.5, 1.0\}$.

After the TG priors have been determined, the same approaches can be used to bridge the TG as when point probabilities were used. However, because fuzzy probabilities are being used, the requirements set by the TG are not as strict as when point probabilities are being used. It is also important note that using fuzzy probabilities to navigate the TG is not the same as sensitivity analysis. Sensitivity analysis aims to determine the impact of uncertainty, which is distinct from navigating the TG which aims to compute the TG priors while accounting for uncertainty.

5.8 Conclusion

This chapter has described the application of various non-probabilistic interpretations of uncertainty to causal networks. These non-probabilistic interpretations include: fuzzy probabilities, probability intervals, Dempster-Shafer models, and Dezert-Smarandache models. One contribution of this chapter is the unification of the these various uncertainty models and their application to causal networks into a unified framework that enables flexibility in the input statistical data as listed in table 5.10. This is demonstrated in the example given in section 5.6.

In section 5.7, the use of causal networks with various uncertainty metrics are utilized to analyze the "technology gaps" in a scenario related to watchlist authentication. Target bounds for the false match rate and the false non-match rate are set. Simple analysis that uses uncertainty inference with the causal network derives accuracy goals that must be met by the biometric system to achieve the set bounds of the false match rate and the false non-match rate.

Lastly, future work can focus on the integration of various uncertainty metrics simultaneously into a causal network. When uncertainty metrics are used simultaneously, each node in the causal network may be assigned a CUT that uses a metric that is different from other nodes, and a node may even be assigned a "mixed CUT" where various uncertainty metrics are present in the same table.

Chapter 6

A Taxonomy and Analysis of Information Fusion Approaches

6.1 Introduction

The problem of "fusion" stems from the need to combine information from various sources. Each of these source is assumed to provide either an observation, or a "model of uncertainty". In section 3.4 the fusion of abstract real valued risk quantities was described. In this chapter, the information being fused is still related to risk. "Context specific fusion" can fuse risk expressed by any structure, while "general fusion" will require that the risk take the form of a convex set of probability distributions known as a "credal set".

Table 6.1 compiles a short list of existing research on the process of fusing observations, belief values, or other metrics into a single quantity that can be utilized for decision support.

This chapter will utilize convex sets of probability distributions, known as credal sets, to form the model of uncertainty in probability distributions. However, unlike most approaches to credal sets which maintain a list of the extreme points, this chapter will focus on "subtypes" of credal sets, namely probability interval distributions and Dempster-Shafer models (Dempster-Shafer models can also describe credal sets in addition to heuristic belief func-

Table 6.1: A description of existing work related to information fusion.

Reference(s)	Contribution
[18]	The transferable belief model.
[72]	Compatibility Relationships: Compatibility functions are functions that determine the "closeness" or "compatibility" between two outcomes. Compatibility functions are used to determine the most "likely" outcome from a set of seemingly contradictory input outcomes.
[73]	Uninorm Aggregation: Functions referred to as "uninorm"s were analyzed as candidates for combining several strengths of belief into a single strength of belief. A "uninorm" is a function for combing values from the range [0,1] into a single value from the range [0,1]. In a uninorm function, a specific value that serves as the identity can be arbitrarily chosen and the function built around it.
[32, 33, 34]	The fusion of credal sets as models of uncertainty and the establishment that fusion can be performed in an efficient and exact manner when credal sets are denoted by listing their extreme points.
[71, 75]	Dempster-Shafer combination/fusion (described in section 3.3.4)
[19, 66, 64]	Dezert-Smarandache fusion.
[36]	A survey of contemporary fusion approaches, most of which rely on quantities such as "strengths of belief" that are highly subjective. In addition, the heuristics used to handle strengths of belief are algorithms designed so that the outputs "make sense" as opposed to obeying an objective criteria.
[10]	The creation of a software package that calculates posterior credal sets such as "CREDO".

tions, as will be discussed in section 6.6). The fusion of credal sets as models of uncertainty is described in [32, 33, 34], and it is already established that fusion can be performed in an efficient and exact manner when credal sets are denoted by listing their extreme points. It is known however, that credal set subtypes such as probability interval distributions and Dempster-Shafer models can describe certain credal sets using dramatically less data than the number of extreme points (see [65, 81] for a discussion on the number of extreme points of probability interval distributions). When a credal set is the set of probability distributions denoted by a specific probability interval distribution or Dempster-Shafer model, the probability interval distribution or Dempster-Shafer model is the more efficient representation in terms of space (and subsequently computational complexity). This is the prime motivation for using the credal set subtypes of probability interval distributions and Dempster-Shafer models, and gives practical purpose to the catalog of fusion approaches presented in this chapter.

The use of intervals to describe probabilities is formally developed in [17] and [37, chapter 5] and was described in detail in section 3.3.3.

Dempster-Shafer (DS) theory is described in [37, chapter 5], [71] and section 3.3.4. In many publications such as [76], Dempster-Shafer models are interpreted as follows: the belief and plausibility respectively form lower and upper bounds for the true probability. This is the interpretation of Dempster-Shafer theory that will be used in this chapter. With this interpretation, Dempster-Shafer models effectively denote a credal set. Dempster's rule of combination, described in [37, chapter 5] and [71], is a popular approach to fusing Dempster-Shafer models. There is however, a major inconsistency with Dempster's rule of combination when Dempster-Shafer models are interpreted as credal sets: the fused Dempster-Shafer model does not describe a credal set that contains all possible probability distributions that result from fusing probability distributions chosen from the credal set of each input Dempster-Shafer model (see definition 6.3). This inconsistency is described in section 6.6.4.

Many approaches to fusion using Dempster-Shafer models focus on "redistributing con-

flict" such as from [21, 60, 61]. "Conflict redistribution" focuses on minimizing or eliminating the renormalization that occurs when Dempster-Shafer models are combined/fused. This chapter, due to its focus on credal sets, will not focus on approaches such as conflict redistribution, since these approaches do not treat Dempster-Shafer models as credal sets.

An important aspect of this chapter is a look at the various algorithms that fuse credal sets. Existing work on this topic include the known Bayesian fusion of credal sets from [33], the calculation of posterior probability intervals from [17, 68, 9], and the creation of software packages that calculate posterior credal sets such as "CREDO" [10]. This chapter will propose a catalog of fusion approaches that utilize probability interval distributions and Dempster-Shafer models. This taxonomy will include existing work and algorithms generated specifically for this chapter.

The structure of this chapter is as follows: Section 6.3 will review two different modes of fusion using point probability distributions. Section 6.4 will review the requirements for fusion involving sets of probability distributions. Section 6.5 will cover the use of probability intervals. Section 6.6 will cover the use of Dempster-Shafer models, and propose an alternative to Dempster's rule of combination.

6.2 Contributions

The contributions of this chapter are:

- The most important contribution of this chapter is a taxonomy and catalog of fusion approaches and algorithms that utilize "subtypes" of credal sets, in this case "probability interval distributions" and "Dempster-Shafer models". All fusion approaches will satisfy an important objective criteria, referred to in this chapter as the "containment property". Special attention is paid to the computational challenges involved. Various approaches are given, which exhibit trade-offs between accuracy and computational complexity. Some of the fusion approaches are already known to the literature (such

as context specific fusion with probability intervals described in [68]), and others were created specifically for this chapter.

- Credal sets are a known highly expressive approach to denoting sets of probability distributions. The utility of probability interval distributions and Dempster-Shafer models as special subtypes of credal sets is demonstrated using the criteria of space (memory) complexity (see sections 6.5.1 and 6.6.1). This in turn motivates use of the algorithms cataloged by this chapter.

- A proposed objective criteria for information fusion referred to as the "containment property" (see section 6.4 for the definitions) is given. Dempster's rule of combination is shown to violate the containment property.

- A distinction is made between two types of information fusion, referred to as "context specific" and "general fusion". Each type of fusion has different information requirements, and the algorithms are different. Context specific fusion requires more prior information, but is less computationally intensive than general fusion. The important distinction between context specific fusion and general fusion is that context specific fusion only requires raw observations as input, while general fusion requires complete credal sets. Context specific fusion follows the hypothesis-observation models used in publications such as [18, 81] and [68, section 4, calculus], and the algorithms are generally polynomial time with respect to the size of the input. General fusion is similar to the direct Bayesian fusion of credal sets. While the direct Bayesian fusion of credal sets can be performed exactly in polynomial time [33, Theorem 2], when credal sets are restricted to specific subtypes, general fusion becomes much more difficult.

6.3 Background

In the literature, convex sets of probability distributions are referred to as "credal sets". Specific "subtypes" of credal sets that will be the focus of investigation include "probability interval distributions" and "Dempster-Shafer models".

The definition of credal sets and the notation related to credal sets can be found in section 3.3.2.

In addition, pseudo code will be used to describe various algorithms. Comments in the pseudo code are denoted using a double forward slash: `//`, or are enclosed by: `/* ... */`.

6.3.1 Two Approaches to Fusion

In this chapter, fusion will occur within the context of trying to identify an object using information gathered by remote sensors.

There are two fusion problems that will be considered by this chapter:

Problem 6.1. Context Specific Fusion: Consider a variable of interest, hypothesis variable H. Given several ($N \geq 1$) observations $O_1, O_2, \ldots, O_N$, we wish to generate a "posterior" credal set $S_\bullet = \mathbf{F}(O_1, O_2, \ldots, O_N)$ that covers the hypothesis variable H. $S_\bullet$ should consolidate all of the observations $O_1, O_2, \ldots, O_N$. The hypothesis variable, H, will be assumed to have $M \geq 2$ possible values, denoted by: $1, 2, \ldots, M$. The subtype of the posterior credal set will be the same as the subtype of the credal set that is the "prior" for H.

General Fusion: Consider a variable of interest, hypothesis variable H. Given several ($N \geq 2$) credal sets $S_1, S_2, \ldots, S_N$ that cover the hypothesis variable H, we wish to generate a "posterior" credal set $S_\bullet = \mathbf{F}(S_1, S_2, \ldots, S_N)$ that covers H and consolidates all of the information from $S_1, S_2, \ldots, S_N$. The hypothesis variable, H, will be assumed to have $M \geq 2$ possible values, denoted by: $1, 2, \ldots, M$. The subtype of the posterior credal set will be the same as the subtype of the input credal sets.

The process of context specific fusion is shown in figure 6.1(a). Observations $O_1, O_2, \ldots, O_N$

are acquired from various sources: in this case, the sources are sensors. Alongside existing data in the form of a prior credal set for H and probability ranges for each observation given each possible value of H, the observations are fused to produce a "posterior" credal set that describes H. This is the approach to fusion used in [18, 81] and [68, section 4, calculus].

The process of general fusion is shown in figure 6.1(b). "Prior" credal sets $S_1, S_2, \ldots, S_N$ are acquired from various sources: in this case, the sources are sensors. The credal sets are fused to produce a "posterior" credal set that describes H. Unlike context specific fusion, general fusion does not require existing data. Despite this however, each sensor must be equipped with the capacity to return a credal set about H, as opposed to raw data and observations. This is the approach to fusion used in many publications such as [25, 33, 73, 78].

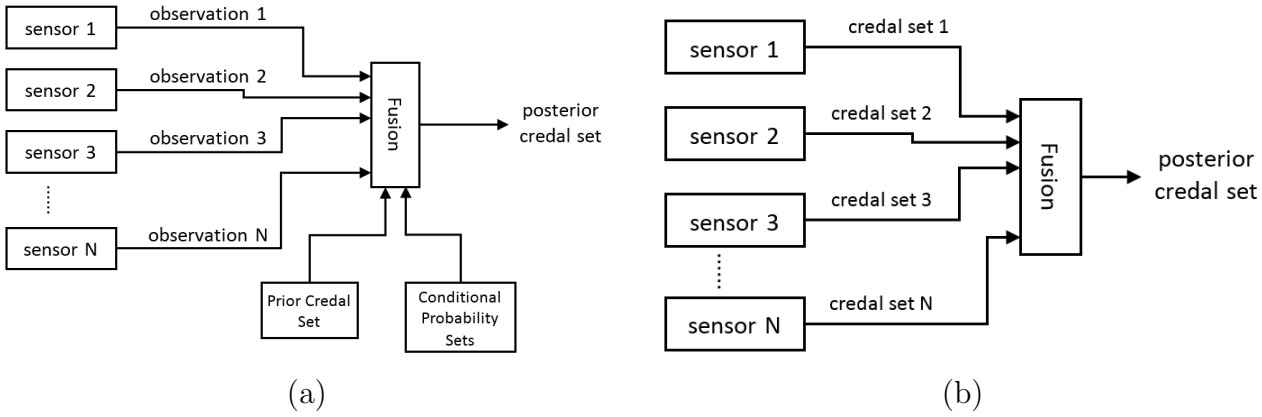


Figure 6.1: (a) The process of Context Specific Fusion. (b) The process of General Fusion.

When the probability distributions in the conditional probability tables in a Bayesian network are replaced with credal sets, the result is a "credal network". Theory related to credal networks can be found in [15]. An important concept related to credal networks is the concept of the "strong extension", which is the tightest convex hull that contains all joint probability distributions allowed by the credal network.

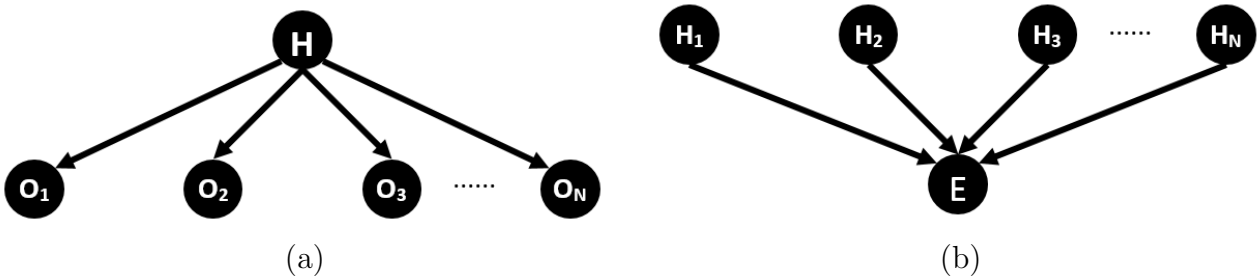
The fact that in "causal networks", the manner of a child node's dependency on its parent does not need to be specified, means that causal networks can include Bayesian networks and credal networks.

A causal network provides a concrete high level model of the scenario of interest. In the

case of fusion, there will be a distinct causal network for each fusion approach.

Figure 6.2(a) displays the causal network that describes the scenario used for context specific fusion. In this scenario, the hypothesis variable H influences each of the observations $O_1, O_2, \ldots, O_N$. For each possible hypothesis, the observations all occur independently (there are no causal links between observations).

Figure 6.2(b) displays the causal network that describes the scenario used for general fusion. Unlike the causal network for context specific fusion, there are instead N hypothesis variables $H_1, H_2, \ldots, H_N$ that correspond to each of the N credal sets $S_1, S_2, \ldots, S_N$. The hypothesis variables are all independent (there are no causal links between the H_i's). There is then a binary variable E which attains 1 if and only if all of the hypothesis variables are equal and 0 if otherwise.



(a) (b)

Figure 6.2: (a) The causal network that describes the scenario envisioned for context specific fusion. (b) The causal network that describes the scenario envisioned for general fusion.

During context specific fusion, the observation variables $O_1, O_2, \ldots, O_N$ are fixed to their observed values, and the posterior credal set for H is computed. During general fusion, the binary variable E is fixed to 1 which forces all of the hypothesis variables to have the same value. The posterior credal set that describes this common value is the resultant posterior credal set for H.

6.3.2 Context Specific Fusion using point probabilities

This section describes the process of context specific fusion when the credal sets consist of single probability distributions. A credal set that contains a single probability distribution

is referred to as a "point probability distribution". Before any fusion can occur, a "prior" probability distribution for H is required. Let the prior probability of $H = j$ for each $j = 1, 2, \ldots, M$ be denoted by p_j. In addition, for each O_i $(i = 1, 2, \ldots, N)$, an observation o_i is received. For each $j = 1, 2, \ldots, M$, $p_{i,j}$ will denote the probability of $O_i = o_i$ provided that $H = j$: $p_{i,j} = \Pr(O_i = o_i | H = j)$. Bayes' rule gives gives the following posterior probability distribution for H:

$$\forall j' \in \{1, 2, \ldots, M\} : \Pr(H = j' | \forall i \in \{1, 2, \ldots, N\} : O_i = o_i) = \frac{p_{j'} \prod_{i=1}^{N} p_{i,j'}}{\sum_{j=1}^{M} p_j \prod_{i=1}^{N} p_{i,j}}$$

As an example of context specific fusion, consider a scenario related to biometric authentication. An individual's identity is to be verified by three ($N = 3$) biometric sensors: face, iris, and fingerprint. There are 2 possible states ($M = 2$) of the individual, which is denoted by the hypothesis variable H: The individual is genuine, $H = 1$; or an impostor, $H = 2$. The $N = 3$ biometric sensors that verify the individual's identity are: the face sensor (O_1); the iris sensor (O_2); and the fingerprint sensor (O_3). Each sensor can return either "match" or "no match". Through careful experimentation, it is known that:

$$\Pr(H = 1) = 0.9 \qquad \Pr(H = 2) = 0.1$$
$$\Pr(O_1 = \text{“match”} | H = 1) = 0.9 \qquad \Pr(O_1 = \text{“match”} | H = 2) = 0.4$$
$$\Pr(O_2 = \text{“no match”} | H = 1) = 0.3 \qquad \Pr(O_2 = \text{“no match”} | H = 2) = 0.6$$
$$\Pr(O_3 = \text{“match”} | H = 1) = 0.7 \qquad \Pr(O_3 = \text{“match”} | H = 2) = 0.2$$

If the face sensor returns $O_1 =$ "match"; the iris sensor returns $O_2 =$ "no match"; and the fingerprint sensor 3 returns $O_3 =$ "match"; then the posterior probability distribution

for H is:

$$\Pr(H = 1|O_1, O_2, O_3) = \frac{(0.9)(0.1)(0.3)(0.7)}{(0.9)(0.1)(0.3)(0.7) + (0.1)(0.6)(0.6)(0.2)} \approx 0.7241$$

$$\Pr(H = 2|O_1, O_2, O_3) = \frac{(0.1)(0.6)(0.6)(0.2)}{(0.9)(0.1)(0.3)(0.7) + (0.1)(0.6)(0.6)(0.2)} \approx 0.2759$$

It should also be noted that observations can be fused in a sequential fashion. For example, the observations $O_1, O_2, \ldots, O_N$ can be fused simultaneously with one large fusion step, but it is also possible to fuse the observations in a sequential fashion. This sequential fusion proceeds as follows. Let $\Pr_0$ be the prior probability distribution of H. Now fuse the single observation O_1 to get the posterior probability distribution $\Pr_1$. To fuse on observation O_2, the prior distribution $\Pr_0$ for H should be replaced with $\Pr_1$, and then the single observation O_2 should be fused using the new prior. This process continues until all of $O_1, O_2, \ldots, O_N$ have been fused. Fusing observations in a sequential manner also provides a means of performing context specific fusion with a computational complexity of $O(N)$. The computational complexity's dependence on M, the domain size of the hypothesis variable, depends on the subtype of credal set used. In the case of point probabilities however, the computational complexity with respect to M is $O(M)$. The overall computational complexity for fusing point probability distributions in a context specific manner is $O(NM)$.

6.3.3 General Fusion using point probabilities

This section describes the general fusion process when the credal sets consist of single probability distributions. For now, assume that each S_i is a single probability distribution with respective probabilities $p_{i,1}, p_{i,2}, \ldots, p_{i,M}$, where $p_{i,j}$ is the prior probability that $H_i = j$. With probability distributions, it is required that $\sum_{j=1}^{M} p_{i,j} = 1$. Since it is required that $H_1 = H_2 = \cdots = H_N (= H)$, we know that $E = 1$. Bayes' rule gives gives the following

posterior probability distribution for H:

$$\forall j' \in \{1, 2, \ldots, M\} : \Pr(H_1 = j' | H_1 = H_2 = \cdots = H_N) = \frac{\prod_{i=1}^{N} p_{i,j'}}{\sum_{j=1}^{M} \prod_{i=1}^{N} p_{i,j}}$$

As an example of general fusion, consider the same biometric verification scenario from the context specific fusion example: a subject that can be one of $M = 2$ states: "genuine" ($H = 1$); or "impostor" ($H = 2$). Again there are $N = 3$ biometric sensors, face, iris, and fingerprint to verify the subject's identity, but instead of these sensors simply returning a direct observation, they instead return their own guess at the probability distribution for H.

Assume that sensor 1 returns a 45% probability of $H_1 = 1$ (the subject is genuine); sensor 2 returns a 60% probability of $H_2 = 1$; and sensor 3 returns a 10% probability of $H_3 = 1$. Since it is known that $H_1 = H_2 = H_3$ (which is equivalent to requiring that $E = 1$), the posterior probability distribution for H, the common value, is:

$$\Pr(H = 1) = \frac{\Pr(H_1 = H_2 = H_3 = 1)}{\Pr(H_1 = H_2 = H_3)} = \frac{(0.45)(0.6)(0.1)}{(0.45)(0.6)(0.1) + (0.55)(0.4)(0.9)} = 0.12$$

$$\Pr(H = 2) = \frac{\Pr(H_1 = H_2 = H_3 = 2)}{\Pr(H_1 = H_2 = H_3)} = \frac{(0.55)(0.4)(0.9)}{(0.45)(0.6)(0.1) + (0.55)(0.4)(0.9)} = 0.88$$

There is additional complexity to the sensors since they now have to return probability distributions as opposed to raw data. Unlike context specific fusion however, prior probability values and conditional probabilities do not have to be accumulated ahead of time (except possibly for the purpose of "calibrating" each sensor to return probabilities).

Similar to context specific fusion, credal sets can also be fused in a sequential fusion. Let credal sets $S_1, S_2, \ldots, S_N$ cover the hypothesis variable H. $S_1, S_2, \ldots, S_N$ can be fused in a single large fusion step, but it is also possible to fuse these credal sets in a sequential fashion as follows: S_1 and S_2 are fused to form S'_2; then S'_2 and S_3 are fused to form S'_3; and so on.

Again, sequential fusion allows general fusion to proceed with a computational complexity of $O(N)$ with respect to N. The computational complexity with respect to M depends on the subtype of credal set used, but for point probabilities the computational complexity is again $O(M)$ with respect to M. The overall computational complexity for fusing point probability distributions in a general manner is $O(NM)$.

The following sections will now focus on fusion where the credal set subtype is a set of probability distributions, as opposed to a single probability distribution.

6.4 Fusion using nontrivial credal sets

This section will describe both context specific and general fusion using credal sets that denote sets of probability distributions as opposed to single probability distributions. These credal sets are referred to as being "nontrivial".

The basic idea of generalizing fusion to nontrivial credal sets, is that the output credal set should contain every possible probability distribution that results from fusion using probability distributions chosen from each input credal set. Ideally, the output credal set should denote as small a set as possible. Details related to context specific and general fusion are given in the next sections.

6.4.1 Context Specific Fusion using Nontrivial Credal Sets

A high level description of the process of context specific fusion using credal sets can be found in [81, 33].

For context specific fusion, a "prior" credal set of the chosen subtype S_0 is needed that describes H. For each observation O_i $(i = 1, 2, \ldots, N)$, let o_i denote the value assigned to O_i. For each $j = 1, 2, \ldots, M$, let $P_{i,j}$ denote the set of all possible values of $\Pr(O_i = o_i | H = j)$. The resultant credal set $S_\bullet$ should now satisfy the following **containment property**:

Definition 6.2. The Containment Property for Context Specific Fusion:

First, choose an arbitrary probability distribution $p_1, p_2, \ldots, p_M$ from S_0. For each $i = 1, 2, \ldots, N$, and $j = 1, 2, \ldots, M$ consider an arbitrary choice of probability $p_{i,j}$ where $p_{i,j} \in P_{i,j}$. The probability distribution given by:

$$\forall j' = 1, 2, \ldots, M : p_{\bullet,j'} = \frac{p_{j'} \prod_{i=1}^{N} p_{i,j'}}{\sum_{j=1}^{M} p_j \prod_{i=1}^{N} p_{i,j}}$$

should now be contained by the posterior credal set $S_\bullet$, no matter the choice of p_j's and $p_{i,j}$'s. $S_\bullet$ is considered "tight" if no other probability distributions are contained. $S_\bullet$ is considered "maximally tight" if there is no other credal set of the *same subtype* $S'_\bullet$ that satisfies the containment property and is a proper subset of $S_\bullet$.

Like with point probabilities, context specific fusion using nontrivial credal sets can proceed in a sequential manner. However, the resultant credal set may not be as tight as the credal set that results from simultaneous fusion.

It is also important to note that **the cost of acquiring each $P_{i,j}$ will not be counted as part of our analysis of the computational complexity** of various algorithms for context specific fusion.

6.4.2 General Fusion using Nontrivial Credal Sets

General fusion using credal sets is described in [33].

For general fusion, the resultant credal set $S_\bullet$ should satisfy the following **containment property**:

Definition 6.3. The Containment Property for General Fusion:

Choose arbitrary probability distributions $\mathrm{Pr}_1, \mathrm{Pr}_2, \ldots, \mathrm{Pr}_N$ from $S_1, S_2, \ldots, S_N$ respectively. The resultant probability distribution from fusing $\mathrm{Pr}_1, \mathrm{Pr}_2, \ldots, \mathrm{Pr}_N$ should be contained by $S_\bullet$, no matter the choice of Pr_i's. $S_\bullet$ is considered "tight" if no other probability distributions are contained. $S_\bullet$ is considered "maximally tight" if there is no other credal

set of the *same subtype* $S'_\bullet$ that satisfies the containment property and is a proper subset of $S_\bullet$.

Like with point probabilities, general fusion using nontrivial credal sets can proceed in a sequential manner. However, the resultant credal set may not be as tight as the structure that results from simultaneous fusion.

When the causal network from Figure 6.2(b) is treated as a credal network, the strong extension [15] bears a similarity to the containment property. It is important to note however, that the "strong extension" of credal networks requires tightness, something that is not a requirement of the containment property. Also in the context of this chapter, the maximally tight posterior credal set of the correct subtype may not be as tight as the convex hull that constitutes the strong extension.

6.4.3 Lower and Upper Probability Bounds

As noted in [17, 80, 81], determining the lower and upper bounds for posterior probabilities requires the simultaneous minimization and maximization of probabilities. Let A denote a condition of interest, and let B denote a condition that is forced to be true (B denotes $\forall i = 1, 2, \ldots, N : O_i = o_i$ in the case of context specific fusion, and B denotes $E = 1$ in the case of general fusion). If $\Pr_L$ and $\Pr_U$ denote lower and upper probability bounds respectively, then ([17]):

$$\Pr_L(A|B) = \frac{\Pr_L(A \wedge B)}{\Pr_L(A \wedge B) + \Pr_U(\neg A \wedge B)} \quad \text{and} \quad \Pr_U(A|B) = \frac{\Pr_U(A \wedge B)}{\Pr_U(A \wedge B) + \Pr_L(\neg A \wedge B)}$$

For the credal set subtypes considered by this chapter, the minimization (maximization) of $\Pr(A \wedge B)$ does not interfere or interact with the maximization (minimization) of $\Pr(\neg A \wedge B)$.

6.4.4 Approximate approaches

In many cases, credal sets are denoted by listing their "extreme points". When credal sets are denoted by listing their extreme points, they are not confined to any subtype such as probability interval distributions or Dempster-Shafer models. [33] states and proves a theorem (referred to in [33] as Theorem 2) that implies that context specific and general fusion using credal sets can be done exactly, meaning that the containment property is satisfied and that the resultant credal set is "tight". This is not necessarily the case if the credal sets are restricted to a specific subtype.

In the context of this chapter, the output credal set has the same subtype as the credal sets used for the prior data in the case of context specific fusion, and the same subtype as the input credal sets in the case of general fusion. However, due to the limitations on the expressive power of each subtype of credal set, it is rarely possible to return a credal set of the desired subtype that is "tight". Moreover, in many cases, finding the tightest possible output credal set of the desired subtype may be computationally intractable, as will be seen in the subsequent sections. Both of these limitations imply that most fusion approaches discussed here will not return a tight credal set of the desired subtype. However, all fusion approaches will satisfy the containment property, something that Dempster's rule of combination (section 6.6.4) fails to satisfy.

In sections 6.5.1 and 6.6.1, probability interval distributions and Dempster-Shafer models are shown to be a more memory efficient alternative to listing the extreme points of credal sets. This increase in memory efficiency is argued to compensate for the decrease in accuracy caused when non-tight credal sets of the desired subtype are returned by fusion.

6.5 Probability Interval Fusion

6.5.1 Probability Intervals and credal sets

This section will give a simple example that demonstrates how a probability interval distribution can have a large number of extreme points, which makes the style of representation that is commonly used for credal sets, listing the extreme points, computationally intractable. Although it is known in [65] that the number of extreme points in a probability interval distribution is large, a concrete simple example is provided here for the convenience of the reader. This subsection will give an example of a probability interval distribution S over the values $1, 2, \ldots, M$, for which the number of extreme points is $\Omega(2^M/M^2)$. In other words, the number of extreme points is exponential with respect to M.

Let M be even. Let the j^{th} probability interval be $[l_j, u_j] = [0, 2/M]$. An extreme probability distribution of S is formed by choosing $M/2$ values from $1, 2, \ldots, M$ to be assigned a probability of $2/M$, and all other probabilities are assigned 0. The number of extreme probability distributions is hence:

$$\binom{M}{M/2} = \frac{M!}{(M/2)!^2}$$

using $\ln(n!) \in [n\ln(n) - n + 1, (n+1)\ln(n) - n + 1]$ gives:

$$\begin{aligned}
\frac{M!}{(M/2)!^2} &= \exp(\ln(M!) - 2\ln((M/2)!)) \\
&\geq \exp((M\ln(M) - M + 1) - 2((M/2+1)\ln(M/2) - M/2 + 1)) \\
&= \exp(M\ln(M) - (M+2)\ln(M/2) - 1) = \frac{2^M \cdot 4}{M^2 \cdot e} \\
&\in \Omega(2^M/M^2)
\end{aligned}$$

Here, big-"Omega" notation is used to denote a lower-bound (the opposite of big-"O" notation). A probability interval distribution requires the storage of $O(M)$ values, while the

credal set requires the storage of $\Omega(2^M/M^2)$ extreme probability distributions.

With this example, it is clear that representing a probability interval distribution by its extreme points is not efficient from a memory perspective, and is hence also inefficient from a time perspective. For instance if $M = 20$, then the number of extreme points is $\binom{20}{10} = 184756$.

6.5.2 Context Specific Fusion with Probability Intervals

A high level description of the process of context specific fusion using probability intervals can be found in [68, section 4, calculus], and [81].

Let S_0 and $[l_1, u_1], [l_2, u_2], \ldots, [l_M, u_M]$ denote the prior probability interval distribution for H.

After the observations $O_i = o_i$ have been received for each $i = 1, 2, \ldots, N$, for each $j = 1, 2, \ldots, M$, the set of possible values of $\Pr(O_i = o_i | H = j)$ is an interval $P_{i,j} = [l_{i,j}, u_{i,j}]$. Note that for each $i = 1, 2, \ldots, N$, that the intervals $[l_{i,1}, u_{i,1}], [l_{i,2}, u_{i,2}], \ldots, [l_{i,M}, u_{i,M}]$ *do not* collectively form a probability interval distribution.

The posterior probability interval distribution for H is determined by computing the smallest and largest possible posterior probabilities for each value of H. Let this posterior distribution be denoted by $[l_{\bullet,1}, u_{\bullet,1}], [l_{\bullet,2}, u_{\bullet,2}], \ldots, [l_{\bullet,M}, u_{\bullet,M}]$.

To find these extremes, let $p_1, p_2, \ldots, p_M$ denote an arbitrary prior probability distribution for H that is contained by S_0, and let $p_{i,j}$ for each $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$ denote an arbitrary probability from the interval $[l_{i,j}, u_{i,j}]$.

For an arbitrary $j' = 1, 2, \ldots, M$, in order to compute $l_{\bullet,j'}$, the probability of $H = j' \wedge \forall i \in \{1, 2, \ldots, N\} : O_i = o_i$ should be minimized, while the probability of $H \neq j' \wedge \forall i \in \{1, 2, \ldots, N\} : O_i = o_i$ should be maximized. This can be done by setting $p_{j'} = l_{j'}$; $p_{i,j'} = l_{i,j'}$ for each $i = 1, 2, \ldots, N$; and $p_{i,j} = u_{i,j}$ for each $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$ where $j \neq j'$. To decide upon each p_j where $j \neq j'$, a greedy maximization approach is used. Each p_j is set to l_j by default, and the following process is repeated: Find $j \in \{1, 2, \ldots, M\} \setminus \{j'\}$

that maximizes $c_j = \prod_{i=1}^{N} p_{i,j}$. Next, p_j should be set to the highest allowed probability (the probability is limited by both u_j and the fact that $\sum_{j=1}^{M} p_j = 1$). j should then be removed from the set $j \in \{1, 2, \ldots, M\} \setminus \{j'\}$, and a new j should be chosen. This process repeats until $\sum_{j=1}^{M} p_j = 1$. A similar process is used to compute each $u_{\bullet,j'}$.

The following algorithm depicts the process of context specific fusion using probability intervals. To save space, the steps involved in computing the upper bounds $u_{\bullet,j'}$ will be shown in parentheses beside the steps for computing the lower bounds $l_{\bullet,j'}$.

for $j = 1$ to M **do**
  $l_{\Pi,j} \leftarrow \prod_{i=1}^{N} l_{i,j}$
  $u_{\Pi,j} \leftarrow \prod_{i=1}^{N} u_{i,j}$
end for
for $j' = 1$ to M **do**
  // $l_{\bullet,j'}$ ($u_{\bullet,j'}$) will be computed.
  for $j = 1$ to M **do**
    $p_j \leftarrow l_j$ ($p_j \leftarrow u_j$)
    if $j = j'$ **then**
      /* The prior probability of $\Pr(H = j' \wedge \forall i = 1, 2, \ldots, N : O_i = o_i)$ should be minimized (maximized). */
      $c_j \leftarrow l_{\Pi,j}$ ($c_j \leftarrow u_{\Pi,j}$)
      $b_j \leftarrow 0$
    else
      /* The prior probability of $\Pr(H \neq j' \wedge \forall i = 1, 2, \ldots, N : O_i = o_i)$ should be maximized (minimized). */
      $c_j \leftarrow u_{\Pi,j}$ ($c_j \leftarrow l_{\Pi,j}$)
      $b_j \leftarrow 1$
    end if

end for

$\sigma \leftarrow \sum_{j=1}^{M} l_j$ ($\sigma \leftarrow \sum_{j=1}^{M} u_j$)

while $\sigma < 1$ ($\sigma > 1$) **do**

Find the j where $b_j = 1$ that maximizes c_j.

$p_j \leftarrow p_j + \min(u_j - l_j, 1 - \sigma)$ ($p_j \leftarrow p_j - \min(u_j - l_j, \sigma - 1)$)

$\sigma \leftarrow \sigma + \min(u_j - l_j, 1 - \sigma)$ ($\sigma \leftarrow \sigma - \min(u_j - l_j, \sigma - 1)$)

$b_j \leftarrow 0$

end while

$l_{\bullet,j'} \leftarrow \frac{p_{j'} \cdot c_{j'}}{\sum_{j=1}^{M} p_j \cdot c_j}$ ($u_{\bullet,j'} \leftarrow \frac{p_{j'} \cdot c_{j'}}{\sum_{j=1}^{M} p_j \cdot c_j}$)

end for

The overall time complexity for context specific fusion using probability intervals is $O(NM + M^2)$.

As an example of context specific fusion using probability intervals, the same example used for point probabilities in section 6.3.2 will be used. This time, however a ± 0.05 margin will be included on each probability:

$$\Pr(H = 1) = [0.85, 0.95] \qquad \Pr(H = 2) = [0.05, 0.15]$$

$$\Pr(O_1 = \text{“match”} | H = 1) = [0.85, 0.95] \qquad \Pr(O_1 = \text{“match”} | H = 2) = [0.35, 0.45]$$

$$\Pr(O_2 = \text{“no match”} | H = 1) = [0.25, 0.35] \qquad \Pr(O_2 = \text{“no match”} | H = 2) = [0.55, 0.65]$$

$$\Pr(O_3 = \text{“match”} | H = 1) = [0.65, 0.75] \qquad \Pr(O_3 = \text{“match”} | H = 2) = [0.15, 0.25]$$

If sensor 1 returns O_1 =“match”; sensor 2 returns O_2 =“no match”; and sensor 3 returns O_3 =“match”; then the posterior probability interval distribution for H is:

$$\Pr(H = 1 | O_1, O_2, O_3) \approx [0.3036, 0.9428] \quad \Pr(H = 2 | O_1, O_2, O_3) \approx [0.0572, 0.6964]$$

By comparison with the example from 6.3.2, it can be seen that the containment property is holding.

6.5.3 General Fusion with Probability Intervals

Each credal set S_i is a probability interval distribution, and the prior probability of $H_i = j$ is a closed interval $[l_{i,j}, u_{i,j}]$ instead of the point probability $p_{i,j}$. S_i now describes a set of probability distributions as opposed to a single probability distribution.

Here it should be note that exact fusion is not possible as probability intervals do not have the necessary expressive power to denote the exact set of possible fused probability distributions. Two approaches to approximate fusion will be covered in the next two sub-sections:

Finding the maximally tight posterior probability interval distribution for general fusion requires an algorithm for solving the following NP-hard problem:

Problem 6.4. Optimum sum of products

Input:

Two positive integers n and m.

Two $n \times m$ arrays of non-negative real numbers: $a_{i,j}$ and $b_{i,j}$ for each $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. It must be the case that:

$$\forall i \in \{1, 2, \ldots, n\} : \forall j \in \{1, 2, \ldots, m\} : 0 \leq a_{i,j} \leq b_{i,j}$$

One n length vector of non-negative real numbers: c_i for each $i = 1, 2, \ldots, n$. It must be the case that:

$$\forall i \in \{1, 2, \ldots, n\} : \sum_{j=1}^{m} a_{i,j} \leq c_i \leq \sum_{j=1}^{m} b_{i,j}$$

In addition, a choice between maximization and minimization must be made.

Internal Variables to be optimized:

One $n \times m$ array of non-negative real numbers: $x_{i,j}$ for each $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. The following restrictions hold:

$$\forall i \in \{1, 2, \ldots, n\} : \forall j \in \{1, 2, \ldots, m\} : a_{i,j} \leq x_{i,j} \leq b_{i,j}$$
$$\forall i \in \{1, 2, \ldots, n\} : \sum_{j=1}^{m} x_{i,j} = c_i$$

Output:

The maximum, or minimum depending on choice, possible value of the expression

$$\sum_{j=1}^{m} \prod_{i=1}^{n} x_{i,j}$$

Problem 6.4 in essence takes n unnormalized probability interval distributions over a domain of m values: $[a_{i,1}, b_{i,1}], [a_{i,2}, b_{i,2}], \ldots, [a_{i,m}, b_{i,m}]$, and extracts from each an unnormalized probability distribution $x_{i,1}, x_{i,2}, \ldots, x_{i,m}$ that sums to c_i. The probability distributions are chosen to either maximize or minimize the probability of agreement between all chosen probability distributions. A proof of the NP-hardness of problem 6.4 is given in appendix B.

It is not hard to show that the $x_{i,j}$'s that optimize $\sum_{j=1}^{m} \prod_{i=1}^{n} x_{i,j}$ attain a "corner state". That is, for each $i = 1, 2, \ldots, N$, $x_{i,j} = a_{i,j}$ or $x_{i,j} = b_{i,j}$ for all but one $j = 1, 2, \ldots, M$. There are a finite number of corner states, so as noted in [80, 81], problem 6.4 can be solved via an exhaustive search of the corner states.

Since problem 6.4 is NP-hard, approximate solutions are necessary for tractable calculations. None of the approximations made in this chapter will violate the containment property. In [9], an optimization problem that encompasses problem 6.4 is solved in an approximate manner using hill climbing iterations.

Approach 1

If computational intractability is not an issue, problem 6.4 can be solved to find the tightest possible lower and upper bounds for the posterior probability distribution. The following algorithm depicts the process of general fusion using probability intervals. To save space, the steps involved in computing the upper bounds $u_{\bullet,j'}$ will be shown in parentheses beside the steps for computing the lower bounds $l_{\bullet,j'}$.

for $j' = 1$ to M **do**

// $l_{\bullet,j'}$ ($u_{\bullet,j'}$) will be computed.

$q \leftarrow \prod_{i=1}^{N} l_{i,j'}$ ($q \leftarrow \prod_{i=1}^{N} u_{i,j'}$)

// q is the minimized (maximized) prior probability of $H_1 = H_2 = \cdots = H_N = j'$.

/* The maximum (minimum) prior probability of $H_1 = H_2 = \cdots = H_N \neq j'$ will now be computed using problem 6.4: */

$n \leftarrow N$

$m \leftarrow M - 1$

for $i = 1$ to N **do**

$c_i \leftarrow 1 - l_{i,j'}$ ($c_i \leftarrow 1 - u_{i,j'}$)

for $j = 1$ to $M - 1$ **do**

if $j < j'$ **then**

$a_{i,j} \leftarrow l_{i,j}$ and $b_{i,j} \leftarrow u_{i,j}$

else

$a_{i,j} \leftarrow l_{i,j+1}$ and $b_{i,j} \leftarrow u_{i,j+1}$

end if

end for

end for

Solve problem 6.4 with the values of n, m, $a_{i,j}$, $b_{i,j}$, c_i, and use maximization (minimization). Assign the result to r.

$l_{\bullet,j'} \leftarrow \frac{q}{q+r}$ ($u_{\bullet,j'} \leftarrow \frac{q}{q+r}$)

end for

The overall computational complexity for approach #1 is $O(NM+Mf(N,M-1))$ where $f(n,m)$ is the computational complexity of problem 6.4. While the arrays are being freshly generated for each application of problem 6.4 in the pseudo code above, the computational complexity assumes a single array can be pre-calculated at a cost of $O(NM)$ and used for all applications of problem 6.4 with different columns ignored.

Problem 6.4 is NP-hard, but when $n = 2$, the problem is greatly simplified. Problem 6.4 when $n = 2$ becomes a bilinear programming problem. The resultant bilinear programming programming problem can be more easily solved, and provides a means through which the probability interval distributions can be fused in a sequential manner. It should be noted however, that the resultant probability interval distribution will not be as tight as the probability interval distribution formed through simultaneous fusion.

When probability interval distributions are fused in a pairwise manner, the computational complexity of approach #1 reduces to $O(NM + NMf(2, M-1))$.

Approach 2

Theory from [17] can be used for the general fusion of probability interval distributions. It should be noted however, that the approach presented here will fail to be maximally tight for the following reasons: In the context of general fusion, the hypothesis variables $H_1, H_2, \ldots, H_N$ are all independent when E is ignored. When a joint probability interval distribution is formed that covers the hypothesis variables, the independence between $H_1, H_2, \ldots, H_N$ can no longer be enforced. This makes possible joint probability distributions over $H_1, H_2, \ldots, H_N$ that do not satisfy the independence between the hypothesis variables.

When $E = 1$, H will denote the common value of $H_1, H_2, \ldots, H_N$.

Ignoring the variable E, the joint probability interval for $\Pr(\forall i \in \{1, 2, \ldots, N\} : H_i = j_i)$, where $\langle j_1, j_2, ..., j_N \rangle$ is an arbitrary assignment to the hypothesis variables $H_1, H_2, ..., H_N$, is $\left[\prod_{i=1}^{N} l_{i,j_i}, \prod_{i=1}^{N} u_{i,j_i}\right]$

For each $j' \in \{1, 2, \ldots, M\}$, the smallest and largest posterior probability $\Pr(H = j'|E = 1)$ is

$$l_{\bullet,j'} = \frac{\Pr_L(H = j' \wedge E = 1)}{\Pr_L(H = j' \wedge E = 1) + \Pr_U(H \neq j' \wedge E = 1)}$$

and

$$u_{\bullet,j'} = \frac{\Pr_U(H = j' \wedge E = 1)}{\Pr_U(H = j' \wedge E = 1) + \Pr_L(H \neq j' \wedge E = 1)}$$

respectively.

For each $j' \in \{1, 2, \ldots, M\}$, $\prod_{i=1}^{N} l_{i,j'}$ is the smallest possible prior probability $\Pr_L(H = j' \wedge E = 1)$. The maximum prior probability $\Pr_U(H \neq j' \wedge E = 1)$ seems to be $\sum_{j:j\neq j'} \prod_{i=1}^{N} u_{i,j}$. However, while each upper bound is attainable, upper bounds may not be *simultaneously attainable*. Another upper bound on the prior probability $\Pr(H \neq j' \wedge E = 1)$ arises from a lower bound on the prior probability $\Pr(H = j' \vee E = 0)$. A lower bound on the prior probability $\Pr(H = j' \vee E = 0)$ that arises directly from the probability intervals is $\prod_{i=1}^{N} l_{i,j'} + L$ where:

$$L = \prod_{i=1}^{N} \sum_{j=1}^{M} l_{i,j} - \sum_{j=1}^{M} \prod_{i=1}^{N} l_{i,j}$$

The maximum prior probability $\Pr_U(H \neq j' \wedge E = 1)$ is: $\min\left(\sum_{j:j\neq j'} \prod_{i=1}^{N} u_{i,j}, 1 - \prod_{i=1}^{N} l_{i,j'} - L\right)$

The smallest possible posterior probability $\Pr_L(H = j'|E = 1)$ is:

$$l_{\bullet,j'} = \frac{\prod_{i=1}^{N} l_{i,j'}}{\min\left(\prod_{i=1}^{N} l_{i,j'} + \sum_{j:j\neq j'} \prod_{i=1}^{N} u_{i,j}, 1 - L\right)}$$

Using a similar argument, the largest possible posterior probability $\Pr_U(H = j'|E = 1)$

is:

$$u_{\bullet,j'} = \frac{\prod_{i=1}^{N} u_{i,j'}}{\max\left(\prod_{i=1}^{N} u_{i,j'} + \sum_{j:j\neq j'} \prod_{i=1}^{N} l_{i,j}, 1-U\right)}$$

where

$$U = \prod_{i=1}^{N}\sum_{j=1}^{M} u_{i,j} - \sum_{j=1}^{M}\prod_{i=1}^{N} u_{i,j}$$

The above gives the complete approach to computing the posterior probability interval distribution $S_\bullet$ for H: $[l_{\bullet,1}, u_{\bullet,1}]; [l_{\bullet,2}, u_{\bullet,2}]; \ldots; [l_{\bullet,M}, u_{\bullet,M}]$.

The overall computational complexity for approach #2 is $O(N \cdot M)$. To achieve this efficiency, the following expressions should be computed in the following order:

$$\left(\forall i \in \{1,2,\ldots,N\} : l_{i,\Sigma} = \sum_{j=1}^{M} l_{i,j} \ ; \ u_{i,\Sigma} = \sum_{j=1}^{M} u_{i,j}\right) \ ; \ L_{\Pi\Sigma} = \prod_{i=1}^{N} l_{i,\Sigma} \ ; \ U_{\Pi\Sigma} = \prod_{i=1}^{N} u_{i,\Sigma}$$

$$\left(\forall j \in \{1,2,\ldots,M\} : l_{\Pi,j} = \prod_{i=1}^{N} l_{i,j} \ ; \ u_{\Pi,j} = \prod_{i=1}^{N} u_{i,j}\right) \ ; \ L_{\Sigma\Pi} = \sum_{j=1}^{M} l_{\Pi,j} \ ; \ U_{\Sigma\Pi} = \sum_{j=1}^{M} u_{\Pi,j}$$

$$\forall j \in \{1,2,\ldots,M\} : l_{\bullet,j} = \frac{l_{\Pi,j}}{\min(l_{\Pi,j} + (U_{\Sigma\Pi} - u_{\Pi,j}), 1 - (L_{\Pi\Sigma} - L_{\Sigma\Pi}))}$$

$$\forall j \in \{1,2,\ldots,M\} : u_{\bullet,j} = \frac{u_{\Pi,j}}{\max(u_{\Pi,j} + (L_{\Sigma\Pi} - l_{\Pi,j}), 1 - (U_{\Pi\Sigma} - U_{\Sigma\Pi}))}$$

As an example of general fusion using approach #2 for probability intervals, the same example used for point probabilities in section 6.3.3 will be used. This time, however a ± 0.05 margin will be included on each probability:

$$\Pr(H_1 = 1) = [0.40, 0.50] \qquad \Pr(H_1 = 2) = [0.50, 0.60]$$

$$\Pr(H_2 = 1) = [0.55, 0.65] \qquad \Pr(H_2 = 2) = [0.35, 0.45]$$

$$\Pr(H_3 = 1) = [0.05, 0.15] \qquad \Pr(H_3 = 2) = [0.85, 0.95]$$

The posterior probability distribution for H, the common value, is:

$$\Pr(H = 1) \approx [0.0411, 0.2468] \quad \Pr(H = 2) \approx [0.7532, 0.9589]$$

By comparison with the example from 6.3.3, it can be seen that the containment property is holding.

6.6 Dempster-Shafer Fusion

6.6.1 Dempster-Shafer models and credal sets

Like with probability intervals, a Dempster-Shafer model S over the domain $1, 2, \ldots, M$ for which the number of extreme points greatly exceeds the size of S will be constructed to prove the utility of Dempster-Shafer models in comparison with listing the extreme points. The size of the Dempster-Shafer model is $O(2^M)$. In this case, the number of extreme points will be $\Omega(M!)$. Let $M \gg 1$ be arbitrary: For each $J \in \text{Set}(S)$, let $m(J) = \frac{1}{2^M - 1}$. An extreme probability distribution is formed by choosing a permutation of $1, 2, \ldots, M$. Let $\rho : \text{Val}(S) \to \text{Val}(S)$ denote this permutation. All probability mass gravitates to $\rho(1)$; followed by $\rho(2)$; and so on. The probability assigned to $\rho(j)$ for each $j = 1, 2, \ldots, M$ is: $p_j = \frac{2^{(M-j)}}{(2^M)-1}$. The number of permutations ρ is $M!$, so the number of extreme points is $\Omega(M!)$.

Again, it is clear that representing a Dempster-Shafer model by its extreme points is not computationally efficient. For instance, if $M = 20$, a Dempster-Shafer model requires $2^{20} - 1 = 1048575$ values, while the number of extreme points is $20! \approx 2.4329 \times 10^{18}$.

6.6.2 Context Specific Fusion with Dempster-Shafer models

An approach to context specific fusion using Dempster-Shafer models is given in [18]. The approach presented here however, will differ from the approach given in [18], since the presented approach will aim to satisfy the containment property. The approach presented here will bear similarities to the approach from [81].

Context specific fusion using Dempster-Shafer models proceeds in a very similar manner to context specific fusion using probability intervals.

Let S_0; $m_0 : \text{Set}(H) \to [0,1]$; and $\text{Bel}_0 : \text{Set}(H) \to [0,1]$ all denote the prior Dempster-Shafer model for H.

After the observations $O_i = o_i$ have been received for each $i = 1, 2, \ldots, N$, for each $j = 1, 2, \ldots, M$, the set of possible values of $\Pr(O_i = o_i | H = j)$ is an interval $P_{i,j} = [l_{i,j}, u_{i,j}]$. Each $P_{i,j}$ is simply an interval, as a full Dempster-Shafer model that covers O_i is not required. Only the scenario of $O_i = o_i$ is under consideration.

The posterior Dempster-Shafer model for H is determined by computing the smallest (or largest) possible posterior probabilities for each nonempty subset of $\text{Val}(H)$. Let the lower (or upper) posterior probability of $J \in \text{Set}(H)$ be denoted by $\text{Bel}_\bullet(J)$ (or $\text{Pl}_\bullet(J)$).

The following algorithm depicts the process of context specific fusion using probability intervals. To save space, the steps involved in computing the plausibilities $\text{Pl}_\bullet$ will be shown in parentheses beside the steps for computing the beliefs $\text{Bel}_\bullet$. (Note however, that only computing the beliefs are necessary for the posterior Dempster-Shafer model.)

for $j = 1$ to M **do**

$l_{\Pi,j} \leftarrow \prod_{i=1}^{N} l_{i,j}$

$u_{\Pi,j} \leftarrow \prod_{i=1}^{N} u_{i,j}$

end for

for all $J' \in \text{Set}(H)$ **do**

// $\text{Bel}_\bullet(J')$ ($\text{Pl}_\bullet(J')$) will be computed.

for $j = 1$ to M **do**

$p_j \leftarrow 0$

if $j \in J'$ **then**

`/* The prior probability of` $\Pr(H \in J' \wedge \forall i = 1, 2, \ldots, N : O_i = o_i)$ `should be minimized (maximized). */`

$c_j \leftarrow l_{\Pi,j}$ ($c_j \leftarrow u_{\Pi,j}$)

else

`/* The prior probability of` $\Pr(H \notin J' \wedge \forall i = 1, 2, \ldots, N : O_i = o_i)$ `should be maximized (minimized). */`

$c_j \leftarrow u_{\Pi,j}$ ($c_j \leftarrow l_{\Pi,j}$)

end if

end for

for all $J \in \mathrm{Set}(H)$ **do**

if $J \subseteq J'$ **then**

Find the $j \in J$ that minimizes (maximizes) c_j.

else if $J \cap J' = \emptyset$ **then**

Find the $j \in J$ that maximizes (minimizes) c_j.

else

Find the $j \in J \setminus J'$ ($j \in J \cap J'$) that maximizes c_j.

end if

$p_j \leftarrow p_j + m_0(J)$

end for

$\mathrm{Bel}_\bullet(J') \leftarrow \frac{\sum_{j' \in J'} p_{j'} \cdot c_{j'}}{\sum_{j=1}^{M} p_j \cdot c_j}$ ($\mathrm{Pl}_\bullet(J') \leftarrow \frac{\sum_{j' \in J'} p_{j'} \cdot c_{j'}}{\sum_{j=1}^{M} p_j \cdot c_j}$)

end for

The overall time complexity for context specific fusion (including the cost of determining the final masses using the inclusion/exclusion principle) using Dempster-Shafer models is

$O(NM + 2^{2M})$. Since the size D of the prior Dempster-Shafer model for H is approximately 2^M, the time complexity is in fact $O(NM + D^2)$.

As an example of context specific fusion using Dempster-Shafer models, the same example used for point probabilities in section 6.3.2 will be used. A ± 0.05 margin will be included to form the prior Dempster-Shafer model for H and each probability interval for the observed evidence:

$$m_0(\{1\}) = 0.85 \qquad m_0(\{2\}) = 0.05 \qquad m_0(\{1,2\}) = 0.1$$

$$\Pr(o_1|H=1) = [0.05, 0.15] \qquad \Pr(o_1|H=2) = [0.55, 0.65]$$

$$\Pr(o_2|H=1) = [0.25, 0.35] \qquad \Pr(o_2|H=2) = [0.55, 0.65]$$

$$\Pr(o_3|H=1) = [0.65, 0.75] \qquad \Pr(o_3|H=2) = [0.15, 0.25]$$

The posterior Dempster-Shafer model for H is:

$$m_\bullet(\{1\}) \approx 0.3036 \quad m_\bullet(\{2\}) \approx 0.0572 \quad m_\bullet(\{1,2\}) \approx 0.6392$$

By comparison with the example from 6.3.2, it can be seen that the containment property is holding.

6.6.3 General Fusion with Dempster-Shafer models

Dempster's rule of combination (described in [71]) performs general fusion of Dmpster-Shafer models. However, Dempster's rule of combination fails to satisfy the containment property.

Each structure S_i is a Dempster-Shafer model. S_i is denoted by either the mass function $m_i : \text{Set}(H_i) \to [0,1]$; or the belief function $\text{Bel}_i : \text{Set}(H_i) \to [0,1]$. Also, the resultant Dempster-Shafer model $S_\bullet$ is denoted by either $m_\bullet : \text{Set}(H) \to [0,1]$; or $\text{Bel}_\bullet : \text{Set}(H) \to [0,1]$.

Finding the tightest possible Dempster-Shafer model for general fusion requires an algorithm for solving the following problem:

Problem 6.5. Optimum sum of products, Dempster-Shafer variant

Input:

Two positive integers n and m.

A set A with m distinct quantities.

An n length array of functions: $f_i : 2^A \setminus \{\emptyset\} \to [0, +\infty)$ for each $i = 1, 2, \ldots, n$.

In addition, a choice between maximization and minimization must be made.

Internal Variables to be optimized:

An n length array of functions: $g_i : 2^A \setminus \{\emptyset\} \to A$ for each $i = 1, 2, \ldots, n$. The following restriction must hold:

$$\forall i \in \{1, 2, \ldots, n\} : \forall J \in 2^A \setminus \{\emptyset\} : g_i(J) \in J$$

An $n \times m$ array of non-negative real numbers: $x_{i,j}$ for each $i = 1, 2, \ldots, n$ and $j \in A$ where:

$$\forall i \in \{1, 2, \ldots, n\} : \forall j \in A : x_{i,j} = \sum_{J \in 2^A \setminus \{\emptyset\} \wedge g_i(J) = j} f_i(J)$$

Output:

The maximum, or minimum depending on choice, possible value of the expression

$$\sum_{j \in A} \prod_{i=1}^{n} x_{i,j}$$

Problem 6.5, which is directly analogous to problem 6.4, in essence takes n unnormalized Dempster-Shafer models over a domain of m values: $A = \{1, 2, \ldots, m\}$. From each Dempster-Shafer model $i = 1, 2, \ldots, n$, each probability mass is focused onto a single element, forming an unnormalized probability distribution $x_{i,1}, x_{i,2}, \ldots, x_{i,m}$. The probability distributions are chosen to either maximize or minimize the probability of agreement between all chosen

probability distributions.

Approach 1

Like with probability intervals, if computational intractability is not an issue, problem 6.5 can be solved to find the tightest Dempster-Shafer model for the posterior probability distribution. The following algorithm depicts the process of general fusion using Dempster-Shafer models. To save space, the steps involved in computing the plausibilities $\text{Pl}_\bullet$ will be shown in parentheses beside the steps for computing the beliefs $\text{Bel}_\bullet$. (Note however, that only computing the beliefs are necessary for the posterior Dempster-Shafer model.)

for all $J' \in \text{Set}(H)$ **do**

`//` $\text{Bel}_\bullet(J')$ `(`$\text{Pl}_\bullet(J')$`) will be computed.`

`/* The minimum prior probability of` $E = 1 \wedge H \in J'$ `(`$E = 1 \wedge H \notin J'$`) will now be computed using problem 6.5: */`

$n \leftarrow N$

$m \leftarrow |J'|$ $(m \leftarrow M - |J'|)$

$A \leftarrow J'$ $(A \leftarrow \text{Val}(H) \setminus J')$

for $i = 1$ to n **do**

for all $J \in 2^A \setminus \{\emptyset\}$ **do**

$f_i(J) \leftarrow m_i(J)$

end for

end for

Solve problem 6.5 with the values of n, m, A, f_i, and use minimization. Assign the result to q.

`/* The maximum prior probability of` $E = 1 \wedge H \notin J'$ `(`$E = 1 \wedge H \in J'$`) will now be computed using problem 6.5: */`

$m \leftarrow M - |J'|$ $(m \leftarrow |J'|)$

$A \leftarrow \text{Val}(H) \setminus J'$ ($A \leftarrow J'$)

for $i = 1$ to n **do**

for all $J \in 2^A \setminus \{\emptyset\}$ **do**

`/* Since the prior probability of` $E = 1 \wedge H \in A$ `is being maximized, probability mass gravitates into` A`: */`

$f_i(J) \leftarrow \sum_{J'' \in \text{Set}(H) \wedge J'' \cap A = J} m_i(J'')$

end for

end for

Solve problem 6.5 with the values of n, m, A, f_i, and use maximization. Assign the result to r.

$\text{Bel}_\bullet(J') \leftarrow \frac{q}{q+r}$ ($\text{Pl}_\bullet(J') \leftarrow \frac{r}{q+r}$)

end for

The overall computational complexity for approach #1 is $O(2^{2M}N + 2^M g(N, M))$ where $g(n, m)$ is the computational complexity of problem 6.5. While the input functions are being freshly generated for each application of 6.5 in the pseudo code above, the computational complexity assumes that all input functions can be pre-calculated at a cost of $O(2^{2M}N)$ and used for all applications of problem 6.5 with different entries ignored.

When Dempster-Shafer models are fused in a pairwise manner, the computational complexity of approach #1 reduces to $O(2^{2M}N + 2^M N g(2, M))$.

Approach 2

The second approach to general fusion using Dempster-Shafer models also uses theory from [17] and is similar to general fusion approach #2 for probability intervals. Like with probability intervals, a joint Dempster-Shafer model is created that covers the variables $H_1, H_2, \ldots, H_N$. Again, like with probability intervals, the joint Dempster-Shafer model will fail to enforce the independence between variables $H_1, H_2, \ldots, H_N$. For this fusion approach

the assumption that $N \geq 2$ is important.

The joint Dempster-Shafer model for the variables $H_1, H_2, \ldots, H_N$, denoted by $S_\times$, is created as follows: consider $J_\times = J_1 \times J_2 \times \cdots \times J_N$ for arbitrary $J_1, J_2, \ldots, J_N \in \text{Set}(H)$. Let the mass assigned to $J_\times$ be: $m_\times(J_\times) = m_1(J_1)m_2(J_2)\ldots m_n(J_N)$. For any $J_\times \in \text{Set}(\{H_1, H_2, \ldots, H_N\})$, if there does not exist any $J_1, J_2, \ldots, J_N \in \text{Set}(H)$ such that $J_\times = J_1 \times J_2 \times \cdots \times J_N$, then the mass assigned to $J_\times$ is 0: $m_\times(J_\times) = 0$.

The calculation of $\text{Bel}_\bullet(J')$ (and $\text{Pl}_\bullet(J')$) for an arbitrary $J' \in \text{Set}(H)$ will now be the focus. When point probabilities are used, the posterior probability for $H \in J'$ is: $\Pr(H \in J') = \frac{\Pr(H \in J' \wedge E=1)}{\Pr(H \in J' \wedge E=1)+\Pr(H \notin J' \wedge E=1)}$. The condition that $E = 1$ requires that $H_1 = H_2 = \cdots = H_N$, and H denotes the common value. As noted in section 6.4,

$$\Pr_L(H \in J' | E = 1) = \frac{\Pr_L(H \in J' \wedge E = 1)}{\Pr_L(H \in J' \wedge E = 1) + \Pr_U(H \notin J' \wedge E = 1)}$$

and

$$\Pr_U(H \in J' | E = 1) = \frac{\Pr_U(H \in J' \wedge E = 1)}{\Pr_U(H \in J' \wedge E = 1) + \Pr_L(H \notin J' \wedge E = 1)}$$

Therefore:

$$\forall J' \in \text{Set}(H): \quad \text{Bel}_\bullet(J') = \frac{\text{Bel}_\times(H \in J' \wedge E = 1)}{\text{Bel}_\times(H \in J' \wedge E = 1) + \text{Pl}_\times(H \in (\text{Val}(H) \setminus J') \wedge E = 1)}$$

$$\forall J' \in \text{Set}(H): \quad \text{Pl}_\bullet(J') = \frac{\text{Pl}_\times(H \in J' \wedge E = 1)}{\text{Pl}_\times(H \in J' \wedge E = 1) + \text{Bel}_\times(H \in (\text{Val}(H) \setminus J') \wedge E = 1)}$$

where

$$\forall J' \in \text{Set}(H): \quad \text{Bel}_{\times}(H \in J' \wedge E = 1) = \sum_{j \in J'} \prod_{i=1}^{N} m_i(\{j\})$$

$$\forall J' \in \text{Set}(H): \quad q_{\times}(H \in J' \wedge E = 1) = \prod_{i=1}^{N} \sum_{J \supseteq J'} m_i(J)$$

$$\forall J' \in \text{Set}(H): \quad \text{Pl}_{\times}(H \in J' \wedge E = 1) = \sum_{J \subseteq J' \wedge J \neq \emptyset} (-1)^{1+|J|} q_{\times}(H \in J \wedge E = 1)$$

Note that the quantities $\text{Bel}_{\times}(H \in \emptyset \wedge E = 1)$ and $\text{Pl}_{\times}(H \in \emptyset \wedge E = 1)$ default to 0.

The expression $\sum_{j \in J'} \prod_{i=1}^{N} m_i(\{j\})$ is non-zero if and only if there exists some $j \in J'$ for which $m_i(\{J\}) > 0$ for all $i = 1, 2, \ldots, N$. For this approach to general fusion using Dempster-Shafer models, masses assigned to singleton elements of $\text{Set}(H)$ are important for the creation of non-trivial Dempster-Shafer models.

The computational complexity of approach #2 is $O(2^{2M}N)$.

As an example of general fusion using approach #2 for Dempster-Shafer models, the same example used for point probabilities in section 6.3.3 will be used. This time however, a ± 0.05 margin will be included to form each Dempster-Shafer model:

$m_1(\{1\}) = 0.40$	$m_1(\{2\}) = 0.50$	$m_1(\{1,2\}) = 0.10$
$m_2(\{1\}) = 0.55$	$m_2(\{2\}) = 0.35$	$m_2(\{1,2\}) = 0.10$
$m_3(\{1\}) = 0.05$	$m_3(\{2\}) = 0.85$	$m_3(\{1,2\}) = 0.10$

The posterior probability distribution for H, the common value, is:

$$m_{\bullet}(\{1\}) \approx 0.0411 \quad m_{\bullet}(\{2\}) \approx 0.7532 \quad m_{\bullet}(\{1,2\}) \approx 0.2057$$

By comparison with the example from 6.3.3, it can be seen that the containment property is holding.

6.6.4 Dempster's Rule of Combination

Dempster's rule of combination performs general fusion of Dempster Shafer models. Dempster's rule of combination, described in [71], proceeds as follows:

$$\forall J' \in \mathrm{Set}(H) : m_\bullet(J') = \frac{1}{K} \sum_{\substack{J_1, J_2, \ldots, J_N \in \mathrm{Set}(H) \\ J_1 \cap J_2 \cap \cdots \cap J_N = J'}} m_1(J_1) m_2(J_2) \ldots m_N(J_N)$$

where K is a normalization constant that ensures that $\sum_{J' \in \mathrm{Set}(H)} m_\bullet(J') = 1$.

As will be shown in the following example, Dempster's rule of combination fails to satisfy the containment property. The example is from [22].

As an example of Dempster's rule of combination failing to satisfy the containment property, let $N = 2$ and $M = 2$. Let Dempster-Shafer model S_1 be defined by:

$$m_1(\{1\}) = 0.1 \quad m_1(\{2\}) = 0.1 \quad m_1(\{1,2\}) = 0.8$$

Let Dempster-Shafer model S_2 be the same: $S_2 = S_1$.

Dempster's rule of combination gives the following resultant Dempster-Shafer model $S_\bullet$:

$$m_\bullet(\{1\}) \approx 0.1735 \quad m_\bullet(\{2\}) \approx 0.1735 \quad m_\bullet(\{1,2\}) \approx 0.6531$$

Now consider probability distribution $\Pr_1 \in S_1$:

$$\Pr_1(1) = 0.1 \quad \Pr_1(2) = 0.9$$

Let probability distribution $\Pr_2 \in S_2$ be the same: $\Pr_2 = \Pr_1$.

Fusing $\Pr_1$ and $\Pr_2$ gives $\Pr_\bullet$:

$$\Pr_\bullet(1) \approx 0.0122 \quad \Pr_\bullet(2) \approx 0.9878$$

It is readily apparent that $\Pr_\bullet \notin S_\bullet$, which violates the containment property for general fusion. This example demonstrates that Dempster's rule of combination violates the containment property for general fusion.

Due to the fact that Dempster's rule of combination fails to satisfy the containment property, general fusion approach #2 is proposed as an alternative to Dempster's rule of combination.

6.7 The taxonomy of fusion approaches

The complete taxonomy of fusion approaches described in this chapter are listed below. This taxonomy is a key contribution of this chapter. N is the number of sensors, and M is the domain size of the hypothesis variable H. $f(n, m)$ denotes the computational complexity of problem 6.4 from section 6.5.3, and $g(n, m)$ denotes the computational complexity of problem 6.5 from section 6.6.3.

- Point probability distributions:
  - Context specific fusion (section 6.3.2): This approach is well known. The posterior credal set is maximally tight. The computational complexity is $O(NM)$.

  - General fusion (section 6.3.3): This approach is well known. The posterior credal set is maximally tight. The computational complexity is $O(NM)$.

- Probability Interval Distributions:

  - Context specific fusion (section 6.5.2): This approach is described in a high level in [68, section 4, calculus] and [81], and utilizes a greedy algorithm. The posterior credal set is maximally tight. The computational complexity is $O(NM + M^2)$.

  - General fusion approach #1 (section 6.5.3): This approach was independently derived for this chapter. The posterior credal set is maximally tight. The computational complexity is $O(NM + Mf(N, M-1))$.

  - General fusion approach #2 (section 6.5.3): This approach uses the existing theory of probability intervals [17]. The posterior is not maximally tight. The computational complexity is $O(NM)$.

- Dempster-Shafer models:

  - Context specific fusion (section 6.6.2): This approach was independently derived for this chapter, and utilizes a greedy algorithm. The posterior is maximally tight. The computational complexity is $O(NM + 2^{2M})$.

  - General fusion approach #1 (section 6.6.3): This approach was independently derived for this chapter. The posterior is maximally tight. The computational complexity is $O(2^{2M}N + 2^M g(N, M))$.

  - General fusion approach #2 (section 6.6.3): This approach was independently derived for this chapter. The posterior is not maximally tight. The computational complexity is $O(2^{2M}N)$.

6.8 Conclusion

This chapter has given a taxonomy of approaches to both context specific and general fusion using both probability interval distributions and Dempster-Shafer models. All listed fusion approaches will satisfied the "containment property". Various aspects of the fusion algorithm are cataloged, such as whether the resultant credal set is "maximally tight", and the computational complexity. Approaches that were given for general fusion exhibit tradeoffs between accuracy and computational complexity. Some of the fusion approaches were already known to the literature (such as context specific fusion with probability intervals described in [68]), and others were created specifically for this chapter.

The containment property, which requires that the fusion of any choice of point probability distributions be contained in the resultant credal set, is presented as an objective requirement that all fusion approaches should satisfy. Dempster's rule of combination is shown to not satisfy the containment property (see section 6.6.4).

Credal sets are convex sets of probability distributions, and a typical approach to denoting credal sets is to list their extreme points. It has been shown in [33] that context specific fusion and general fusion can be exactly and computationally efficiently performed by listing the extreme points of credal sets. Exact fusion requires that the containment property holds and that the resultant model is tight. This at first seems to imply that listing the extreme points of credal sets are the optimal approach to describing convex sets of probability distributions. This chapter shows however, that representing probability interval distributions and Dempster-Shafer models using lists of their extreme points can lead to excessive memory requirements and poor computational efficiency. Therefore, this chapter proposes probability intervals and Dempster-Shafer models as a computationally tractable alternative to the listing of extreme points.

Unlike listing extreme points, context specific and general fusion using probability interval distributions and Dempster-Shafer models can rarely be performed exactly. Moreover, probability intervals and Dempster-Shafer models lack the expressive power to denote a tight

posterior credal set. All approaches to fusion proposed here satisfy the containment property, and the approaches presented have varying levels of speed and accuracy.

There are many directions for future work. Problems 6.4 and 6.5 can be further investigated for more accurate and computationally efficient algorithms despite problem 6.4 being NP-hard. The algorithms for context specific fusion and general fusion can be generalized to "credal networks" (existing work on credal networks can be found in [9, 10, 13, 16]). The approaches presented in this chapter can be generalized by adding "credibility" weights to each sensor, as all sensors may not be equally credible. Lastly, the presented approaches can be investigated for specific applications of sensor fusion.

Chapter 7

Layered Probability Models

7.1 Introduction

This chapter will focus on the difficulty of creating graphical probability models from raw data. The process of generating a Markov or Bayesian network from complete or incomplete data has been well studied. Listed in table 7.1 is reference material that uses several of the known approaches.

The primary goal of this chapter is to develop an alternative to Markov and Bayesian networks as a graphical probability model. This alternative, referred to as a **layered probability model**, will be designed to enable quick probabilistic inference by speeding up the process of marginalizing variables. However, the most important property that this alternative graphical model will have is the ability "ignore" parts of the model and still attain meaningful posterior probabilities from probabilistic inference. In Markov and Bayesian networks, you cannot simply ignore parts of the model, such as factors and conditional probability tables, during probabilistic inference and expect to get a meaningful result.

In Figure 7.1, a simplified depiction of the process of both generating a layered probability model and using it for probabilistic inference is shown. Given raw statistical data related to set of variables that are to be modeled, approximations that are either manually or auto-

Table 7.1: A description of existing work related to the generation of probability models from raw data.

Reference(s)	Contribution
[40]	A summary of various approaches to training Markov and Bayesian networks. Approaches that determine the numerical values that populate factors and conditional probability tables include Maximum Likelihood Estimation and Expectation Maximization. Approaches that determine the structure of Markov and Bayesian networks include a variety of optimization approaches.
[45]	[45] describes a software tool known as the "Bayesian Network Toolbox for Matlab". This multi-purpose tool box uses a variety of algorithms to compute both the parameters and structure of Markov and Bayesian networks. Parameter learning uses the well established maximum likelihood and maximum posterior estimation, as well as expectation maximization. Structure learning uses approaches that search for an optimal model while respecting specified constraints.
[49]	A stack of "Hidden Markov Models" is trained and used for a specific application, which in the context of this paper, is tracking office activity.

matically chosen are used to generate "layers" (these "layers" are referred to as "correlation terms" and are further described in section 7.2). This newly formed "layered model" will then be used in straightforwards probabilistic inference. Unlike the use of Markov networks, marginalization will be trivial and variable conditioning will be difficult (which is opposite the case with Markov networks).

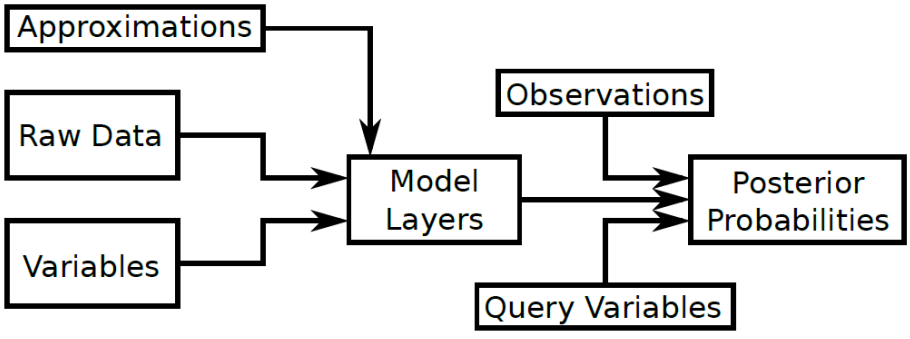


Figure 7.1: A high level simplified depiction of the process of both generating a layered probability model, and using the model to derive posterior probabilities.

7.2 Modeling Correlations

The most difficult task in generating a joint probability model that covers several random variables is that of finding and modeling the correlations that exist between the random variables.

If all variables were independent from each other, then the joint probability distribution is simply $\Pr(V) = \prod_{x \in \mathcal{X}} \Pr(V[x])$ for each $V \in \text{Val}(\mathcal{X})$. This is not the case in most cases however, and the probability distributions are more complex. The basic model $\Pr(V) = \prod_{x \in \mathcal{X}} \Pr(V[x])$ however can serve as a basis for more complex models.

Consider for instance two variables $x, y \in \mathcal{X}$, and that we are interested in computing the marginal probability distribution $\Pr(\langle v_x, v_y \rangle)$ for each $v_x \in \text{Val}(x)$ and $v_y \in \text{Val}(y)$. Computing the marginal probabilities distributions $\Pr(v_x)$ for each $v_x \in \text{Val}(x)$, and $\Pr(v_y)$ for each $v_y \in \text{Val}(y)$, if $x \perp y$ then $\Pr(\langle v_x, v_y \rangle) = \Pr(v_x) \Pr(v_y)$. When x and y are not independent, then $\Pr(\langle v_x, v_y \rangle) = \Pr(v_x) \Pr(v_y) + t(v_x, v_y)$ where $t(v_x, v_y)$ is a "correlation term" that "corrects" the approximate joint probability distribution $\Pr(v_x) \Pr(v_y)$ to match $\Pr(\langle v_x, v_y \rangle)$. Note that summing the entries of t holding all but one variable constant should result in 0: $\sum_{v_x \in \text{Val}(x)} t(v_x, v_y) = 0$ for each $v_y \in \text{Val}(y)$, and $\sum_{v_y \in \text{Val}(y)} t(v_x, v_y) = 0$ for each $v_x \in \text{Val}(x)$. Correlation terms are formally defined below:

Definition 7.1. Given an arbitrary set of variables $\mathcal{Y}$, a **correlation term** t which covers $\mathcal{Y}$ is a function $t : \text{Val}(\mathcal{Y}) \to \mathbb{R}$ for which each variable $x \in \mathcal{Y}$ and assignment $V \in \text{Val}(\mathcal{Y} \setminus x)$, it is the case that $\sum_{v_x \in \text{Val}(x)} t(\langle v_x, V \rangle) = 0$. In essence, the sum of t along any variable holding the others constant is 0.

The set of variables $\mathcal{Y}$ covered by t is denoted by $\text{Var}(t) = \mathcal{Y}$.

A correlation term may also occasionally be referred to as a "correction term" due to its use in "correcting" joint probability distributions.

Definition 7.2. A **layered probability model** consists of a set T of correlation terms that describe a probability distribution.

Inside a set T of correlation terms, all correlation terms will be assumed to be distinguishable from each other even if they cover the same variables and have the same values.

Given a set of correlation terms T, the marginal probability distribution over $\mathcal{Y} \subseteq \mathcal{X}$ generated by the **layered probability model** with correlation term set T, denoted by $\Pr(V_{\mathcal{Y}} : T)$ for each $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$, is recursively defined by the following:

- Any correlation term $t \in T$ for which $\text{Var}(t) \not\subseteq \mathcal{Y}$ has no impact on $\Pr(V_{\mathcal{Y}} : T)$.
- If there exists no correlation term $t \in T$ for which $\text{Var}(t) \subseteq \mathcal{Y}$, then

$$\Pr(V_{\mathcal{Y}} : T) = \frac{1}{\prod_{x \in \mathcal{Y}} |\text{Val}(x)|}$$

- If there exists a correlation term $t \in T$ for which $\text{Var}(t) \subseteq \mathcal{Y}$, then

$$\Pr(V_{\mathcal{Y}} : T) = \Pr(V_{\mathcal{Y}} : T \setminus \{t\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \text{Var}(t)] : T \setminus \{t\}) t(V_{\mathcal{Y}}[\text{Var}(t)])$$

In essence the total probability distribution can be constructed by sequentially adding each correlation term in sequence.

As an example, there is the following variable set $\mathcal{Y} = \{a, b, c\}$ where $|\text{Val}(a)| = 4$; $|\text{Val}(b)| = 2$; and $|\text{Val}(c)| = 3$. There is the following correlation terms:

$$T = \{t_a(a), t_b(b), t_c(c), t_{a,b}(a, b), t_{a,c}(a, c), t_{b,c}(b, c), t_{a,b,c}(a, b, c)\}$$

The joint probability distribution $\Pr(\langle v_a, v_b, v_c \rangle : T)$ can be generated as follows:

$$\Pr(\langle v_a, v_b, v_c \rangle : \emptyset) = \frac{1}{4 \cdot 2 \cdot 3} = \frac{1}{24}$$

$$\Pr(\langle v_a, v_b, v_c \rangle : \{t_a\}) = \frac{1}{24} + \frac{1}{6} t_a(v_a) = (\frac{1}{4} + t_a(v_a)) \frac{1}{6}$$

Note that $\Pr(v_a : \{t_a\}) = \frac{1}{4} + t_a(V[a])$ and $\Pr(\langle v_b, v_c\rangle : \{t_a\}) = \frac{1}{6}$.

$$\begin{aligned}\Pr(\langle v_a, v_b, v_c\rangle : \{t_a, t_b\}) =&(\frac{1}{4} + t_a(v_a))\frac{1}{6} + ((\frac{1}{4} + t_a(v_a))\frac{1}{3})t_b(v_b)\\ =&(\frac{1}{4} + t_a(v_a))(\frac{1}{2} + t_b(v_b))\frac{1}{3}\end{aligned}$$

Note that $\Pr(v_a : \{t_a, t_b\}) = \frac{1}{4} + t_a(v_a)$, $\Pr(v_b : \{t_a, t_b\}) = \frac{1}{2} + t_b(v_b)$, and $\Pr(v_c : \{t_a, t_b\}) = \frac{1}{3}$.

$$\begin{aligned}\Pr(\langle v_a, v_b, v_c\rangle : \{t_a, t_b, t_c\}) =&(\frac{1}{4} + t_a(v_a))(\frac{1}{2} + t_b(v_b))\frac{1}{3} + (\frac{1}{4} + t_a(v_a))(\frac{1}{2} + t_b(v_b))t_c(v_c)\\ =&(\frac{1}{4} + t_a(v_a))(\frac{1}{2} + t_b(v_b))(\frac{1}{3} + t_c(v_c))\end{aligned}$$

Note that $\Pr(v_a : \{t_a, t_b, t_c\}) = \frac{1}{4} + t_a(v_a)$, $\Pr(v_b : \{t_a, t_b, t_c\}) = \frac{1}{2} + t_b(v_b)$, and $\Pr(v_c : \{t_a, t_b, t_c\}) = \frac{1}{3} + t_c(v_c)$.

To simplify subsequent expressions, let $P_a(v_a) = \frac{1}{4} + t_a(v_a)$; $P_b(v_b) = \frac{1}{2} + t_b(v_b)$; and $P_c(v_c) = \frac{1}{3} + t_c(v_c)$.

$$\begin{aligned}\Pr(\langle v_a, v_b, v_c\rangle : \{t_a, t_b, t_c, t_{a,b}\}) =&P_a(v_a)P_b(v_b)P_c(v_c) + t_{a,b}(a, b)P_c(v_c)\\ =&(P_a(v_a)P_b(v_b) + t_{a,b}(a, b))P_c(v_c)\end{aligned}$$

$$\Pr(\langle v_a, v_b, v_c\rangle : \{t_a, t_b, t_c, t_{a,b}, t_{a,c}\}) = P_a(v_a)P_b(v_b)P_c(v_c) + t_{a,b}(a, b)P_c(v_c) + t_{a,c}(a, c)P_b(v_b)$$

$$\Pr(\langle v_a, v_b, v_c\rangle : \{t_a, t_b, t_c, t_{a,b}, t_{a,c}, t_{b,c}\}) =$$

$$P_a(v_a)P_b(v_b)P_c(v_c) + t_{a,b}(a,b)P_c(v_c) + t_{a,c}(a,c)P_b(v_b) + t_{b,c}(b,c)P_a(v_a)$$

In total,

$$\Pr(\langle v_a, v_b, v_c\rangle : \{t_a, t_b, t_c, t_{a,b}, t_{a,c}, t_{b,c}, t_{a,b,c}\}) =$$

$$P_a(v_a)P_b(v_b)P_c(v_c) + t_{a,b}(a,b)P_c(v_c) + t_{a,c}(a,c)P_b(v_b) + t_{b,c}(b,c)P_a(v_a) + t_{a,b,c}(a,b,c)$$

The joint probability distribution $\Pr(V_{\mathcal{Y}} : T)$ does not depend on the ordering of T used for the recursive definition. To establish this fact, the following theorem will be proven:

Theorem 7.3. *Given a set $\mathcal{Y}$ of variables, and a set of correlation terms T, then for each $V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y})$, it is the case that:*

$$\Pr(V_{\mathcal{Y}} : T) = \sum_{S \in \mathrm{partition}(T:\mathcal{Y})} \frac{1}{\prod_{x \in \mathcal{Y} \setminus \mathrm{Var}(S)} |\mathrm{Val}(x)|} \prod_{t \in S} t(V_{\mathcal{Y}}[\mathrm{Var}(t)])$$

where $\mathrm{partition}(T : \mathcal{Y})$ *is the set of all subsets (including the empty set) of T such that for any $S \in \mathrm{partition}(T : \mathcal{Y})$, it is the case that*

$$\forall t \in S : \mathrm{Var}(t) \subseteq \mathcal{Y}$$

and

$$\forall t_1, t_2 \in S : t_1 \neq t_2 \implies \mathrm{Var}(t_1) \cap \mathrm{Var}(t_2) = \emptyset$$

Note that $\emptyset \in \mathrm{partition}(T : \mathcal{Y})$, and that the empty product is 1.

For any $S \subseteq T$, it is defined that $\mathrm{Var}(S) = \bigcup_{t \in S} \mathrm{Var}(t)$ (note that the empty union is $\emptyset$)

The proof will be given in the appendix.

In the above formulation for $\Pr(V_{\mathcal{Y}} : T)$, there is no preferred ordering for the correlation terms from T.

Of equal importance is that the definition of the probability distribution generated by a set of correlation terms is consistent under the operation of marginalization:

Theorem 7.4. *Given a set $\mathcal{Y}$ of variables, and a set of correlation terms T, then for any subset of variables $\mathcal{Z} \subset \mathcal{Y}$,*

$$\forall V_{\mathcal{Z}} \in \mathrm{Val}(\mathcal{Z}) : \Pr(V_{\mathcal{Z}} : T) = \sum_{V' \in \mathrm{Val}(\mathcal{Y} \setminus \mathcal{Z})} \Pr(\langle V_{\mathcal{Z}}, V' \rangle : T)$$

The proof will be given in the appendix.

Another important result is that if there are two correlation terms $t_1, t_2 \in T$ such that $\mathrm{Var}(t_1) = \mathrm{Var}(t_2)$, then t_1 and t_2 can be removed from T and replaced by $t_1 + t_2$ without influencing the probability $\Pr(V_{\mathcal{Y}} : T)$.

Theorem 7.5. *Let $\mathcal{Y}$ be an arbitrary set of variables, and let T be an arbitrary set of correlation terms. If there exists two correlation terms $t_1, t_2 \in T$ such that $\mathrm{Var}(t_1) = \mathrm{Var}(t_2)$, then removing t_1 and t_2 and adding $t_1 + t_2$ to T to get $T' = (T \setminus \{t_1, t_2\}) \cup \{t_1 + t_2\}$ will not change $\Pr(V_{\mathcal{Y}} : T)$ for all assignments $V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y})$:*

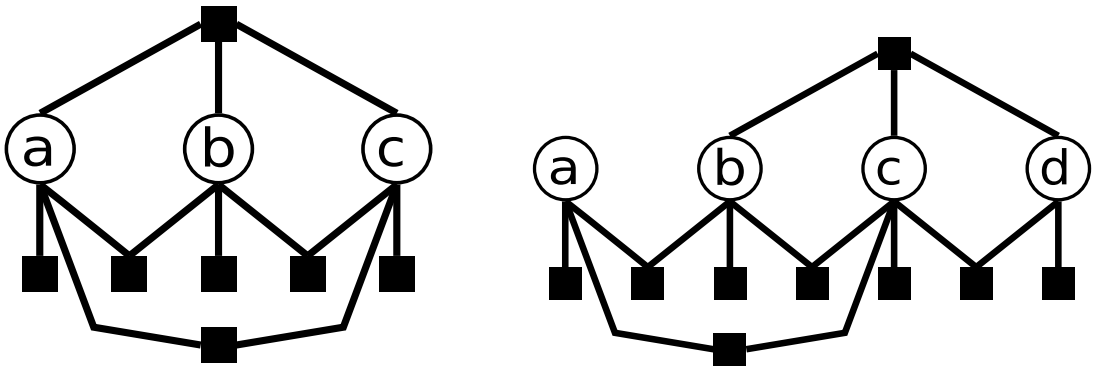
$$\forall V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}} : T) = \Pr(V_{\mathcal{Y}} : T')$$

The proof will be given in the appendix.

7.3 The graphical model

Given a set of variables $\mathcal{Y}$ and a set of correlation terms T, a bipartite graph can be created that depicts both of the variables and correlation terms. The bipartite graph is defined as follows:

Definition 7.6. Given a set of variables $\mathcal{Y}$ and a set of correlation terms T, the **correlation graph** $G(\mathcal{Y} : T)$ is bipartite graph where the variables from $\mathcal{Y}$ form the vertices/nodes from one partition, and the correlation terms from T form the vertices/nodes from the other partition. Given a variable $x \in \mathcal{Y}$ and correlation term $t \in T$, an edge exists between x and t if and only if $x \in \text{Var}(t)$.



(a) (b)

Figure 7.2: (a) The bipartite graph that depicts a 3 variable model where all correlation terms exist. (b) A bipartite graph that depicts a 4 variable model where not all correlation terms exist.

Figure 7.2 depicts 2 example correlation graphs. The correlation graph on the left depicts a simple 3 variable model ($\mathcal{Y} = \{a, b, c\}$) where all correlation terms exist: $T = \{t_a(a), t_b(b), t_c(c), t_{a,b}(a, b), t_{a,c}(a, c), t_{b,c}(b, c), t_{a,b,c}(a, b, c)\}$. The correlation graph on the right depicts a 4 variable model ($\mathcal{Y} = \{a, b, c, d\}$) where the existing correlation terms are: $T = \{t_a(a), t_b(b), t_c(c), t_d(d), t_{a,b}(a, b), t_{b,c}(b, c), t_{c,d}(c, d), t_{a,c}(a, c), t_{b,c,d}(b, c, d)\}$.

The correlation graph can be used to infer variable independence relationships. If a variable x is not under consideration and is eliminated via marginalization, then x can be removed from the correlation graph alongside all correlation terms that cover x. If the correlation graph can be partitioned into separate disconnected path components (a path component is a sub-graph where any pair of nodes is connected by at least one path), then the variables sets from each path component are mutually independent.

Theorem 7.7. *Let $G(\mathcal{Y} : T)$ denote an arbitrary correlation graph over the set of variables $\mathcal{Y}$ with the set of correlation terms T. If the variable set $\mathcal{Y}$ can be partitioned into disjoint subsets $\mathcal{Y}_1, \mathcal{Y}_2, \ldots \mathcal{Y}_n$ where $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \cdots \cup \mathcal{Y}_n$, and for any $i \neq j$ there are no paths that link $\mathcal{Y}_i$ to $\mathcal{Y}_j$, then $\mathcal{Y}_1 \perp \mathcal{Y}_2 \perp \cdots \perp \mathcal{Y}_n$.*

The proof will be given in the appendix.

As an example, from the correlation graph in figure 7.2(b), it can be seen $\{a, b\} \perp d$. Variable sets $\{a, b\}$ and d become separated when variable c is eliminated via marginalization, which eliminates node c and all correlation terms connected to c.

7.4 Computing the correlation terms

Knowing beforehand the probabilities $\Pr(V_{\mathcal{X}})$ for all assignments $V_{\mathcal{X}} \in \mathrm{Val}(\mathcal{X})$, the correlation terms can be computed recursively from the terms that cover small numbers of variables up to terms that cover large numbers of variables. This done according to the following algorithm:

Correlation term generation algorithm 1:

Input: An arbitrary set $\mathcal{X}$ of variables, and the complete probability distribution over the assignments from $\mathrm{Val}(\mathcal{X})$.

$T \leftarrow \emptyset$ `// The set of correlation terms starts off empty.`

for $n = 1$ to $|\mathcal{X}|$ **do**

`// All correlation terms that cover` n `variables will be computed.`

for every subset $\mathcal{Y} \subseteq \mathcal{X}$ such that $|\mathcal{Y}| = n$ **do**

`// The correlation term for variable set` $\mathcal{Y}$ `will be computed.`

Compute the marginal probability distribution $\Pr(V_{\mathcal{Y}} : T)$ over the variables from $\mathcal{Y}$ using only the correlation terms that are currently present in T.

Compute the true marginal probability distribution $\Pr(V_{\mathcal{Y}})$ using the input data.

$\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : t_{\mathcal{Y}}(V_{\mathcal{Y}}) \leftarrow \Pr(V_{\mathcal{Y}}) - \Pr(V_{\mathcal{Y}} : T)$ `// Assign values to the new correlation term.`

$T \leftarrow T \cup \{t_{\mathcal{Y}}\}$ `// Add the new correlation term to the set.`

end for

end for

return: The set T of correlation terms.

By computing the correlation terms over smaller sets of variables before computing the correlation terms over the larger sets of variables, it is guaranteed that each correlation term t satisfies the important property that $\forall x \in \text{Var}(t) : \forall V \in \text{Val}(\text{Var}(t) \setminus x) : \sum_{v_x \in \text{Val}(x)} t(\langle v_x, V \rangle) = 0$.

Theorem 7.8. ***Correlation term generation algorithm 1*** *generates valid correlation terms:*

$$\forall t \in T : \forall x \in \text{Var}(t) : \forall V \in \text{Val}(\text{Var}(t) \setminus x) : \sum_{v_x \in \text{Val}(x)} t(\langle v_x, V \rangle) = 0$$

The proof will be given in the appendix.

Correlation term generation algorithm 1 extracts a correlation term for every possible subset of variables. In practice however, a user may wish to ignore some correlation terms. In these cases, subsequently generated correlation terms may not satisfy the properties required of correlation terms. When this occurs, these correlation terms must be "normalized" so that summing with respect to any single variable always totals to 0. Contrary to the term "normalization", the values are not scaled, but are instead shifted. Algorithm **Correlation term normalization algorithm**, described below, "normalizes" an input correlation term so that summing with respect to any single variable always totals to 0.

Correlation term normalization algorithm:

Input: A arbitrary subset $\mathcal{Y}$ of variables, and a function $t_0 : \text{Val}(\mathcal{Y}) \rightarrow \mathbb{R}$.

`// Function` t_0 `may not be a valid correlation term and may have to be adjusted so that the result is 0 when any variable is summed along.`

Initialize $t \leftarrow t_0$

for every variable x from $\mathcal{Y}$ **do**

for every assignment V from $\text{Val}(\mathcal{Y} \setminus x)$ **do**

Compute the sum $S = \sum_{v_x \in \text{Val}(x)} t(\langle v_x, V \rangle)$

for every assignment v_x from $\text{Val}(x)$ **do**

$t(\langle v_x, V \rangle) \leftarrow t(\langle v_x, V \rangle) - \frac{S}{|\text{Val}(x)|}$

end for

end for

end for

return: The new correlation term t.

The function call will be denoted by **normalize**(t_0) ($\mathcal{Y}$ can be inferred from $\text{Var}(t_0)$).

In many cases, a user may wish to heuristically ignore correlation terms that are "unimportant". To decide whether a correlation term is important or not, the concept of "Shannon entropy" will be employed. The concept of Shannon entropy is reviewed in section 3.5. In this context, differences in the Shannon entropy will be used to determine which correlation terms to keep, and which terms to ignore. Below is shown a variant of **Correlation term generation algorithm 1** that omits correlation terms that fail to reduce the entropy by a specified amount ϵ:

Correlation term generation algorithm 2:

Input: An arbitrary set $\mathcal{X}$ of variables, the complete probability distribution over the assignments from $\text{Val}(\mathcal{X})$, and a tolerance quantity ϵ.

$T \leftarrow \emptyset$ `// The set of correlation terms starts off empty.`

for $n = 1$ to $|\mathcal{X}|$ **do**

`// All correlation terms that cover n variables will be computed, but some`

`may be omitted.`

for every subset $\mathcal{Y} \subseteq \mathcal{X}$ such that $|\mathcal{Y}| = n$ **do**

`// The correlation term for variable set` $\mathcal{Y}$ `will be computed.`

Compute the marginal probability distribution $\Pr(V_{\mathcal{Y}} : T)$ over the variables from $\mathcal{Y}$ using only the correlation terms that are currently present in T.

Compute the true marginal probability distribution $\Pr(V_{\mathcal{Y}})$ using the input data.

$\forall V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y}) : t_{\mathcal{Y}}(V_{\mathcal{Y}}) \leftarrow \Pr(V_{\mathcal{Y}}) - \Pr(V_{\mathcal{Y}} : T)$ `// Assign values to the new correlation term.`

$t_{\mathcal{Y}} \leftarrow$ **normalize**$(t_{\mathcal{Y}})$ `// Normalize the correlation terms to account for previous correlation terms that were omitted.`

Compute the the total probability distributions $\Pr(V_{\mathcal{X}} : T)$ and $\Pr(V_{\mathcal{X}} : T \cup \{t\})$ and compute the entropy difference: $\Delta\mathbf{H} = \mathbf{H}(\Pr(V_{\mathcal{X}} : T \cup \{t\})) - \mathbf{H}(\Pr(V_{\mathcal{X}} : T))$ `// The entropy difference is ideally negative.`

if $\Delta\mathbf{H} < -\epsilon$ **then**

$T \leftarrow T \cup \{t\}$ `// Add the new correlation term to the set only if the entropy decrease is significant enough.`

end if

end for

end for

return: The set T of correlation terms.

7.5 Variable conditioning

In section 7.3, the process of eliminating a variable via marginalization is as simple as eliminating all correlation terms that contain said variable, which will also probably break the correlation graph into several path components. Conditioning variables given evidence is a significantly less simple task however. This is the opposite of Markov networks where

conditioning variables is a trivial matter, and it is marginalization that is computationally prohibitive.

While a computationally efficient algorithm for conditioning variables is not yet developed for layered probability models, a direct algorithm for probabilistic inference will be given here. A computationally efficient algorithm for variable conditioning is the subject of future work.

Probabilistic inference using layered probability models:

Input: An arbitrary set of $\mathcal{X}$ variables, a set T of correlation terms, a set $\mathcal{Y}$ of query variables, a set $\mathcal{Z}$ of observed variables where $\mathcal{Y} \cap \mathcal{Z} = \emptyset$, and observations $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$.

Eliminate from T all correlation terms $t \in T$ where $\text{Var}(t) \not\subseteq \mathcal{Y} \cup \mathcal{Z}$.

Generate the total marginal probability distribution $\Pr(V_{\mathcal{Y}\cup\mathcal{Z}} : T)$ over the variables from $\mathcal{Y} \cup \mathcal{Z}$ using only the correlation terms that remain in T.

Compute the posterior probability distribution $\Pr(V_{\mathcal{Y}}|V_{\mathcal{Z}})$ over the variables from $\mathcal{Y}$ using Bayes' rule:

$\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}}|V_{\mathcal{Z}}) = \frac{\Pr(\langle V_{\mathcal{Y}}, V_{\mathcal{Z}}\rangle)}{\Pr(V_{\mathcal{Z}})}$

return: The posterior probability distribution $\Pr(V_{\mathcal{Y}}|V_{\mathcal{Z}})$ over the variables from $\mathcal{Y}$.

7.6 Computational Example

This section will give an example of generating a layered probability model from an arbitrary probability distribution. 4 abstract random variables A, B, C, D will be used, with the respective domains $\text{Val}(A) = \{a_1, a_2, a_3\}$; $\text{Val}(B) = \{b_1, b_2\}$; $\text{Val}(C) = \{c_1, c_2, c_3\}$; and $\text{Val}(D) = \{d_1, d_2\}$. The C++ code that is used to perform the process of generating the layered probability model can be found at

`https://github.com/sceastwo/layered_approximations_program_files.git` [2].

7.6.1 Generating the layered model

The complete probability distribution is:

C		c_1		c_2		c_3	
D		d_1	d_2	d_1	d_2	d_1	d_2
A	B	$\Pr(A, B, C, D)$					
a_1	b_1	0.000082	0.036725	0.012596	0.052700	0.038121	0.031270
a_1	b_2	0.022826	0.058384	0.053619	0.048651	0.011345	0.055972
a_2	b_1	0.046299	0.033464	0.019809	0.000976	0.005956	0.023749
a_2	b_2	0.009599	0.010811	0.064416	0.029043	0.007760	0.000304
a_3	b_1	0.000581	0.024624	0.034645	0.037220	0.039213	0.039565
a_3	b_2	0.010832	0.043206	0.029375	0.022945	0.003717	0.039599

The *complete set* of resultant correlation terms are:

$t_{0,0,0,1}(D)$

D	d_1	d_2
$t_{0,0,0,1}(D)$	-0.089209	0.089209

$t_{0,0,1,0}(C)$

C	c_1	c_2	c_3
$t_{0,0,1,0}(C)$	-0.035900	0.072662	-0.036762

$t_{0,0,1,1}(C, D)$

C	c_1		c_2		c_3	
D	d_1	d_2	d_1	d_2	d_1	d_2
$t_{0,0,1,1}(C, D)$	-0.031964	0.031964	0.047681	-0.047681	-0.015717	0.015717

$t_{0,1,0,0}(B)$

B	b_1	b_2
$t_{0,1,0,0}(B)$	-0.022405	0.022405

$t_{0,1,0,1}(B, D)$

B	b_1		b_2	
D	d_1	d_2	d_1	d_2
$t_{0,1,0,1}(B,D)$	0.001110	-0.001110	-0.001110	0.001110

$t_{0,1,1,0}(B,C)$

B	b_1			b_2		
C	c_1	c_2	c_3	c_1	c_2	c_3
$t_{0,1,1,0}(B,C)$	-0.000278	-0.035955	0.036233	0.000278	0.035955	-0.036233

$t_{0,1,1,1}(B,C,D)$

C	c_1		c_2		c_3	
D	d_1	d_2	d_1	d_2	d_1	d_2
B	$t_{0,1,1,1}(B,C,D)$					
b_1	0.003658	-0.003658	-0.021056	0.021056	0.017398	-0.017398
b_2	-0.003658	0.003658	0.021056	-0.021056	-0.017398	0.017398

$t_{1,0,0,0}(A)$

A	a_1	a_2	a_3
$t_{1,0,0,0}(A)$	0.088958	-0.081147	-0.007811

$t_{1,0,0,1}(A,D)$

A	a_1		a_2		a_3	
D	d_1	d_2	d_1	d_2	d_1	d_2
$t_{1,0,0,1}(A,D)$	-0.034885	0.034885	0.050243	-0.050243	-0.015359	0.015359

$t_{1,0,1,0}(A,C)$

C	c_1	c_2	c_3
A	$t_{1,0,1,0}(A,C)$		
a_1	-0.007586	-0.003882	0.011469
a_2	0.025165	0.011858	-0.037022
a_3	-0.017578	-0.007975	0.025554

$t_{1,0,1,1}(A, C, D)$

C	c_1		c_2		c_3	
D	d_1	d_2	d_1	d_2	d_1	d_2
A	$t_{1,0,1,1}(A, C, D)$					
a_1	-0.001698	0.001698	-0.008592	0.008592	0.010290	-0.010290
a_2	0.007865	-0.007865	0.004872	-0.004872	-0.012736	0.012736
a_3	-0.006166	0.006166	0.003720	-0.003720	0.002446	-0.002446

$t_{1,1,0,0}(A, B)$

A	a_1		a_2		a_3	
B	b_1	b_2	b_1	b_2	b_1	b_2
$t_{1,1,0,0}(A, B)$	-0.030190	0.030190	0.009810	-0.009810	0.020380	-0.020380

$t_{1,1,0,1}(A, B, D)$

B	b_1		b_2	
D	d_1	d_2	d_1	d_2
A	$t_{1,1,0,1}(A, B, D)$			
a_1	-0.003457	0.003457	0.003457	-0.003457
a_2	-0.005719	0.005719	0.005719	-0.005719
a_3	0.009176	-0.009176	-0.009176	0.009176

$t_{1,1,1,0}(A, B, C)$

B	b_1			b_2		
C	c_1	c_2	c_3	c_1	c_2	c_3
A	$t_{1,1,1,0}(A, B, C)$					
a_1	-0.010461	0.012708	-0.002247	0.010461	-0.012708	0.002247
a_2	0.029073	-0.028693	-0.000380	-0.029073	0.028693	0.000380
a_3	-0.018612	0.015985	0.002628	0.018612	-0.015985	-0.002628

$t_{1,1,1,1}(A, B, C, D)$

C		c_1		c_2		c_3	
D		d_1	d_2	d_1	d_2	d_1	d_2
A	B	$t_{1,1,1,1}(A,B,C,D)$					
a_1	b_1	-0.004447	0.004447	-0.002681	0.002681	0.007128	-0.007128
a_1	b_2	0.004447	-0.004447	0.002681	-0.002681	-0.007128	0.007128
a_2	b_1	0.007484	-0.007484	0.002303	-0.002303	-0.009787	0.009787
a_2	b_2	-0.007484	0.007484	-0.002303	0.002303	0.009787	-0.009787
a_3	b_1	-0.003038	0.003038	0.000379	-0.000379	0.002659	-0.002659
a_3	b_2	0.003038	-0.003038	-0.000379	0.000379	-0.002659	0.002659

When a minimum entropy reduction requirement of ϵ is introduced, correlation terms begin to be omitted, and the remaining terms are changed somewhat. Below is a table that lists the correlation terms that exist for various values of ϵ, and the L1 distance between the original probability distribution, $\Pr_{\text{true}}$, and the probability distribution generated from the reduced set of correlation terms, $\Pr_{\text{reduced}}$. The L1 distance used here is the average absolute distance between probability values: the sum of all absolute probability differences divided by the domain size $\frac{1}{|\text{Val}(\{A,B,C,D\})|}\sum_{V\in\text{Val}(\{A,B,C,D\})}|\Pr_{\text{true}}(V)-\Pr_{\text{reduced}}(V)|$.

In the table a string of 15 bits will denote the inclusion of the 15 possible correlation terms in the order $t_{0,0,0,1}(D)$, $t_{0,0,1,0}(C)$, $t_{0,0,1,1}(C,D)$, $t_{0,1,0,0}(B)$, $t_{0,1,0,1}(B,D)$, $t_{0,1,1,0}(B,C)$, $t_{0,1,1,1}(B,C,D)$, $t_{1,0,0,0}(A)$, $t_{1,0,0,1}(A,D)$, $t_{1,0,1,0}(A,C)$, $t_{1,0,1,1}(A,C,D)$, $t_{1,1,0,0}(A,B)$, $t_{1,1,0,1}(A,B,D)$, $t_{1,1,1,0}(A,B,C)$, and $t_{1,1,1,1}(A,B,C,D)$

ϵ	inclusion flags	L1($\Pr_{\text{true}}, \Pr_{\text{reduced}}$)
0.0001	111101111101111	0.00324365
0.001	111101111101111	0.00324365
0.01	111001111101011	0.00386915
0.02	101001111110010	0.00678593
0.03	000000011100010	0.0105278
0.04	000000001000010	0.0119901
0.05	000000000000010	0.0139454
0.1	000000000000000	0.0157617

7.6.2 Probabilistic inference example

This section will give a simple example of probabilistic inference using the probability distribution $\Pr_{\text{reduced}}$ that was generated from the correlation terms acquired with various values of ϵ.

Given a scenario where $A = a_2$, then the posterior probability distributions for D given various values of ϵ are given below:

ϵ	$\Pr_{\text{reduced}}(d_1 \mid a_2)$	$\Pr_{\text{reduced}}(d_2 \mid a_2)$
all terms are present	0.610022	0.389978
0.0001	0.610022	0.389978
0.001	0.610022	0.389978
0.01	0.610022	0.389978
0.02	0.610022	0.389978
0.03	0.727936	0.272064
0.04	0.672447	0.327553
0.05	0.500000	0.500000
0.1	0.500000	0.500000

7.7 Conclusions and Future Work

This chapter has introduced a possible alternative to Markov networks, which were referred to as "layered probability models". The layered probability models were rigorously defined, and the mathematics related to layered probability models was established. A working algorithm for generating layered probability models from raw statistical data was provided and demonstrated on an arbitrarily generated probability distribution.

From the brief description of the layered probability models, there exist many questions that need to be addressed before the layered probability models become a feasible alternative to Markov networks, Bayesian networks, etc.. These questions and areas of investigation include:

- **Computationally efficient algorithms for generating the total marginal probability distribution**: While trimming away correlation terms that have no impact on the sought after marginal probability distribution is a trivial matter, generating the total probability distribution if a large number of variables and correlation terms are present in the posterior probability distribution can be a computationally challenging task.

- **Modifying the model to account for variable conditioning**: Unlike marginalization, variable conditioning cannot be performed by merely conditioning correlation terms. Without computationally efficient algorithms for variable conditioning, the best that can be done for probabilistic inference is to first eliminate unused variables via marginalization, and then compute the total marginal probability distribution and then proceed with conditioning via a straightforwards application of Bayes' rule.

Chapter 8

Concluding Remarks

Motivated by the need for computationally efficient, robust, and mathematically sound risk analysis tools for biometric enabled authentication systems, this thesis has provided the following solutions:

- An approximate approach to probabilistic inference that breaks a large Bayesian network into "modules" for the purpose of reducing the computational complexity of probabilistic inference. Benefits include in particular:
  - The limited size of each module enables it to be constructed with limited statistical data.
  - Modules can be linked to form larger "modular models". The modules can be chosen from a pre-compiled library.
  - Probabilistic inference performed on modular models has a computational complexity that is linear with respect to the number of modules.
- The application of various uncertainty metrics to causal networks using the framework of a "unified inference engine". The unified inference engine provides:
  - A variety of interpretations of uncertainty.

  - The flexibility of choosing an uncertainty metric to best match limitations on the input statistical data and requirements on the output data.
  - A variety of posterior uncertainty models that when taken together form a more thorough picture of the current scenario.
  - The possibility of "mixing" the uncertainty metrics in the same causal network as an avenue for future work.

- An expansive taxonomy of information fusion approaches that are based on sets of probability distributions (credal sets). This taxonomy will enable practitioners to select an appropriate fusion approach based on the following criteria: the availability of statistical data to utilize context specific fusion instead of general fusion; the complexity of the credal set (probability interval distribution or Dempster-Shafer model); the desired "tightness" of the output credal set; and the computational complexity of the fusion algorithm. In addition:
  - A distinction between "context specific fusion", and "general fusion" is made. Both approaches require different amounts of statistical data to use, and accept different forms of input. Context specific fusion should be chosen when there is plenty of statistical data to generate the prior credal set, and to generate the likelihood intervals when the observations have been made at each sensor. General fusion should be chosen when statistical data is lacking and a direct fusion of credal sets is necessary.
  - The introduction of the "containment property" as a mathematically objective requirement for fusion using uncertainty metrics that correspond to credal sets.

- The presentation of potentially effective alternative graphical models, referred to as "layered probability models", that can be used in place of Markov and Bayesian networks. This alternative has the following benefits:

- There exists a simple algorithm for creating the model from raw statistical data.
- The process of marginalizing variables is trivial.

To establish the layered probability models as an effective alternative to Markov and Bayesian networks, the following problems will need to be addressed in future work:

- The creation of computationally efficient algorithms for generating total and marginal probability distributions from a set of correlation terms. The current algorithm for generating a total probability distribution is computationally inefficient if a complete set of correlation terms is present.
- The incorporation of a framework for variable conditioning: Opposite to Markov networks, variable conditioning with layered probability models is difficult while marginalization is trivial. For layered probability models to be an effective alternative, variable conditioning needs to made faster than the straightforward application of Bayes' rule.

These proposed solutions form computationally efficient, robust, and mathematically sound risk analysis tools for biometric enabled authentication systems.

Bibliography

[1] ARENA: Simulation software. Rockwell Automation.

[2] C++ code "Layered approximations program files". Git Hub repository at: https://github.com/sceastwo/layered_approximations_program_files.git.

[3] *IATA: Automated Border Control. Implementation Guide.*

[4] Software package "Causal Networks and Uncertainty Metrics". Git Hub repository at: https://github.com/sceastwo/Causal_Networks_and_Uncertainty_Metrics.git.

[5] Software package "Dempster-Shafer Bayesian Network (DSBN-01)". Biometrics Laboratory, University of Calgary, Canada, http://www.ucalgary.ca/btlab.

[6] *IATA: Checkpoint of the future. Executive summary. 4th Proof.*, 2014.

[7] A. N. Al-Raisi and A. M. Al-Khouri. Iris recognition and the challenge of homeland and border control security in uae. *Telematics and Informatics*, 25(2):117–132, 2008.

[8] S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen. Hugin–a shell for building bayesian belief universes for expert systems. In *International Joint Conference on Artificial Intelligence*, volume 89, pages 1080–1085, 1989.

[9] A. Antonucci, C. P. De Campos, D. Huber, and M. Zaffalon. Approximating credal network inferences by linear programming. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 13–24, Berlin Heidelberg, 2013. Springer.

[10] A. Antonucci, D. Huber, M. Zaffalon, P. Luginbuhl, I. Chapman, and R. Ladouceur. CREDO: a military decision-support system based on credal networks. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1942–1949, 2013.

[11] J. F. Baldwin and E. D. Tomaso. Inference and learning in fuzzy bayesian networks. In *Proc. 12th IEEE Int. Conf. Fuzzy Systems*, volume 1, pages 630–635, 2003.

[12] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

[13] A. Cano, M. Gomez, S. Moral, and J. Abellan. Hill-climbing and branch-and-bound algorithms for exact and approximate inference in credal networks. *International Journal of Approximate Reasoning*, 44(3):261–280, 2007.

[14] H. Chan and A. Darwiche. Sensitivity analysis in bayesian networks: From single to multiple parameters. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 67–75. AUAI Press, 2004.

[15] F. G. Cozman. Credal networks. *Artificial intelligence*, 120(2):199–233, 2000.

[16] F. G. Cozman. Graphical models for imprecise probabilities. *International Journal of Approximate Reasoning*, 39(2):167–184, 2005.

[17] L. M. De Campos, J. F. Huete, and S. Moral. Probability intervals: a tool for uncertain reasoning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2(2):167–196, 1994.

[18] F. Delmotte and P. Smets. Target identification based on the transferable belief model interpretation of Dempster-Shafer model. *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(4):457–471, 2004.

[19] J. Dezert and F. Smarandache. On the generation of hyper-powersets for the DSmT. arXiv preprint math/0309431, 2003.

[20] J. Dezert, A. Tchamova, D. Han, and J. M. Tacnet. Why Dempster's fusion rule is not a generalization of Bayes fusion rule. *Proc. 16th IEEE Int. Conf. Information Fusion*, pages 1127–1134, 2013.

[21] J. Dezert, A. Tchamova, F. Smarandache, and P. Konstantinova. Target type tracking with PCR5 and Dempster's rules: a comparative analysis. In *9th International Conference on Information Fusion*. IEEE, 2006.

[22] S. C. Eastwood and S. N. Yanushkevich. Risk assessment in authentication machines. In R. Abielmona, R. Falcon, N. Zincir-Heywood, and H. Abbas, editors, *Recent Advances in Computational Intelligence in Defense and Security, Springer series on Studies in Computational Intelligence*, pages 391–420. Springer, 2016.

[23] EU, European Commission, B-1049, Brussels. *European Union: Smart Borders*, 2014. http://ec.europa.eu/dgs/home-affairs/what-we-do/policies/borders-and-visas/smart-borders/index_en.htm.

[24] EU, European Commission, B-1049, Brussels. *European Union: Technical Study on Smart Borders*, 2014.

[25] P. Guo and H. Tanaka. Decision making with interval probabilities. *European Journal of Operational Research*, 203(2):444–454, 2010.

[26] R. Haenni. Ordered valuation algebras: a generic framework for approximating inference. *International Journal of Approximate Reasoning*, 37(1):1–41, 2004.

[27] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices, 1971.

[28] K. Hwang and S. Cho. Landmark detection from mobile life log using a modular bayesian network model. *Expert Systems with Applications*, 36(10):12065–12076, 2009.

[29] F. V. Jensen, K. G. Olesen, and S. K. Andersen. An algebra of bayesian belief universes for knowledge based systems. *Networks*, 20(5):637–659, 1990.

[30] R. Jirousek. An attempt to define graphical models in dempster-shafer theory of evidence. *Combining Soft Computing and Statistical Methods in Data Analysis*, pages 361–368, 2010.

[31] R. Jirousek. Local computations in dempster–shafer theory of evidence. *International Journal of Approximate Reasoning*, 53(8):1155–1167, 2012.

[32] A. Karlsson. *Evaluating credal set theory as a belief framework in high-level information fusion for automated decision-making*. PhD thesis, Orebro University, 2010.

[33] A. Karlsson, R. Johansson, and S. F. Andler. Characterization and empirical evaluation of Bayesian and credal combination operators. *Journal of Advances in Information Fusion*, 6(2):150–166, 2011.

[34] A. Karlsson and H. J. Steinhauer. Evaluation of evidential combination operators. In *Proceedings of the 8th International Symposium on Imprecise Probability: Theory and Applications*, volume 13, 2013.

[35] N. Kawasaki and U. Kiencke. Standard platform for sensor fusion on advanced driver assistance system using bayesian network. In *2004 IEEE Intelligent Vehicles Symposium*. IEEE, 2004.

[36] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

[37] G. J. Klir. *Uncertainty and information: foundations of generalized information theory*. John Wiley & Sons, 2005.

[38] J. Kohlas and P. P. Shenoy. Computation in valuation algebras. *Algorithms for Uncertainty and Defeasible Reasoning, Handbook of Defeasible Reasoning and Uncertainty Management Systems*, 5:5–39, 2000.

[39] J. Kohlas and N. Wilson. Semiring induced valuation algebras: Exact and approximate local computation algorithms. *Artificial Intelligence*, 172(11):1360–1399, 2008.

[40] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[41] K. Kristensen and I. A. Rasmussen. The use of a bayesian network in the design of a decision support system for growing malting barley without use of pesticides. *Computers and Electronics in Agriculture*, 33(3):197–217, 2002.

[42] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[43] R. D. Labati, A. Genovese, E. Munoz, V. Piuri, F. Scotti, and G. Sforza. Biometric recognition in automated border control: A survey. *ACM Comp. Surv.*, 49(2):A1–A39, 2016.

[44] K. Laskey. Sensitivity analysis for probability assessments in bayesian networks. *IEEE Transactions on Systems, Man and Cybernetics*, 25(6):901–909, 1995.

[45] K. Murphy. The Bayes Net Toolbox for Matlab. *Computing science and statistics*, 33(2), 2001.

[46] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[47] K. Nguyen, S. Denman, S. Sridharan, and C. Fookes. Score-level multibiometric fusion based on Dempster-Shafer theory incorporating uncertainty factors. *IEEE Transactions on Human-Machine Systems*, 45(1):132–140, 2015.

[48] M. Nuppency. *Automated border control – state of play and latest developments*. Federal Office for Information Security, Germany, 2014.

[49] N. Oliver, A. Garg, and E. Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding*, 96(2):163–180, 2004.

[50] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3):241–288, 1986.

[51] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[52] Y. Qi and T. Minka. Tree-structured approximations by expectation propagation. In *Advances in Neural Information Processing Systems*, pages 193–200, 2004.

[53] J. Ren, I. Jenkinson, J. Wang, D. L. Xu, and J. B. Yang. An offshore risk analysis method using fuzzy Bayesian network. *J. Offshore Mechanics and Arctic Engineering*, 131(4):041101–1–12, 2009.

[54] Research and Development Unit, Warsaw. *Frontex: Best practice technical guidelines for automated border control (ABC) systems*, 2012.

[55] E. Segal and et al. Learning module networks. In *Proc. 19th Conf. Uncertainty in Artificial Intelligence*, pages 525–534, 2002.

[56] G. Shafer, P. P. Shenoy, and K. Mellouli. Propagating belief functions in qualitative markov trees. *International Journal of Approximate Reasoning*, 1(4):349–400, 1987.

[57] G. R. Shafer and P. P. Shenoy. Probability propagation. *Annals of mathematics and Artificial Intelligence*, 2(1–4):327–351, 1990.

[58] C. Simon, P. Weber, and A. Evsukoff. Bayesian networks inference algorithm to implement dempster shafer theory in reliability analysis. *Reliability Engineering and System Safety*, 93:950–963, 2008.

[59] M. Sipser. *Introduction to the Theory of Computation, second edition*. Course Technology, Cengage Learning, 2006.

[60] F. Smarandache and J. Dezert. Information fusion based on new proportional conflict redistribution rules. In *7th International Conference on Information Fusion*. IEEE, 2005.

[61] F. Smarandache and J. Dezert. A simple proportional conflict redistribution rule. *International Journal of Applied Mathematics and Statistics*, 3, 2005.

[62] F. Smarandache, J. Dezert, and J. Tacnet. Fusion of sources of evidence with different importances and reliabilities. In *13th Conference on Information Fusion (FUSION), 2010*. IEEE, 2010.

[63] L. J. Spreeuwers, A. J. Hendrikse, and K. J. Gerritsen. Evaluation of automatic face recognition for automatic border control on actual data recorded of travellers at schiphol airport. In *Proc. IEEE Int. Conf. Biometrics Special Interest Group (BIOSIG)*, pages 1–6, 2012.

[64] A. Tchamova, T. Semerdjiev, and J. Dezert. Estimation of target behavior tendencies using Dezert-Smarandache theory. In *Proceedings of the 6th International Conference on Information Fusion (Fusion 2003)*, pages 1349–1356, 2003.

[65] B. Tessem. Interval probability propagation. *International Journal of Approximate Reasoning*, 7(3–4):95–120, 1992.

[66] W. Van Norden, F. Bolderheij, and C. Jonker. Combining system and user belief on classification using the DSmT combination rule. In *11th International Conference on Information Fusion, 2008*. IEEE, 2008.

[67] F. Voorbraak. On the justification of dempster's rule of combination. *Artificial Intelligence*, 48(2):171–197, 1991.

[68] P. Walley. Measures of uncertainty in expert systems. *Artificial intelligence*, 83(1):1–58, 1996.

[69] M. Welling, T. P. Minka, and Y. W. Teh. Structured region graphs: Morphing EP into GBP. arXiv preprint arXiv:1207.1426, 2012.

[70] N. Yager and T. Dunstone. The biometric menagerie. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):220–230, 2010.

[71] R. R. Yager. On the Dempster-Shafer framework and new combination rules. *Information sciences*, 41(2):93–137, 1987.

[72] R. R. Yager. A framework for multi-source data fusion. *Information Sciences*, 163(1):175–200, 2004.

[73] R. R. Yager. On the determination of strength of belief for decision support under uncertainty – Part II: fusing strengths of belief. *Fuzzy Sets and Systems*, 142(1):129–142, 2004.

[74] R. R. Yager. Human behavioral modeling using fuzzy and Dempster–Shafer theory. *Social Computing, Behavioral Modeling, and Prediction*, pages 89–99, 2008.

[75] R. R. Yager. On the fusion of imprecise uncertainty measures using belief structures. *Information Sciences*, 181(15):3199–3209, 2011.

[76] R. R. Yager and D. P. Filev. Including probabilistic uncertainty in fuzzy logic controller modeling using Dempster-Shafer theory. *IEEE Trans. Systems, Man, and Cybernetics*, 25(8):1221–1230, 1995.

[77] R. R. Yager and L. Liu, editors. *Classic works of the Dempster-Shafer theory of belief functions*, volume 219. Springer, 2008.

[78] R. R. Yager and F. Petry. An intelligent quality-based approach to fusing multi-source probabilistic information. *Information Fusion*, 31:127–136, 2016.

[79] S. Yanushkevich and V. Shmerko. Automated Border Control Part I: Overview, Trends and Challenges. Technical report, Defense Research & Development Canada (DRDC) and Canadian Border Service Agency (CBSA), 2015.

[80] M. Zaffalon. A credal approach to naive classification. In *1st International Symposium on Imprecise Probabilities and Their Applications*, volume 99, pages 405–414, 1999.

[81] M. Zaffalon. The naive credal classifier. *Journal of statistical planning and inference*, 105(1):5–21, 2002.

Appendix A

Dempster-Shafer Graphical Model Proofs

Proof of Theorem 5.10:

The theorem statement is:

> Consider an arbitrary conditional DS table over x, dependent on $\mathcal{Y}$. Let $F_x = \prod_{V_\mathcal{Y} \in \text{Val}(\mathcal{Y})} F_{x|V_\mathcal{Y}} = \prod_{V_\mathcal{Y} \in \text{Val}(\mathcal{Y})} \textbf{balloon}\left(D_{x|V_\mathcal{Y}} \middle| V_\mathcal{Y}\right)$ be the total DSF associated with x that is generated from the conditional DS table. If F_x is marginalized to eliminate x, then the resultant DSF is simply the vacuous DS model over $\mathcal{Y}$: $\textbf{marg}\left(F_x|\mathcal{Y}\right) = \mathbf{1}_\mathcal{Y}$.

The proof is:

We will prove that for every focal element J of F_x, $J[\mathcal{Y}] = \text{Val}(\mathcal{Y})$ and the weights of the focal elements of F_x do not need to be normalized to sum to 1.

For each $V_\mathcal{Y} \in \text{Val}(\mathcal{Y})$, choose a focal element $J_{V_\mathcal{Y}}$ from $\mathcal{E}(D_{x|V_\mathcal{Y}})$. We will show that $J = \bigcap_{V_\mathcal{Y} \in \mathcal{Y}} \textbf{balloon}\left(J_{V_\mathcal{Y}} \middle| V_\mathcal{Y}\right)$ satisfies $J[\mathcal{Y}] = \text{Val}(\mathcal{Y})$.

For each $V'_\mathcal{Y} \in \text{Val}(\mathcal{Y})$, choose a $V'_x \in \text{Val}(x)$ such that $V'_x \in J_{V'_\mathcal{Y}}$. It is clear that $\langle V'_\mathcal{Y}, V'_x \rangle \in \textbf{balloon}\left(J_{V'_\mathcal{Y}} \middle| V'_\mathcal{Y}\right)$. By definition of the ballooning extension, for all $V''_\mathcal{Y} \in \text{Val}(\mathcal{Y})$ such that $V''_\mathcal{Y} \neq V'_\mathcal{Y}$, $\langle V'_\mathcal{Y}, V'_x \rangle \in \textbf{balloon}\left(J_{V''_\mathcal{Y}} \middle| V''_\mathcal{Y}\right)$ by default. Therefore $\langle V'_\mathcal{Y}, V'_x \rangle \in J$, and

hence $V'_{\mathcal{Y}} \in J[\mathcal{Y}]$. Since $V'_{\mathcal{Y}} \in J[\mathcal{Y}]$ for all $V'_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$, $J[\mathcal{Y}] = \text{Val}(\mathcal{Y})$.

Since $J \neq \emptyset$, no weight is lost by the DSF multiplication. Therefore, $\textbf{marg}\,(F_x|\mathcal{Y}) = \mathbf{1}_{\mathcal{Y}}$.

□

Proof of Theorem 5.11:

The theorem statement is:

> Given an arbitrary DS-BN, no normalization of focal element weights is needed to produce the final DS model after all factors are multiplied together.

The proof is:

For each $x \in \mathcal{X}$ and $V_{\text{Pa}(x)} \in \text{Val}(\text{Pa}(x))$, when $D_{x|V_{\text{Pa}(x)}}$ is converted to DSF $F_{x|V_{\text{Pa}(x)}} = \textbf{balloon}\left(D_{x|V_{\text{Pa}(x)}} \middle| V_{\text{Pa}(x)}\right)$ via the ballooning extension, the weights remain normalized.

For each $x \in \mathcal{X}$ and $V_{\text{Pa}(x)} \in \text{Val}(\text{Pa}(x))$, let $J_{x|V_{\text{Pa}(x)}}$ be an arbitrary focal element of $F_{x|V_{\text{Pa}(x)}}$. Let $J'_{x|V_{\text{Pa}(x)}} = J_{x|V_{\text{Pa}(x)}}[\mathcal{X}]$. We will construct an assignment $V \in \text{Val}(\mathcal{X})$ that belongs to $J'_{x|V_{\text{Pa}(x)}}$ for all $x \in \mathcal{X}$ and $V_{\text{Pa}(x)} \in \text{Val}(\text{Pa}(x))$. This will prove that $\bigcap_{x\in\mathcal{X}}\bigcap_{V_{\text{Pa}(x)}\in\text{Val}(\text{Pa}(x))} J'_{x|V_{\text{Pa}(x)}}$ is non-empty, which will in turn imply that no weight is lost during multiplication.

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ where for each x_i, $\text{Pa}(x_i) \subseteq \{x_1, \ldots, x_{i-1}\}$. For $i = 1, 2, \ldots, N$ in the given order, we will select $V[x_i]$ given $V[\{x_1, \ldots, x_{i-1}\}]$.

For an arbitrary $V' \in \text{Val}(\text{Pa}(x_i))$ where $V' \neq V[\text{Pa}(x_i)]$, $V[\{x_1, x_2, \ldots, x_i\}] \in J'_{x_i|V'}[\{x_1, x_2, \ldots, x_i\}]$ no matter the choice of $V[x_i]$ using the definition of the ballooning extension. For $V' = V[\text{Pa}(x_i)]$, $V[x_i]$ must be chosen from the focal element in $D_{x_i|V'}$ that was extended to $J'_{x_i|V'}$ so that $V[\{x_1, x_2, \ldots, x_i\}] \in J'_{x_i|V'}[\{x_1, x_2, \ldots, x_i\}]$. Repeating this selection process yields a $V \in \text{Val}(\mathcal{X})$ that belongs to all of the $J'_{x|V_{\text{Pa}(x)}}$ for all $x \in \mathcal{X}$ and $V_{\text{Pa}(x)} \in \text{Val}(\text{Pa}(x))$. Therefore no weight is lost during multiplication and no normalization is needed to form the resultant DS model. □

Proof of Theorem 5.12:

The theorem statement is:

For a set of DSFs $\{F_1, F_2, \ldots, F_k\}$; evidence variables $\mathcal{Z}$ and evidence $V_{\mathcal{Z}}$,

$$\mathbf{cond}\left(\prod_{i=1}^{k} F_i \middle| V_{\mathcal{Z}}\right) = \prod_{i=1}^{k} \mathbf{cond}\left(F_i | V_{\mathcal{Z}}\right)$$

The proof is:

Let $\mathcal{Y} = \bigcup_{i=1}^{k} \mathrm{Var}(F_i)$.

To establish this theorem, it is sufficient to observe that for any choice of focal elements $J_i \in \mathcal{E}(F_i)$, that: $\left(\left(\bigcap_{i=1}^{k} J_i\right) \cap \{V_{\mathcal{Z}}\}\right)[\mathcal{Y} \setminus \mathcal{Z}] = \bigcap_{i=1}^{k} \left((J_i \cap \{V_{\mathcal{Z}}\})[\mathcal{Y} \setminus \mathcal{Z}]\right)$

From this observation, the focal elements of $\mathbf{cond}\left(\prod_{i=1}^{k} F_i \middle| V_{\mathcal{Z}}\right)$ and $\prod_{i=1}^{k} \mathbf{cond}\left(F_i | V_{\mathcal{Z}}\right)$ are equivalent and have the same probability masses. □

Proof of Theorem 5.13:

The theorem statement is:

For a variable $x \in \mathcal{X}$; a DSF F_1 for which $x \in \mathrm{Var}(F_1)$; and another DSF F_2 for which $x \notin \mathrm{Var}(F_2)$, it is the case that:

$$\mathbf{marg}\left(F_1 \times F_2 | \mathrm{Var}(F_1 \times F_2) \setminus x\right) = \mathbf{marg}\left(F_1 | \mathrm{Var}(F_1) \setminus x\right) \times F_2$$

The proof is:

Let $\mathcal{Y} = \mathrm{Var}(F_1) \cup \mathrm{Var}(F_2)$. To establish this theorem, it is sufficient to observe that for any choice of focal elements $J_1 \in \mathcal{E}(F_1)$ and $J_2 \in \mathcal{E}(F_2)$, that: $(J_1 \cap J_2)[\mathcal{Y} \setminus x] = J_1[\mathrm{Var}(F_1) \setminus x] \cap J_2$ (Recall that cylindrical extensions are implicitly performed when taking set intersections.).

From this observation, the focal elements of $\mathbf{marg}\left(F_1 \times F_2 | \mathrm{Var}(F_1 \times F_2) \setminus x\right)$ and $\mathbf{marg}\left(F_1 | \mathrm{Var}(F_1) \setminus x\right) \times F_2$ are equivalent and have the same probability masses. □

Appendix B

The NP-hardness of Problem 6.4

The satisfiability problem (SAT) from propositional logic is known to be NP-complete by the Cook-Levin theorem [59, pg. 276].

The formulation of the SAT problem given in [59, pg. 271] is:

Problem B.1. Satifiability (SAT) formulation 1

Input A propositional formula ϕ of length m, with at most n binary propositional variables.

Output A binary yes/no that indicates if there exists an assignment to the n propositional variables such that ϕ evaluates to "true".

Here however, an alternate formulation of the SAT problem is used that is equivalent to the first formulation:

Problem B.2. Satifiability (SAT) formulation 2

Input A set of n binary propositional variables $x_1, x_2, \ldots, x_n \in \{0, 1\}$.

A set of m clauses $\phi_1, \phi_2, \ldots, \phi_m$. Each clause is a disjunction: $\phi_j = l_{j,1} \vee l_{j,2} \vee \cdots \vee l_{j,n}$ where $l_{j,i}$ is either F(false); x_i; or $\neg x_i$.

Output A binary yes/no that indicates if there exists an assignment to the n propositional variables such that every ϕ_j evaluates to "true".

Both formulations of SAT are polynomial time reducible to each other. Formulation 2 can be envisioned as a specific instance of formulation 1 with $\phi = \phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_m$, and so formulation 2 is readily polynomial time reducible to formulation 1. Formulation 1 can be reduced to formulation 2 in polynomial time via the following process: given the expression tree for ϕ, an extra propositional variable can be created for each interior node. A node's dependence on its children can be encoded via a small set of disjunctive clauses. Hence, the condition that ϕ return true can be encoded by a set of disjunctive clauses that can be generated in polynomial time. This set of disjunctive clauses constitutes the polynomial time reduction of formulation 1 to formulation 2.

To establish that problem 6.4 is NP-hard, it is sufficient to show that SAT (formulation 2) is polynomial time reducible to problem 6.4. Polynomial time reducible means that SAT can be solved in polynomial time provided that a polynomial time algorithm exists for problem 6.4. SAT can be solved by problem 6.4 in the following manner:

Start with the input to SAT: A set of n binary propositional variables $x_1, x_2, \ldots, x_n \in \{0, 1\}$, and a set of m clauses $\phi_1, \phi_2, \ldots, \phi_m$.

SAT is solved via problem 6.4 by the following algorithm:

$n' \leftarrow n + m$

$m' \leftarrow 2n$

for $i' = 1$ to n **do**

  for $i = 1$ to n **do**

    if $i = i'$ **then**

      $a_{i',2i-1} \leftarrow 0$ and $a_{i',2i} \leftarrow 0$

    else

      $a_{i',2i-1} \leftarrow 1$ and $a_{i',2i} \leftarrow 1$

    end if

    $b_{i',2i-1} \leftarrow 1$ and $b_{i',2i} \leftarrow 1$

end for

$c_{i'} \leftarrow 2n - 1$

end for

for $j = 1$ to m **do**

for $i = 1$ to n **do**

if $l_{j,i} \equiv x_i$ **then**

$a_{n+j,2i-1} \leftarrow 0$ and $a_{n+j,2i} \leftarrow 1$

else if $l_{j,i} \equiv \neg x_i$ **then**

$a_{n+j,2i-1} \leftarrow 1$ and $a_{n+j,2i} \leftarrow 0$

else

$a_{n+j,2i-1} \leftarrow 1$ and $a_{n+j,2i} \leftarrow 1$

end if

$b_{n+j,2i-1} \leftarrow 1$ and $b_{n+j,2i} \leftarrow 1$

end for

$c_{n+j} \leftarrow 2n - 1$

end for

Solve problem 6.4 with the values of $n = n'$, $m = m'$, $a_{i,j}$, $b_{i,j}$, c_i, and use maximization. Assign the result to r.

if $r \geq n$ **then**

return: yes ($\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_m$ is satisfiable)

else

return: no ($\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_m$ is unsatisfiable)

end if

Figure B.1 depicts the use of problem 6.4 to solve the SAT problem involving the clauses: $\phi_1 = x_1 \vee x_2 \vee x_3$; $\phi_2 = F \vee x_2 \vee \neg x_3$; $\phi_3 = \neg x_1 \vee \neg x_2 \vee F$; and $\phi_4 = \neg x_1 \vee \neg x_2 \vee \neg x_3$. Note that problem 6.4 is optimized by a "corner state", wherein the parameters do not take

on intermediate values. For each of the top n rows, one of x_i or $\neg x_i$ is chosen to be true by forcing the corresponding row entry to 0. The product of the corresponding column is forced to 0. The products of at least n columns are 0, so the sum of products is at most n. For each of the bottom m rows, a supporting literal for clause ϕ_j is chosen by again forcing the corresponding row entry to 0. If the chosen supporting literal does not match the choice of x_i's in the top n rows, then another column has a 0 product and the sum of products falls below n. If the clauses are all simultaneously satisfiable, then there exists a choice of assignments to each x_i and a choice of supporting literal for each ϕ_j so that the product of n columns is 1, and the sum of products attains a maximum of n.

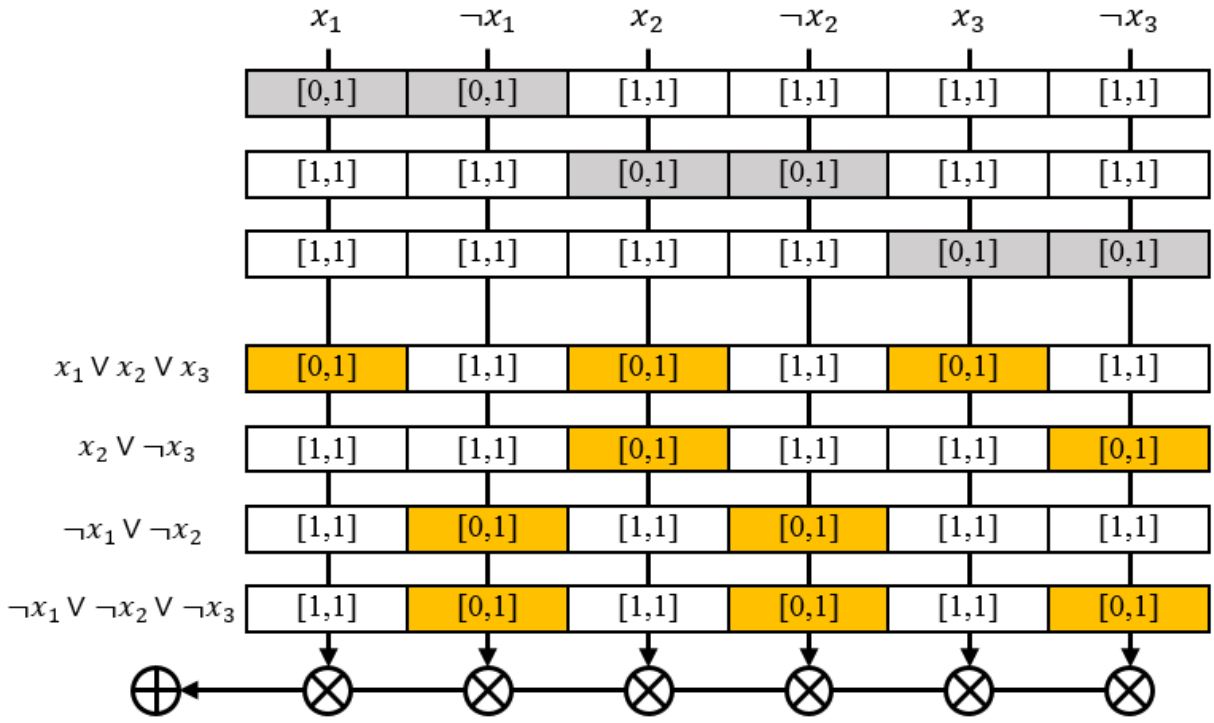


Figure B.1: A visual depiction of setting up problem 6.4 to solve the SAT problem.

Appendix C

Layered Probability Model Proofs

Proof of Theorem 7.3

The theorem statement is:

Given a set $\mathcal{Y}$ of variables, and a set of correlation terms T, then for each $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$, it is the case that:

$$\Pr(V_{\mathcal{Y}} : T) = \sum_{S \in \text{partition}(T:\mathcal{Y})} \frac{1}{\prod_{x \in \mathcal{Y} \setminus \text{Var}(S)} |\text{Val}(x)|} \prod_{t \in S} t(V_{\mathcal{Y}}[\text{Var}(t)])$$

where $\text{partition}(T : \mathcal{Y})$ is the set of all subsets (including the empty set) of T such that for any $S \in \text{partition}(T : \mathcal{Y})$, it is the case that

$$\forall t \in S : \text{Var}(t) \subseteq \mathcal{Y}$$

$$\forall t_1, t_2 \in S : t_1 \neq t_2 \implies \text{Var}(t_1) \cap \text{Var}(t_2) = \emptyset$$

Note that $\emptyset \in \text{partition}(T : \mathcal{Y})$, and that the empty product is 1.

For any $S \subseteq T$, it is defined that $\text{Var}(S) = \bigcup_{t \in S} \text{Var}(t)$ (note that the empty union is $\emptyset$)

The proof is:

Theorem 7.3 will be proven via induction.

Base case: For the base case, assume that there is no $t \in T$ such that $\text{Var}(t) \subseteq \mathcal{Y}$.

By definition, $\Pr(V_{\mathcal{Y}} : T) = \frac{1}{\prod_{x \in \mathcal{Y}} |\text{Val}(x)|}$

By definition, $\text{partition}(T : \mathcal{Y}) = \{\emptyset\}$ ($\text{partition}(T : \mathcal{Y})$ contains only the empty set of correlation terms)

It can be derived that:

$$\sum_{S \in \text{partition}(T:\mathcal{Y})} \frac{1}{\prod_{x \in \mathcal{Y} \setminus \text{Var}(S)} |\text{Val}(x)|} \prod_{t \in S} t(V_{\mathcal{Y}}[\text{Var}(t)])$$
$$= \frac{1}{\prod_{x \in \mathcal{Y} \setminus \emptyset} |\text{Val}(x)|} \prod_{t \in \emptyset} t(V_{\mathcal{Y}}[\text{Var}(t)]) = \frac{1}{\prod_{x \in \mathcal{Y}} |\text{Val}(x)|} = \Pr(V_{\mathcal{Y}} : T)$$

This establishes the base case.

Inductive case: For the inductive case, assume that there is a $t \in T$ such that $\text{Var}(t) \subseteq \mathcal{Y}$, and that the theorem holds for $T \setminus \{t\}$.

More specifically, the following inductive hypotheses will be assumed:

$$\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}} : T \setminus \{t\}) = \sum_{S \in \text{partition}(T \setminus \{t\} : \mathcal{Y})} \frac{1}{\prod_{x \in \mathcal{Y} \setminus \text{Var}(S)} |\text{Val}(x)|} \prod_{t' \in S} t'(V_{\mathcal{Y}}[\text{Var}(t')])$$

(which occurs from the substitutions $\mathcal{Y} \leftarrow \mathcal{Y}$ and $T \leftarrow T \setminus \{t\}$). This equation will be referenced with $(I1)$.

and the hypothesis:

$$\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \text{Var}(t)] : T \setminus \{t\})$$
$$= \sum_{S \in \text{partition}(T \setminus \{t\} : \mathcal{Y} \setminus \text{Var}(t))} \frac{1}{\prod_{x \in (\mathcal{Y} \setminus \text{Var}(t)) \setminus \text{Var}(S)} |\text{Val}(x)|} \prod_{t' \in S} t'(V_{\mathcal{Y}}[\text{Var}(t')])$$

(which occurs from the substitutions $\mathcal{Y} \leftarrow \mathcal{Y} \setminus \text{Var}(t)$ and $T \leftarrow T \setminus \{t\}$). This equation will be referenced with $(I2)$.

By the recursive definition, $\Pr(V_{\mathcal{Y}} : T) = \Pr(V_{\mathcal{Y}} : T \setminus \{t\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \text{Var}(t)] : T \setminus \{t\}) t(V_{\mathcal{Y}}[\text{Var}(t)])$. This equation will be referenced with (R).

It can be derived that: $\text{partition}(T : \mathcal{Y}) = \text{partition}(T \setminus \{t\} : \mathcal{Y}) \cup \{S \cup \{t\} | S \in \text{partition}(T \setminus \{t\} : \mathcal{Y} \setminus \text{Var}(t))\}$. This equation will be referenced with (P).

The following derivation can now occur starting from the expression:

$$\sum_{S \in \text{partition}(T:\mathcal{Y})} \frac{1}{\prod_{x \in \mathcal{Y} \setminus \text{Var}(S)} |\text{Val}(x)|} \prod_{t' \in S} t'(V_{\mathcal{Y}}[\text{Var}(t')])$$

(Using (P) gives:)

$$= \sum_{S \in \text{partition}(T \setminus \{t\}:\mathcal{Y})} \frac{1}{\prod_{x \in \mathcal{Y} \setminus \text{Var}(S)} |\text{Val}(x)|} \prod_{t' \in S} t'(V_{\mathcal{Y}}[\text{Var}(t')])$$
$$+ \sum_{S \in \text{partition}(T \setminus \{t\}:\mathcal{Y} \setminus \text{Var}(t))} \frac{1}{\prod_{x \in \mathcal{Y} \setminus (\text{Var}(S) \cup \text{Var}(t))} |\text{Val}(x)|} t(V_{\mathcal{Y}}[\text{Var}(t)]) \prod_{t' \in S} t'(V_{\mathcal{Y}}[\text{Var}(t')])$$

(Using $(I1)$ gives:)

$$= \Pr(V_{\mathcal{Y}} : T \setminus \{t\})$$
$$+ t(V_{\mathcal{Y}}[\text{Var}(t)]) \sum_{S \in \text{partition}(T \setminus \{t\}:\mathcal{Y} \setminus \text{Var}(t))} \frac{1}{\prod_{x \in (\mathcal{Y} \setminus \text{Var}(t)) \setminus \text{Var}(S)} |\text{Val}(x)|} \prod_{t' \in S} t'(V_{\mathcal{Y}}[\text{Var}(t')])$$

(Using $(I2)$ gives:)

$$= \Pr(V_{\mathcal{Y}} : T \setminus \{t\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \text{Var}(t)] : T \setminus \{t\}) t(V_{\mathcal{Y}}[\text{Var}(t)])$$

(Using (R) gives:)

$$= \Pr(V_{\mathcal{Y}} : T)$$

This establishes the inductive case. □

Proof of Theorem 7.4

The theorem statement is:

Given a set $\mathcal{Y}$ of variables, and a set of correlation terms T, then for any subset of variables $\mathcal{Z} \subset \mathcal{Y}$,

$$\forall V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z}) : \Pr(V_{\mathcal{Z}} : T) = \sum_{V' \in \text{Val}(\mathcal{Y} \setminus \mathcal{Z})} \Pr(\langle V_{\mathcal{Z}}, V' \rangle : T)$$

The proof is:

Theorem 7.4 will be proven via induction.

Base case: For the base case, assume that $T = \emptyset$. It can be easily derived that for any $V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z})$ that:

$$\sum_{V' \in \text{Val}(\mathcal{Y} \setminus \mathcal{Z})} \Pr(\langle V_{\mathcal{Z}}, V' \rangle : \emptyset) = \sum_{V' \in \text{Val}(\mathcal{Y} \setminus \mathcal{Z})} \frac{1}{\prod_{x \in \mathcal{Y}} |\text{Val}(x)|} = \frac{\prod_{x \in \mathcal{Y} \setminus \mathcal{Z}} |\text{Val}(x)|}{\prod_{x \in \mathcal{Y}} |\text{Val}(x)|} = \frac{1}{\prod_{x \in \mathcal{Z}} |\text{Val}(x)|}$$

$$= \Pr(V_{\mathcal{Z}} : \emptyset)$$

This establishes the base case.

Inductive case: For the inductive case, assume that there exists a correlation term t such that $t \in T$. The inductive hypotheses that will be assumed are:

$$\forall V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z}) : \Pr(V_{\mathcal{Z}} : T \setminus \{t\}) = \sum_{V' \in \text{Val}(\mathcal{Y} \setminus \mathcal{Z})} \Pr(\langle V_{\mathcal{Z}}, V' \rangle : T \setminus \{t\})$$

(which occurs from the substitutions $\mathcal{Y} \leftarrow \mathcal{Y}$ and $T \leftarrow T \setminus \{t\}$ and $\mathcal{Z} \leftarrow \mathcal{Z}$). This equation will be referenced with $(I1)$.

$$\forall V_{\mathcal{Z}} \in \text{Val}(\mathcal{Z}) : \Pr(V_{\mathcal{Z}}[\mathcal{Z} \setminus \text{Var}(t)] : T \setminus \{t\}) = \sum_{V' \in \text{Val}((\mathcal{Y} \setminus \mathcal{Z}) \setminus \text{Var}(t))} \Pr(\langle V_{\mathcal{Z}}[\mathcal{Z} \setminus \text{Var}(t)], V' \rangle : T \setminus \{t\})$$

(which occurs from the substitutions $\mathcal{Y} \leftarrow \mathcal{Y} \setminus \text{Var}(t)$ and $T \leftarrow T \setminus \{t\}$ and $\mathcal{Z} \leftarrow \mathcal{Z} \setminus \text{Var}(t)$).

This equation will be referenced with $(I2)$.

There are 3 possible cases:

- $\mathrm{Var}(t) \not\subseteq \mathcal{Y}$
- $\mathrm{Var}(t) \subseteq \mathcal{Y}$ but $\mathrm{Var}(t) \not\subseteq \mathcal{Z}$
- $\mathrm{Var}(t) \subseteq \mathcal{Z}$

When $\mathrm{Var}(t) \not\subseteq \mathcal{Y}$, then t has no effect on $\Pr(\langle V_{\mathcal{Z}}, V' \rangle : T)$ or $\Pr(V_{\mathcal{Z}} : T)$ so the conclusion immediately follows.

When $\mathrm{Var}(t) \subseteq \mathcal{Y}$, then the following derivation occurs for an arbitrary $V_{\mathcal{Z}} \in \mathrm{Val}(\mathcal{Z})$:

$$\sum_{V' \in \mathrm{Val}(\mathcal{Y} \setminus \mathcal{Z})} \Pr(\langle V_{\mathcal{Z}}, V' \rangle : T)$$

(Using the recursive definition gives:)

$$= \sum_{V' \in \mathrm{Val}(\mathcal{Y} \setminus \mathcal{Z})} \Bigg(\Pr(\langle V_{\mathcal{Z}}, V' \rangle : T \setminus \{t\}) + \Pr(\langle V_{\mathcal{Z}}[\mathcal{Z} \setminus \mathrm{Var}(t)], V'[(\mathcal{Y} \setminus \mathcal{Z}) \setminus \mathrm{Var}(t)] \rangle : T \setminus \{t\}) \cdot t(\langle V_{\mathcal{Z}}[\mathcal{Z} \cap \mathrm{Var}(t)], V'[(\mathcal{Y} \setminus \mathcal{Z}) \cap \mathrm{Var}(t)] \rangle) \Bigg)$$

(Using $(I1)$ gives:)

$$= \Pr(V_{\mathcal{Z}} : T \setminus \{t\}) + \left(\sum_{V' \in \mathrm{Val}((\mathcal{Y} \setminus \mathcal{Z}) \setminus \mathrm{Var}(t))} \Pr(\langle V_{\mathcal{Z}}[\mathcal{Z} \setminus \mathrm{Var}(t)], V' \rangle : T \setminus \{t\}) \right) \cdot \left(\sum_{V' \in \mathrm{Val}((\mathcal{Y} \setminus \mathcal{Z}) \cap \mathrm{Var}(t))} t(\langle V_{\mathcal{Z}}[\mathcal{Z} \cap \mathrm{Var}(t)], V' \rangle) \right)$$

(Using $(I2)$ and the fact that a correlation term summed over any variable is 0 gives:)

$$= \Pr(V_{\mathcal{Z}} : T \setminus \{t\}) + \Pr(V_{\mathcal{Z}}[\mathcal{Z} \setminus \mathrm{Var}(t)] : T \setminus \{t\}) \cdot \begin{cases} 0 & ((\mathcal{Y} \setminus \mathcal{Z}) \cap \mathrm{Var}(t) \neq \emptyset) \\ t(V_{\mathcal{Z}}[\mathcal{Z} \cap \mathrm{Var}(t)]) & ((\mathcal{Y} \setminus \mathcal{Z}) \cap \mathrm{Var}(t) = \emptyset) \end{cases}$$

$$= \begin{cases} \Pr(V_{\mathcal{Z}} : T \setminus \{t\}) & (\mathrm{Var}(t) \not\subseteq \mathcal{Z}) \\ \Pr(V_{\mathcal{Z}} : T \setminus \{t\}) + \Pr(V_{\mathcal{Z}}[\mathcal{Z} \setminus \mathrm{Var}(t)] : T \setminus \{t\}) t(V_{\mathcal{Z}}[\mathrm{Var}(t)]) & (\mathrm{Var}(t) \subseteq \mathcal{Z}) \end{cases}$$

(Using the recursive definition gives:)

$$= \Pr(V_{\mathcal{Z}} : T)$$

This establishes the inductive case. □

Proof of Theorem 7.5

The theorem statement is:

Let $\mathcal{Y}$ be an arbitrary set of variables, and let T be an arbitrary set of correlation terms. If there exists two correlation terms $t_1, t_2 \in T$ such that $\mathrm{Var}(t_1) = \mathrm{Var}(t_2)$, then removing t_1 and t_2 and adding $t_1 + t_2$ to T to get $T' = (T \setminus \{t_1, t_2\}) \cup \{t_1 + t_2\}$ will not change $\Pr(V_{\mathcal{Y}} : T)$ for all assignments $V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y})$:

$$\forall V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}} : T) = \Pr(V_{\mathcal{Y}} : T')$$

The proof is:

$\mathrm{Var}(t_1) = \mathrm{Var}(t_2)$ by assumption, so let this common set be denoted by $\mathcal{Z}$. Let $V_{\mathcal{Y}} \in \mathrm{Val}(\mathcal{Y})$ be an arbitrary assignment to the variables from $\mathcal{Y}$.

If $\mathcal{Z} \not\subseteq \mathcal{Y}$, then neither t_1 nor t_2 will have any impact on the probability $\Pr(V_{\mathcal{Y}} : T)$, and neither will the presence of $t_1 + t_2$ have any impact on the probability $\Pr(V_{\mathcal{Y}} : T')$. This means that $\Pr(V_{\mathcal{Y}} : T) = \Pr(V_{\mathcal{Y}} : T')$.

Now assume that $\mathcal{Z} \subseteq \mathcal{Y}$.

By definition for each $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$,

$$\begin{aligned}
\Pr(V_{\mathcal{Y}} : T) =& \Pr(V_{\mathcal{Y}} : T \setminus \{t_1\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \mathcal{Z}] : T \setminus \{t_1\}) t_1(V_{\mathcal{Y}}[\mathcal{Z}]) \\
=& (\Pr(V_{\mathcal{Y}} : T \setminus \{t_1, t_2\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \mathcal{Z}] : T \setminus \{t_1, t_2\}) t_2(V_{\mathcal{Y}}[\mathcal{Z}])) \\
&+ \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \mathcal{Z}] : T \setminus \{t_1, t_2\}) t_1(V_{\mathcal{Y}}[\mathcal{Z}]) \\
=& \Pr(V_{\mathcal{Y}} : T \setminus \{t_1, t_2\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \mathcal{Z}] : T \setminus \{t_1, t_2\})(t_1(V_{\mathcal{Y}}[\mathcal{Z}]) + t_2(V_{\mathcal{Y}}[\mathcal{Z}])) \\
=& \Pr(V_{\mathcal{Y}} : (T \setminus \{t_1, t_2\}) \cup \{t_1 + t_2\}) \\
=& \Pr(V_{\mathcal{Y}} : T')
\end{aligned}$$

□

Proof of Theorem 7.7

The theorem statement is:

> Let $G(\mathcal{Y} : T)$ denote an arbitrary correlation graph over the set of variables $\mathcal{Y}$ with the set of correlation terms T. If the variable set $\mathcal{Y}$ can be partitioned into disjoint subsets $\mathcal{Y}_1, \mathcal{Y}_2, \ldots \mathcal{Y}_n$ where $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \cdots \cup \mathcal{Y}_n$, and for any $i \neq j$ there are no paths that link $\mathcal{Y}_i$ to $\mathcal{Y}_j$, then $\mathcal{Y}_1 \perp \mathcal{Y}_2 \perp \cdots \perp \mathcal{Y}_n$.

The proof is:

Theorem 7.7 will be proven via induction.

Base case: For the base case, assume that $T = \emptyset$.

For an arbitrary $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$,

$$\Pr(V_{\mathcal{Y}} : \emptyset) = \frac{1}{\prod_{x \in \mathcal{Y}} |\text{Val}(x)|} = \frac{1}{\prod_{i=1}^{n} \prod_{x \in \mathcal{Y}_i} |\text{Val}(x)|} = \prod_{i=1}^{n} \frac{1}{\prod_{x \in \mathcal{Y}_i} |\text{Val}(x)|} = \prod_{i=1}^{n} \Pr(V_{\mathcal{Y}}[\mathcal{Y}_i] : \emptyset)$$

This establishes the base case.

Inductive case: For the inductive case, assume that there exists a correlation term t such that $t \in T$. Since there are no paths connecting $\mathcal{Y}_i$ to $\mathcal{Y}_j$ for any $i \neq j$, there must exist $\mathcal{Y}_i$ such that $\text{Var}(t) \subseteq \mathcal{Y}_i$. The inductive hypotheses that will be assumed are:

$$\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}} : T \setminus \{t\}) = \prod_{j=1}^{n} \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T \setminus \{t\})$$

(which occurs from the substitutions $\mathcal{Y}_j \leftarrow \mathcal{Y}_j$ and $T \leftarrow T \setminus \{t\}$). This equation will be referenced with $(I1)$.

$$\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \text{Var}(t)] : T \setminus \{t\}) = \prod_{j=1}^{n} \begin{cases} \Pr(V_{\mathcal{Y}}[\mathcal{Y}_i \setminus \text{Var}(t)] : T \setminus \{t\}) & (j = i) \\ \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T \setminus \{t\}) & (j \neq i) \end{cases}$$

(which occurs from the substitutions $\mathcal{Y}_i \leftarrow \mathcal{Y}_i \setminus \text{Var}(t)$ and $\mathcal{Y}_j \leftarrow \mathcal{Y}_j$ for all $(j \neq i)$ and $T \leftarrow T \setminus \{t\}$). This equation will be referenced with $(I2)$.

For an arbitrary $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$, the following derivation can occur:

(Using the recursive definition gives:)

$$\Pr(V_{\mathcal{Y}} : T) = \Pr(V_{\mathcal{Y}} : T \setminus \{t\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y} \setminus \text{Var}(t)] : T \setminus \{t\}) t(V_{\mathcal{Y}}[\text{Var}(t)])$$

(Using $(I1)$ and $(I2)$ gives:)

$$= \prod_{j=1}^{n} \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T \setminus \{t\}) + \prod_{j=1}^{n} \left\{ \begin{array}{cc} \Pr(V_{\mathcal{Y}}[\mathcal{Y}_i \setminus \text{Var}(t)] : T \setminus \{t\}) & (j = i) \\ \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T \setminus \{t\}) & (j \neq i) \end{array} \right\} t(V_{\mathcal{Y}}[\text{Var}(t)])$$

(factoring out common terms gives:)

$$= \prod_{j=1}^{n} \left\{ \begin{array}{ll} 1 & (j = i) \\ \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T \setminus \{t\}) & (j \neq i) \end{array} \right\}$$

$$\cdot (\Pr(V_{\mathcal{Y}}[\mathcal{Y}_i] : T \setminus \{t\}) + \Pr(V_{\mathcal{Y}}[\mathcal{Y}_i \setminus \text{Var}(t)] : T \setminus \{t\}) t(V_{\mathcal{Y}}[\text{Var}(t)]))$$

(Using the recursive definition gives:)

$$= \prod_{j=1}^{n} \left\{ \begin{array}{ll} 1 & (j = i) \\ \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T \setminus \{t\}) & (j \neq i) \end{array} \right\} \cdot \Pr(V_{\mathcal{Y}}[\mathcal{Y}_i] : T)$$

(Since $\text{Var}(t) \not\subseteq \mathcal{Y}_j$ for $j \neq i$ gives:)

$$= \prod_{j=1}^{n} \left\{ \begin{array}{ll} 1 & (j = i) \\ \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T) & (j \neq i) \end{array} \right\} \cdot \Pr(V_{\mathcal{Y}}[\mathcal{Y}_i] : T)$$

$$= \prod_{j=1}^{n} \Pr(V_{\mathcal{Y}}[\mathcal{Y}_j] : T)$$

This establishes the inductive case. □

Proof of Theorem 7.8

The theorem statement is:

> **Correlation term generation algorithm 1** generates valid correlation terms:
>
> $$\forall t \in T : \forall x \in \text{Var}(t) : \forall V \in \text{Val}(\text{Var}(t) \setminus x) : \sum_{v_x \in \text{Val}(x)} t(\langle v_x, V \rangle) = 0$$

The proof is:

For each $n = 1, 2, ..., |\mathcal{X}|$, T_n will denote the set of correlation terms T at the **start** of iteration n.

Recall that ϵ denotes the empty assignment, as explained in the notation chapter.

This proof will first establish via induction that at the **start** of each iteration $n = 1, 2, ..., |\mathcal{X}|$ of the outer loop, that the remainder $\Pr(V_{\mathcal{X}}) - \Pr(V_{\mathcal{X}} : T_n)$ where $V_{\mathcal{X}} \in \text{Val}(\mathcal{X})$ satisfies the following properties: For each subset $\mathcal{Y} \subseteq \mathcal{X}$ where $|\mathcal{Y}| < n$, it is the case that $\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : \sum_{V \in \text{Val}(\mathcal{X} \setminus \mathcal{Y})}(\Pr(\langle V_{\mathcal{Y}}, V \rangle) - \Pr(\langle V_{\mathcal{Y}}, V \rangle : T_n)) = 0$.

Base case: The base case occurs at the **start** of iteration $n = 1$. The initial probability distribution is $\Pr(V_{\mathcal{X}} : \emptyset) = \frac{1}{\prod_{x \in \mathcal{X}} |\text{Val}(x)|}$ where $V_{\mathcal{X}} \in \text{Val}(\mathcal{X})$. It is a trivial matter to show that $\sum_{V_{\mathcal{X}} \in \text{Val}(\mathcal{X})}(\Pr(V_{\mathcal{X}}) - \Pr(V_{\mathcal{X}} : \emptyset)) = 1 - 1 = 0$.

This establishes the base case.

Inductive case: The inductive case occurs at the **start** of an iteration where $n > 1$.

Let $\mathcal{Y}$ denote an arbitrary set of variables where $|\mathcal{Y}| = n-1$. Let $t_{\mathcal{Y}}$ denote the correlation term computed for $\mathcal{Y}$: $\text{Var}(t_{\mathcal{Y}}) = \mathcal{Y}$. The following two important observations must be made:

- For all correlation terms corresponding to sets of size $n - 1$, only $t_{\mathcal{Y}}$ has any influence on the marginal probability distribution $\Pr(V_{\mathcal{Y}} : T_n)$ where $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$. This means that $\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : \Pr(V_{\mathcal{Y}} : T_n) = \Pr(V_{\mathcal{Y}} : T_{n-1} \cup \{t_{\mathcal{Y}}\})$

- Only correlation terms from T_{n-1} have any impact on the marginal probability distribution $\Pr(V_{\mathcal{Y}} : T)$ where $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$ at the time when $t_{\mathcal{Y}}$ is being computed. This means that $\forall V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y}) : t_{\mathcal{Y}}(V_{\mathcal{Y}}) = \Pr(V_{\mathcal{Y}}) - \Pr(V_{\mathcal{Y}} : T_{n-1})$.

For an arbitrary assignment $V_{\mathcal{Y}} \in \text{Val}(\mathcal{Y})$,

$$\sum_{V \in \text{Val}(\mathcal{X} \setminus \mathcal{Y})} (\Pr(\langle V_{\mathcal{Y}}, V \rangle) - \Pr(\langle V_{\mathcal{Y}}, V \rangle : T_n)) = \Pr(V_{\mathcal{Y}}) - \Pr(V_{\mathcal{Y}} : T_n)$$

$$= \Pr(V_{\mathcal{Y}}) - \Pr(V_{\mathcal{Y}} : T_{n-1} \cup \{t_{\mathcal{Y}}\}) = \Pr(V_{\mathcal{Y}}) - (\Pr(V_{\mathcal{Y}} : T_{n-1}) + \Pr(\epsilon : T_{n-1}) t_{\mathcal{Y}}(V_{\mathcal{Y}}))$$

$$= (\Pr(V_{\mathcal{Y}}) - \Pr(V_{\mathcal{Y}} : T_{n-1})) - 1 \cdot t_{\mathcal{Y}}(V_{\mathcal{Y}}) = t_{\mathcal{Y}}(V_{\mathcal{Y}}) - t_{\mathcal{Y}}(V_{\mathcal{Y}}) = 0$$

This establishes the inductive case.

Lastly, to prove that every correlation term is valid, let $t \in T$ be arbitrary, and let $x \in \text{Var}(t)$ and $V \in \text{Val}(\text{Var}(t) \setminus x)$ both be arbitrary.

$$\sum_{v_x \in \text{Val}(x)} t(\langle v_x, V \rangle) = \sum_{v_x \in \text{Val}(x)} (\Pr(\langle v_x, V \rangle) - \Pr(\langle v_x, V \rangle : T_{|\text{Var}(t)|}))$$
$$= \sum_{V' \in \text{Val}(\mathcal{X} \setminus (\text{Var}(t) \setminus x))} (\Pr(\langle V, V' \rangle) - \Pr(\langle V, V' \rangle : T_{|\text{Var}(t)|}))$$

Letting $n = |\text{Var}(t)|$ and $\mathcal{Y} = \text{Var}(t) \setminus x$, the above expression becomes

$$\sum_{V' \in \text{Val}(\mathcal{X} \setminus \mathcal{Y})} (\Pr(\langle V, V' \rangle) - \Pr(\langle V, V' \rangle : T_n)) = 0$$

using the previously proven statement.

□

Appendix D

Uncertainty Metrics Software Tool Details

A software package entitled "Causal Networks and Uncertainty Metrics" [4] (which can be found on GitHub at `https://github.com/sceastwo/Causal_Networks_and_Uncertainty_Metrics.git`), is used to enable many of the calculations related to uncertainty inference from chapter 5.

This appendix details the architecture of the program "Causal Networks and Uncertainty Metrics".

D.1 Software purpose

The program "Causal Networks and Uncertainty Metrics" [4] is intended to assist with computations related to the "unified inference engine" from chapter 5. The program reads a file containing input data in the form of tree/decision diagram structured conditional uncertainty tables (CUTs), as well as a list of instructions. While executing the input instructions, output instructions can print computed values and structures to both the screen and to an output file. This appendix will detail the various data types; the syntax of the input files; and the instructions that are available.

D.2 Non-recursive data vs recursive data

An important distinction that exists with data is whether it is non-recursive or recursive.

- **Non-recursive data**: Envisioning the data as a directed graph, non-recursive data takes the form of a directed tree. All edges flow from the single root towards the leafs. Non-recursive data structures are the core of the program's architecture. Note that non-recursive data instances can be nested in other instances of the same types, but an instance cannot contain a chain of references back to itself, nor can two chains of references converge.

- **Recursive data**: Envisioning the data as a directed graph, recursive data takes the form of an arbitrary directed graph, and can have loops both directed and undirected. Recursive data structures are the next layer of the program's architecture. Note that recursive data structures can reference themselves in any manner.

D.3 Non-recursive data

D.3.1 simple data

The class simple data is a **non-recursive** unified data type that can store any data that does not have a recursive structure. The class stores a type identifier describing the **type** of data that is being referenced, and an anonymous pointer to the data itself.

The following fields are present (the **bold** text denotes the data type):

type_identifier `the_type_id` stores an identifier for the **type** of data that is being referenced.

int `the_type_size` stores the size of the data block that is being referenced.

void* `the_element` is an anonymous pointer that references the data.

The various data structure **type**s that can be stored as `simple_data` are:

void : (`the_type_id` = `TI_VOID`) A null data structure.

An array of bytes : (`the_type_id` = `TI_MEM_BLOCK`) A generic array of bytes of length `the_type_size`.

bool : (`the_type_id` = `TI_BOOL`) A boolean value.

uchar : (`the_type_id` = `TI_UCHAR`) An unsigned byte.

uint16 : (`the_type_id` = `TI_UINT16`) An unsigned 16-bit integer.

int : (`the_type_id` = `TI_INT`) A signed 32-bit integer.

double : (`the_type_id` = `TI_DOUBLE`) A long floating point number.

doublex : (`the_type_id` = `TI_DOUBLEX`) A long floating point number that can attain special values such as $+\infty$, $-\infty$, and NaN.

fuzzy : (`the_type_id` = `TI_FUZZY`) A triangular fuzzy number.

string : (`the_type_id` = `TI_STRING`) A string of characters.

box : (`the_type_id` = `TI_BOX`) A container class that contains a single simple_data structure.

data_pair : (`the_type_id` = `TI_PAIR`) A pair of simple_data structures.

list : (`the_type_id` = `TI_LIST`) A list of simple_data structures.

D.3.2 boxes

The "box" is a simple container class with a single field that stores a single simple_data instance.

D.3.3 pairs

Often, simple data is needed to be bundled into pairs or more complex data structures to allow for collections of data to be passed from variable to variable. These structures are referred to as **data pair**s and are **non-recursive**.

D.3.4 lists

Similar to endless arrays, a common **non-recursive** data structure that will be encountered are arrays of simple_data that can dynamically extend themselves as more entries are required. These dynamically adjustable arrays are referred to as **list**s.

D.3.5 simple data ordering

Given any two simple data values d_1 and d_2, an ordering is established such that it is always the case that exactly one of the following is true: $d_1 < d_2$; $d_1 = d_2$; or $d_1 > d_2$. Simple data types are lumped together in the ordering, and the ordering of the simple data types is the same as the order in which the types are listed above.

D.4 Recursive data

D.4.1 nodes

A node is a node in a directed graph that are used to implement **recursive** data structures. Complex data structures such as decision trees/diagrams are manifested via a directed graph that is comprised of nodes.

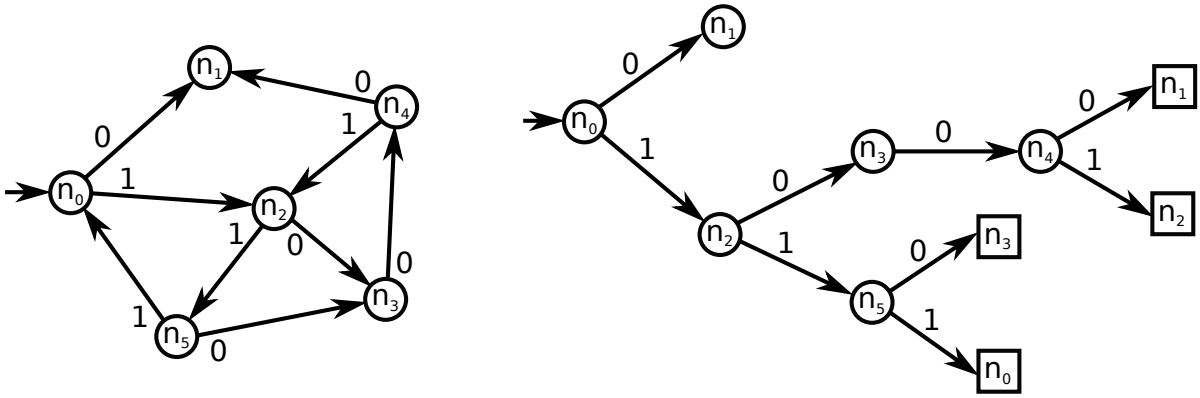
A node consists of:

- A simple data field.
- An ordered list of pointers to the child nodes. The list is indexed starting from 0.

Any directed graph G will be assigned a root node $n_0 = \mathbf{root}(G)$. It will be required that every node and edge from G be reachable from n_0 by a path that traverses edges only in their preferred direction. A directed graph will be identified by its root node n_0, and it will be possible for two directed graphs to share nodes. With respect to graphs sharing nodes, any arbitrary node n from G represents the "subgraph" of G that is rooted at n and consists of all nodes and edges from G that reachable from n be traversing a sequence of edges in their preferred direction. Graphs will be more precisely referred to as **subgraphs**, since each graph will be a subgraph in a global network of nodes. An arbitrary subgraph will be denoted via its root node.

D.4.2 sub-graph tracing

Many subroutines that process a directed graph require that the subgraph that is rooted at the input node be traced in a depth-first fashion.



The tracing in this example proceeds as follows:

- Start at the root node n_0.
- Transition to child 0 of node n_0, which is n_1.
- Node n_1 has no children, so backtrack to node n_0.
- Transition to child 1 of n_0 which is n_2.

- Transition to child 0 of n_2 which is n_3.
- Transition to child 0 of n_3 which is n_4.
- Child 0 of n_4 (which is n_1) has already been visited, so remain at n_4.
- Child 1 of n_4 (which is n_2) has already been visited, so remain at n_4.
- The children of n_4 have been accounted for, so backtrack to n_3.
- The children of n_3 have been accounted for, so backtrack to n_2.
- Transition to child 1 of n_2 which is n_5.
- Child 0 of n_5 (which is n_3) has already been visited, so remain at n_5.
- Child 1 of n_5 (which is n_0) has already been visited, so remain at n_5.
- The children of n_5 have been accounted for, so backtrack to n_2.
- The children of n_2 have been accounted for, so backtrack to n_0.
- The children of n_0 have been accounted for, so the trace finishes.

D.4.3 sub-graph equivalence

Given a subgraphs G_1 and G_2, the concept of "equivalence" between G_1 and G_2 will be more general than simply requiring that G_1 and G_2 have the same root node and hence the same set of nodes and edges. Let $T_1 = \textbf{full\_expand}(G_1)$ denote the possibly infinite tree that is formed by "unrolling" G_1 by searching through G_1 without any concern to whether nodes are being revisited. Let $T_2 = \textbf{full\_expand}(G_2)$ denote the possibly infinite tree that is formed by "unrolling" G_2 by searching through G_2 without any concern to whether nodes are being revisited. G_1 and G_2 are equivalent if and only if T_1 and T_2 have the exact same structure and values assigned to corresponding nodes.

D.4.4 product subgraphs

Binary operators that act on subgraphs G_1 and G_2 first require that a "product subgraph" $G_1 \times G_2$ be formed. If trees $T_1 = \textbf{full\_expand}(G_1)$ and $T_2 = \textbf{full\_expand}$ have the same structure, then the product $T_1 \times T_2$ is a tree with the same structure as T_1 and T_2. The value assigned to each node from $T_1 \times T_2$ is a **data_pair** where the first field contains the value assigned to the corresponding node from T_1 and the second field contains the value assigned to the corresponding node from T_2. It is then required that the product subgraph $G_1 \times G_2$ satisfy $\textbf{full\_expand}(G_1 \times G_2) = T_1 \times T_2$.

If T_1 does not have the same structure as T_2, then the structure of $T_1 \times T_2$ can be the union or the intersection of the structures of T_1 and T_2. If the union of the structures of T_1 and T_2 is chosen, then for a node n that exists for T_1 but not for T_2, the **data_pair** assigned to n in $T_1 \times T_2$ will use **void** as the value from T_2, and vice versa. Again it is then required that the product subgraph $G_1 \times G_2$ satisfy $\textbf{full\_expand}(G_1 \times G_2) = T_1 \times T_2$.

An algorithm for generating the product subgraph proceeds as follows:

- For any node n; let $\textbf{data}(n)$ denote the simple data assigned to n; let $\textbf{degree}(n)$ denote the number of children (the outwards degree of n); and for any $i \in \{0, 1, ..., \textbf{degree}(n)-1\}$, let $n[i]$ denote child i.

- Let $\{n_{1,1}, n_{1,2}, ..., n_{1,k_1}\}$ denote the set of nodes from G_1, and let $\{n_{2,1}, n_{2,2}, ..., n_{2,k_2}\}$ denote the set of nodes from G_2. $n_{1,1}$ is the root node of G_1, and $n_{2,1}$ is the root node of G_2.

- Let $u \in \{0, 1\}$ denote a binary flag that is 1 if the union of the structures of G_1 and G_2 is being created, and is 0 if the intersection of the structures of G_1 and G_2 is being created.

Create a "sink" node $n_{1,0}$ for G_1 and a "sink" node $n_{2,0}$. For each $i \in \{0, 1, 2, ..., k_1\}$ and $j \in \{0, 1, 2, ..., k_2\}$, create a node $\langle n_{1,i}, n_{2,j} \rangle$. These nodes have the following properties:

- $\mathbf{data}(\langle n_{1,0}, n_{2,0}\rangle) = \langle \mathbf{void}, \mathbf{void}\rangle$ and $\mathbf{degree}(\langle n_{1,0}, n_{2,0}\rangle) = 0$

- For each $i \in \{0, 1, 2, ..., k_1\}$, $\mathbf{data}(\langle n_{1,i}, n_{2,0}\rangle) = \langle \mathbf{data}(n_{1,i}), \mathbf{void}\rangle$;

  $\mathbf{degree}(\langle n_{1,i}, n_{2,0}\rangle) = u \cdot \mathbf{degree}(n_{1,i})$;

  and for each $c \in \{0, 1, ..., \mathbf{degree}(\langle n_{1,i}, n_{2,0}\rangle) - 1\}$, $\langle n_{1,i}, n_{2,0}\rangle[c] = \langle n_{1,i}[c], n_{2,0}\rangle$.

- For each $j \in \{0, 1, 2, ..., k_2\}$, $\mathbf{data}(\langle n_{1,0}, n_{2,j}\rangle) = \langle \mathbf{void}, \mathbf{data}(n_{2,j})\rangle$;

  $\mathbf{degree}(\langle n_{1,0}, n_{2,j}\rangle) = u \cdot \mathbf{degree}(n_{2,j})$;

  and for each $c \in \{0, 1, ..., \mathbf{degree}(\langle n_{1,0}, n_{2,j}\rangle) - 1\}$, $\langle n_{1,0}, n_{2,j}\rangle[c] = \langle n_{1,0}, n_{2,j}[c]\rangle$.

- For each $i \in \{0, 1, 2, ..., k_1\}$ and $j \in \{0, 1, 2, ..., k_2\}$,

  $\mathbf{data}(\langle n_{1,i}, n_{2,j}\rangle) = \langle \mathbf{data}(n_{1,i}), \mathbf{data}(n_{2,j})\rangle$;

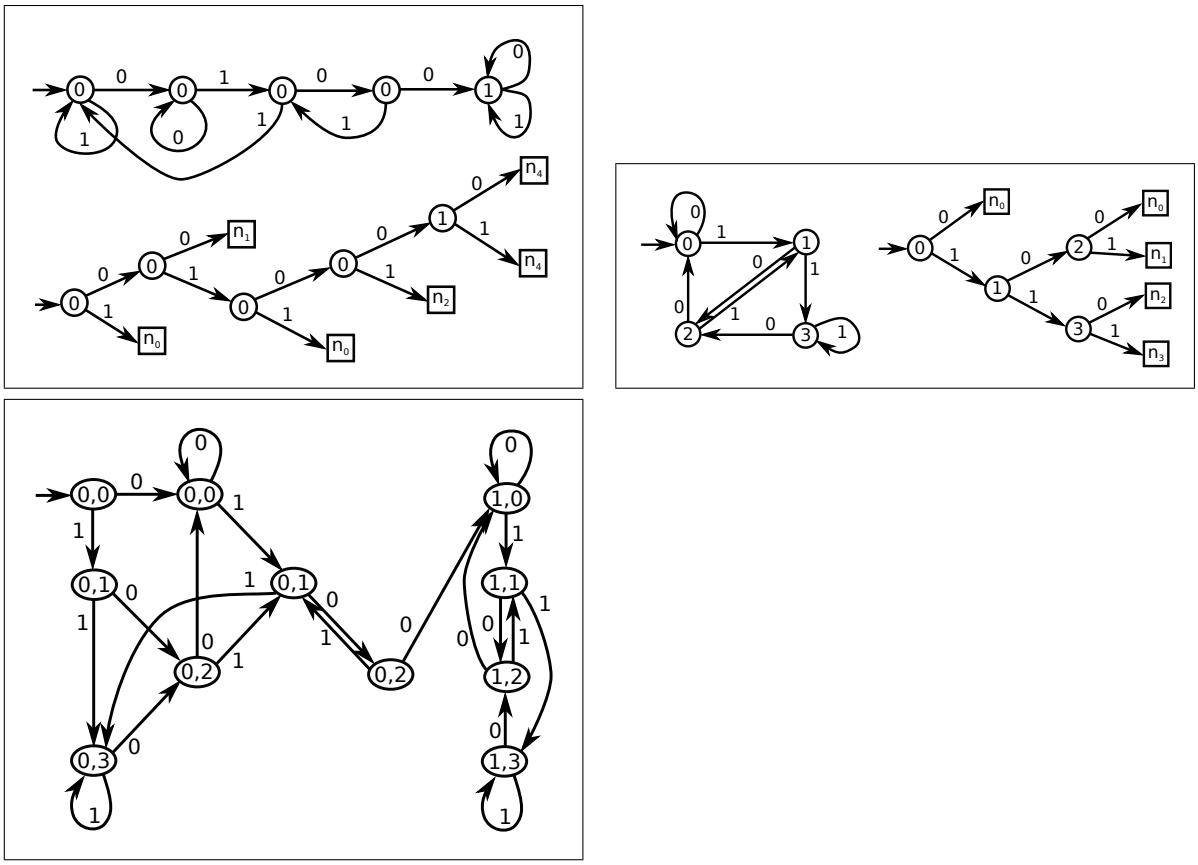
  $$\mathbf{degree}(\langle n_{1,i}, n_{2,j}\rangle) = \begin{cases} \max(\mathbf{degree}(n_{1,i}), \mathbf{degree}(n_{2,j})) & (u = 1) \\ \min(\mathbf{degree}(n_{1,i}), \mathbf{degree}(n_{2,j})) & (u = 0) \end{cases};$$

  and for each $c \in \{0, 1, ..., \mathbf{degree}(\langle n_{1,i}, n_{2,j}\rangle) - 1\}$,

  $$\langle n_{1,i}, n_{2,j}\rangle[c] = \left\langle \begin{cases} n_{1,i}[c] & (c < \mathbf{degree}(n_{1,i})) \\ n_{1,0} & (c \geq \mathbf{degree}(n_{1,i})) \end{cases}, \begin{cases} n_{2,j}[c] & (c < \mathbf{degree}(n_{2,j})) \\ n_{2,0} & (c \geq \mathbf{degree}(n_{2,j})) \end{cases} \right\rangle.$$

$G_1 \times G_2$ is the subgraph rooted at $\langle n_{1,1}, n_{2,1}\rangle$. All nodes $\langle n_{1,i}, n_{2,j}\rangle$ that are not part of $G_1 \times G_2$ are deleted.

Below is an example that shows the product subgraph of two subgraphs:



Top left: A directed graph and its associated depth-first tracer tree that denotes a finite state machine that returns 1 if and only if the sequence 0100 is detected.

Top right: A directed graph and its associated depth-first tracer tree that denotes a finite state machine that returns the decimal number equivalent to the final two bits of the input sequence.

Bottom left: The product directed graph of the two previous directed graphs.

D.5 File parsing and tokens

D.5.1 tokens

All input files are parsed into tokens. The rules for extracting the tokens are described in detail below:

To aid in the description, the set of all characters is given the following taxonomy:

Alphabetic

All letters `aAbB...zZ`

Underscore `_`

Numeric `0123456789`

Symbols

sign `+-`

decimal point `.`

double quote `"`

escape character `\`

other characters

Whitespace

To aid in describing the tokens themselves, notation related to regular expressions will be used:

- $\emptyset$ denotes the empty string.
- `literal` denotes the single string "literal".
- $\text{string}_1\text{string}_2$ denotes the concatenation of string_1 and string_2.
- ($\text{option}_1 \mid \text{option}_2 \mid ... \mid \text{option}_n$) denotes a choice between options option_1, option_2, ..., option_n.
- (fragment)* denotes "fragment" repeated 0 or more times. The parentheses may be omitted if not necessary.

- (fragment)$^+$ denotes "fragment" repeated 1 or more times. The parentheses may be omitted if not necessary.
- (options$_2$ − options$_1$) denotes options$_2$ with all options from "options$_1$" excluded.
- **A** denotes an alphabetic character (including the underscore).
- **N** denotes an arbitrary numeric digit.
- **S** denotes an arbitrary symbol.

The tokens themselves belong to one of the following categories:

Alpha-numeric An alpha-numeric token starts with an alphabetic character (letters + underscore) and consists of any sequence of alphabetic and numeric characters. An alpha-numeric token ends on the last alphabetic or numeric character before a non alphabetic and non numeric character. The general form is

$$\mathbf{A}(\mathbf{A} \mid \mathbf{N})^*$$

Integer An integer token is continuous string of numeric characters where the first character may be `+` or `-`. At least one numeric character is required. The plus sign `+` or minus sign `-` on its own is a symbol. The general form is

$$(\emptyset \mid \boxed{+} \mid \boxed{-})\mathbf{N}^+$$

Fixed-point A fixed point number. The general form that is read is

$$(\emptyset \mid \boxed{+} \mid \boxed{-})\mathbf{N}^*\boxed{.}\mathbf{N}^*$$

After processing, 0s pad the decimal point if necessary giving

$$(\emptyset \mid \boxed{+} \mid \boxed{-})(\mathbf{N}^+ \mid \boxed{0})\boxed{.}(\mathbf{N}^+ \mid \boxed{0})$$

Floating-point A floating point number. The general form that is read is

$$(\emptyset \mid \boxed{+} \mid \boxed{-})\mathbf{N}^*\boxed{.}\mathbf{N}^*\boxed{\mathtt{e}}(\emptyset \mid \boxed{+} \mid \boxed{-})\mathbf{N}^*$$

After processing, 0s pad the decimal point and exponent if necessary giving

$$(\emptyset \mid \boxed{+} \mid \boxed{-})(\mathbf{N}^+ \mid \boxed{0})\boxed{.}(\mathbf{N}^+ \mid \boxed{0})\boxed{\mathtt{e}}(\emptyset \mid \boxed{+} \mid \boxed{-})(\mathbf{N}^+ \mid \boxed{0})$$

String A string of characters. The general form that is read is

$$\boxed{"}(((\mathbf{A} \mid \mathbf{N} \mid \mathbf{S}) - (\boxed{"} \mid \boxed{\backslash} \mid \boxed{\{} \mid \boxed{\}})) \mid (\boxed{\backslash}(\mathbf{A} \mid \mathbf{N} \mid \mathbf{S})))^*\boxed{"}$$

After the encapsulating double quotes have been removed and the escape characters have been resolved, the general form is

$$(\mathbf{A} \mid \mathbf{N} \mid \mathbf{S})^*$$

Symbol An isolated symbol:

$$\mathbf{S} - (\boxed{.} \mid \boxed{"} \mid \boxed{\{} \mid \boxed{\}})$$

In addition, a comment is inert text that is skipped by the token reader. A comment is enclosed by curly braces {}.

D.5.2 simple data syntax

For data input and output, it is necessary to represent simple data as a list of tokens that can be read or written by both a human and machine. This section will describe the text syntax that is used for various types using the same syntax for regular expressions that was described above. When whitespace is present, the amount of type of whitespace is irrelevant. The underlined text denotes fields that filled with relevant data.

void : `0`

An array of bytes : No text representation.

bool : `bool` (`0` | `1`)

uchar : `c` the character

uint16 : `uint16` the integer

int : `i` the integer

double : `d` the double

doublex : `dx` (the double | `pos_inf` | `neg_inf` | `NaN`)

fuzzy : `fuzzy (` lower doublex `,` center doublex `,` upper doublex `)`

string : `s "` the string `"`

Use \ to escape \, ", {, or } (\\ ↦ \; \" ↦ "; \{ ↦ {; and \} ↦ })

box : `box` the data

data_pair : `pair <` data 1 `,` data 2 `>`

list : `list` length `<` (∅ | (data(`,` data)*)) `>`

D.5.3 recursive data syntax

A directed graph starts with a definition of its root node.

A root node has the following syntax:

$$\underline{\text{root node}} = (\underline{\text{new node}} \mid \underline{\text{molded table}})$$

An interior/child node has the following syntax:

$$\underline{\text{child node}} = (\underline{\text{new node}} \mid \underline{\text{back reference}} \mid \underline{\text{molded table}})$$

The meaning of the fields new node, back reference, and molded table are explained below:

new node syntax

A new node in the subgraph, denoted by field new node, in the directed graph is represented in text via the following format:

$$\underline{\text{new node}} = \boxed{\texttt{new}}\ \underline{\text{the data}}(\boxed{\texttt{*}} \mid (\boxed{\texttt{<}}\ \underline{\text{child node}}(\boxed{\texttt{,}}\ \underline{\text{child node}})^*\boxed{\texttt{>}}))$$

Field the data refers to the data stored directly in the current node.

The symbol `*` indicates the absence of children. A pair of triangular braces `< >` denotes a list of children separated by commas (`,`).

back reference syntax

A "back reference", denoted by field back reference, references a previously defined node in the subgraph's definition.

absolute reference Denoted by `ref` node index, an absolute reference is an integer that indexes the referenced node's position (starting from 0) when the nodes are ordered according to when they were defined.

relative reference Denoted by `addr` relative address `@`, a relative reference is a list of symbols that describes the position of the referenced node relative to the node whose child is currently being described. The relative address is a list of the following symbols. Starting with a reference to the node whose child is currently being described,

- `!` moves the reference pointer to the root node.
- `^` moves the reference pointer up to the prime parent of the node currently being referenced. The "prime parent" of a node is the parent that precedes the node during a depth-first trace.
- an integer token moves the reference pointer to the child indexed by the integer.

molded table syntax

In many scenarios, such as the representation of conditional probability tables, the directed graph depicts a multi-dimensional array in the form of a decision diagram. This decision diagram will often have a full tree structure except for some layers where the decision variables have no impact. In inactive decision layers, all children of each decision node reference the same child. Denoting a full tree using the above notation is often cumbersome, and a simpler approach would be to list all of the leaf values, alongside data describing the shape of the decision tree/diagram.

A molded table, denoted by field molded table, has the syntax:

molded table

= `table` num of layers ▭ (variable description ▭)* ▭ default value ▭ (array entry ▭)$^+$

The field variable description has the syntax:

variable description

= (`normal` | `inactive` | `specific`) variable name ▭ domain size ▭ (∅ | specific child index)

Field num of layers denotes the nonnegative number of decision layers in the decision tree, not counting the leaf node layer. The num of layers field is followed by a list of details about each layer, starting from the root layer. Each description is denoted by the field variable description, whose syntax is given above.

Each decision layer corresponds to a variable whose name is any simple data value. The variable's name, denoted by the field variable name is assigned to the data field of each node in the variable's decision layer. In addition, all nodes in a variable's decision layer have the same number of children equal to the size of the variable's domain, indicated by the field domain size. Each variable is one of three types described below:

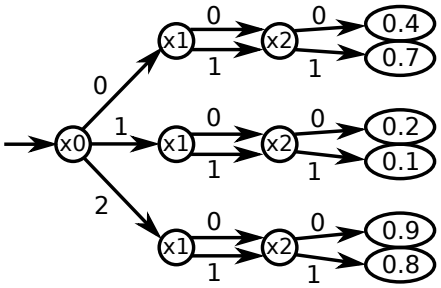
- `normal` indicates that the children of each node in the current layer all reference different children, and that the variable has an impact on the decision. The field specific child index is not used in this case.
- `inactive` indicates that the children of each node in the current layer all reference the same child. This effectively renders the variable "inactive" in the decision making process. The field specific child index is not used in this case.
- `specific` indicates that all child references except for the child indexed by specific child index reference a node that leads to a leaf node labeled with the simple data default value.

After the list of variable description fields, a simple data default value default value is provided, which is then followed by a list of simple data values array entry that populate the data field of each leaf node.

Below is shown an example molded table: the syntax on the left, and the manifested directed graph/decision diagram is shown on the right. There are 3 decision variables named

x0, x1, and x2, with respective domain sizes 3, 2, and 2. Variables x0 and x2 are "normal", while variable x1 is "inactive".

```
table 3
normal   s "x0" 3
inactive s "x1" 2
normal   s "x2" 2
0
d 0.4  d 0.7  d 0.2  d 0.1  d 0.9  d 0.8
```



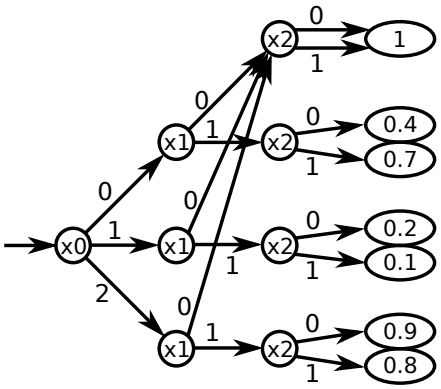
Without the molded table syntax, the decision diagram would be denoted by the following syntax:

```
new s "x0" <
    new s "x1" < new s "x2" <
        new d 0.4 * ,
        new d 0.7 * > ,
    addr 0 @ > ,
    new s "x1" < new s "x2" <
        new d 0.2 * ,
        new d 0.1 * > ,
    addr 0 @ > ,
    new s "x1" < new s "x2" <
        new d 0.9 * ,
        new d 0.8 * > ,
    addr 0 @ > >
```

Below is shown another example molded table: the syntax on the left, and the manifested directed graph/decision diagram is shown on the right. There are 3 decision variables named x0, x1, and x2, with respective domain sizes 3, 2, and 2. Variables x0 and x2 are "normal",

while variable `x1` is "specific". For the nodes labeled with `x1`, all children except for child 1 reference an identity decision diagram that always returns the specified default value: `bool 1`

```
table 3
normal   s "x0" 3
specific s "x1" 2 1
normal   s "x2" 2
bool 1
d 0.4  d 0.7  d 0.2  d 0.1  d 0.9  d 0.8
```



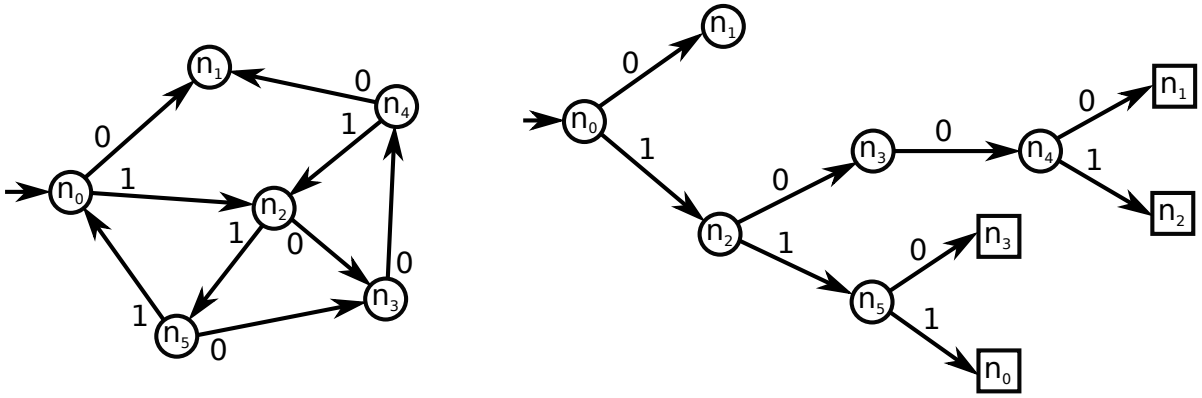
Without the molded table syntax, the decision diagram would be denoted by the following syntax:

```
new s "x0" <
    new s "x1" <
        new s "x2" <
            new bool 1 * ,
            addr 0 @ > ,
        new s "x2" <
            new d 0.4 * ,
            new d 0.7 * > > ,
    new s "x1" <
        addr ! 0 0 @ ,
        new s "x2" <
            new d 0.2 * ,
            new d 0.1 * > > ,
    new s "x1" <
        addr ! 0 0 @ ,
```

```
new s "x2" <
    new d 0.9 * ,
    new d 0.8 * > > >
```

example directed graph

The following directed graph can be denoted by the following strings:



The directed graph is shown on the left, while the depth-first tracer tree is shown on the right.

Using absolute back references gives the string:

```
new s "n_0" < new s "n_1" * , new s "n_2" < new s "n_3" < new s "n_4" < ref
1 , ref 2 > > , new s "n_5" < ref 3 , ref 0 > > >
```

Using relative back references gives the string:

```
new s "n_0" < new s "n_1" * , new s "n_2" < new s "n_3" < new s "n_4" < addr
! 0 @ , addr ! 1 @ > > , new s "n_5" < addr ! 1 0 @ , addr ! @> > >
```

or the string:

```
new s "n_0" < new s "n_1" * , new s "n_2" < new s "n_3" < new s "n_4" < addr
^ ^ ^ 0 @ , addr ^ ^ @ > > , new s "n_5" < addr ^ 0 @ , addr ^ ^ @ > > >
```

D.6 Arithmetic

Arithmetic can be performed on elements of all data types, even on elements with different data types. The main unary operators are negation and inverting, and the main binary operators are addition, subtraction, multiplication, division, maximization, and minimization. When arithmetic is performed on two subgraphs, the product subgraph is formed, and for each node in the product subgraph, the ordered pair in the data field is replaced by the result of applying the binary operator to the two entries of the ordered pair.

D.7 Input Syntax

Given an experiment number i, two input files are required: "`Test Files/`i`_data.txt`" and "`Test Files/`i`_instructions.txt`".

The file "`Test Files/`i`_data.txt`" contains the input data that is to be processed. The input data takes the form of a list of directed graphs. The format and syntax of "`Test Files/`i`_data.txt`" is the following:

number of directed graphs

name 1 graph 1

name 2 graph 2

...

name n graph n

Each name is a single token, and each graph is self contained without any intersection with the other input graphs.

The file "`Test Files/`*i*`_instructions.txt`" contains the instructional data that describes how the input data is to be processed. The format and syntax of
"`Test Files/`*i*`_instructions.txt`" is the following:

command 1 argument list 1

command 2 argument list 2

...

command m argument_list_m

Each command is followed by a list of arguments that are relevant to that command. The possible commands are listed below. Arguments are denoted by the underlined text. Each string argument must be read as a single token, which is done by enclosing the string in double quotes.

- `assign_new_subgraph` dest name new graph : Creates a new subgraph defined by new graph, and assigns the root node to dest name.
- `assign_node` dest name source name depth path indices : Assigns a node to dest name that is determined by the following: Starting from the node assigned to source name, a total of depth directed edges are traversed, indexed by path indices.
- `copy_node` dest name source name : Copies the node assigned to source name and assigns the copy to dest name. The copy node has the same children as the copied

node.

- `copy_subgraph` dest name source name : Copies the subgraph rooted at the node assigned to source name and assigns the root of the copy to dest name.

- `condense` dest name source name : Creates a condensed copy of the subgraph rooted at the node assigned to source name and assigns the root of the condensed copy to dest name.

- `expand` dest name source name level : Copies the subgraph rooted at the node assigned to source name and expands the copy subgraph at level level (0 is the root level) and assigns the root of the expanded copy to dest name.

- `subgraph_product` dest name max degree flag source name 1 source name 2 : Creates a "product subgraph" from the subgraphs rooted at source name 1 and source name 2, and assigns the root node to dest name. max degree flag is `0` if children that are not common to both nodes are excluded, and is `1` if otherwise.

- `binary_operator` dest name operator max degree flag source name 1 source name 2 : Applies the binary operator operator to the subgraphs rooted at source name 1 and source name 2 and assigns the result to dest name. `+` denotes addition; `*` denotes multiplication; `-` denotes subtraction; `/` denotes division; `max` denotes maximization; and `min` denotes minimization. max degree flag is `0` if children that are not common to both nodes are excluded, and is `1` if otherwise.

- `add_subgraphs` dest name max_degree_flag n arg_1 arg_2 ... arg_n : Assigns the sum of the n subgraphs rooted at arg_1 arg_2 ... arg_n to dest name. max_degree_flag is `0` if children that are not common to both nodes are excluded, and is `1` if otherwise.

- `multiply_subgraphs` dest name max_degree_flag n arg_1 arg_2 ... arg_n : Assigns the product of the n subgraphs rooted at arg_1 arg_2 ... arg_n to dest name. max_degree_flag is `0` if children that are not common to both nodes are excluded, and is `1` if otherwise.

- `unary_operator` dest name operator source name : Applies the unary operator operator to the subgraph rooted at source name and assigns the result to dest name. `-` denotes negation; and `/` denotes inversion.

- `negate_subgraph` dest name source name : Assigns the negative of the subgraph rooted at source name to dest name.

- `invert_subgraph` dest name source name : Assigns the inverse (reciprocal) of the subgraph rooted at source name to dest name.

- `marginalize` dest name keep degree flag source name level : Copies the subgraph rooted at source name to the dest name. The copy is then modified as follows: Trace through the subgraph rooted at dest name. When a node in the depth-first tracer tree is encountered at a depth of level, all child subgraphs are added together into a single subgraph that the child pointers now point to. If keep degree flag is `1`, then the degree of the node remains constant with all child pointers now pointing to the sum node, otherwise at most 1 child pointer that points to the sum node exists (if the degree is 0, then no children exist). Level 0 refers to the root level.

- `condition` dest name keep degree flag source name level child index : Copies the subgraph rooted at source name to the dest name. The copy is then modified as follows: Trace through the subgraph rooted at dest name. When a node in the depth-first tracer tree is encountered at a depth of level, all child subgraphs, save for the child subgraph indexed by child index, are deleted. If keep degree flag is `1`, then the degree of the node remains constant with all child pointers now pointing to the remaining child subgraph, otherwise 1 child pointer points to the remaining subgraph. Level 0 refers to the root level.

- `DS_binary_operator` dest name operator max degree flag source name 1 source name 2 : Applies the binary operator operator to the subgraphs rooted at source name 1 and

source name 2 and assigns the result to dest name. The root nodes are handled differently from the command `binary_operator`. The resultant root node has a child corresponding to each pairwise application of the binary operator to a child of source name 1 and a child of source name 2. `+` denotes addition; `*` denotes multiplication; `-` denotes subtraction; `/` denotes division; `max` denotes maximization; and `min` denotes minimization. max degree flag is `0` if children that are not common to both nodes are excluded, and is `1` if otherwise.

- `DS_collapse` dest name source name : Copies the DS structure rooted at source name and collapses it by unifying together focal element pairs with common focal elements and adding the weights; excluding empty focal elements; as well as normalizing the weights so that all weights sum to 1.

- `print_string` the string : Prints the string the string to the file "`Test Files/output_`i`.txt`" and to the console. the string must be encapsulated by double quotes to be read as a single token.

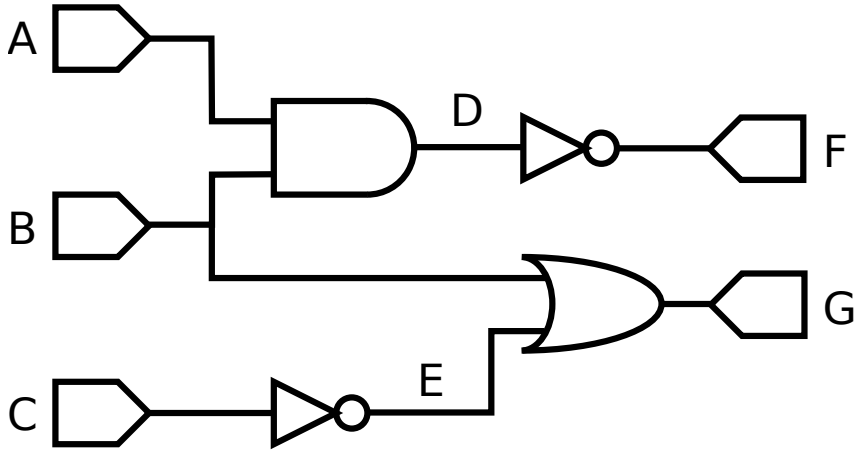
- `print_data` the source : Prints the simple data stored in the node indexed by the source to the file "`Test Files/output_`i`.txt`" and to the console.

- `print_subgraph` the source : Prints the subgraph rooted at the node assigned to the source to the file "`Test Files/output_`i`.txt`" and to the console.

- `comment` text `comment_end` : This command does nothing, and exists only to allow comments in the instructions file.

D.8 Example

This section will demonstrate the utility of the system by performing an analysis of a noisy multivalued logic circuit. The multivalued logic that is being used is ternary (base 3) logic. The noisy ternary circuit that will be under consideration is shown below:



The prior probability distributions that describe the input variables are given below:

A	0	1	2
$\Pr(A)$	0.33	0.33	0.34

B	0	1	2
$\Pr(B)$	0.7	0.2	0.1

C	0	1	2
$\Pr(C)$	0.1	0.8	0.1

$D = A$ AND B: A multivalued AND ideally returns the smaller of the two operands, but in this scenario, the output might not be pulled down to the smaller value.

$E =$ NOT C: A multivalued NOT ideally returns the complement of the operand, but in this scenario, the output might not be pulled to the correct output.

D	0	1	2
$\Pr(D\|A=0,B=0)$	1.00	0.00	0.00
$\Pr(D\|A=0,B=1)$	0.90	0.10	0.00
$\Pr(D\|A=0,B=2)$	0.80	0.10	0.10
$\Pr(D\|A=1,B=0)$	0.90	0.10	0.00
$\Pr(D\|A=1,B=1)$	0.00	1.00	0.00
$\Pr(D\|A=1,B=2)$	0.00	0.90	0.10
$\Pr(D\|A=2,B=0)$	0.80	0.10	0.10
$\Pr(D\|A=2,B=1)$	0.00	0.90	0.10
$\Pr(D\|A=2,B=2)$	0.00	0.00	1.00

E	0	1	2
$\Pr(E\|C=0)$	0.10	0.10	0.80
$\Pr(E\|C=1)$	0.10	0.80	0.10
$\Pr(E\|C=2)$	0.80	0.10	0.10

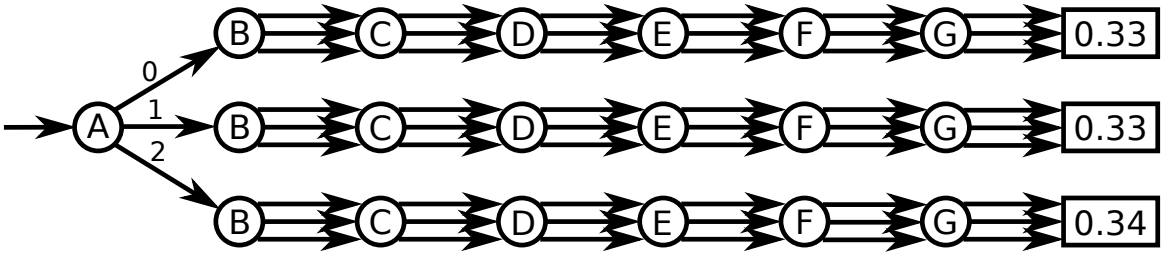
F = NOT D: A multivalued NOT ideally returns the complement of the operand, but in this scenario, the output might not be pulled to the correct output.

$G = B$ OR E: A multivalued OR ideally returns the larger of the two operands, but in this scenario, the output might not be pulled up to the larger value.

F	0	1	2
$\Pr(F\|D=0)$	0.10	0.10	0.80
$\Pr(F\|D=1)$	0.10	0.80	0.10
$\Pr(F\|D=2)$	0.80	0.10	0.10

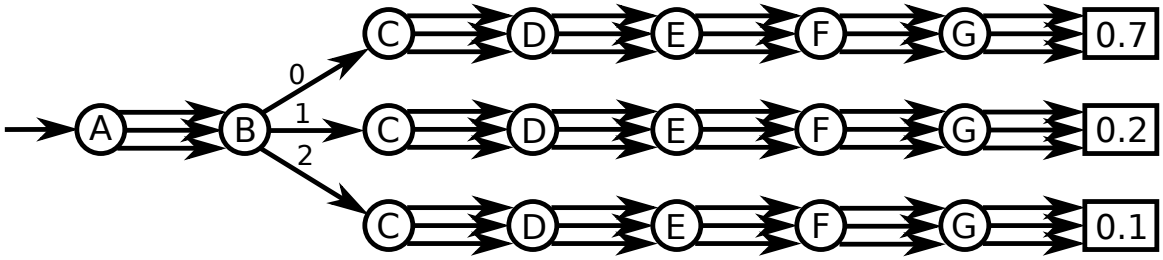
G	0	1	2
$\Pr(G\|B=0,E=0)$	1.00	0.00	0.00
$\Pr(G\|B=0,E=1)$	0.10	0.90	0.00
$\Pr(G\|B=0,E=2)$	0.10	0.10	0.80
$\Pr(G\|B=1,E=0)$	0.10	0.90	0.00
$\Pr(G\|B=1,E=1)$	0.00	1.00	0.00
$\Pr(G\|B=1,E=2)$	0.00	0.10	0.90
$\Pr(G\|B=2,E=0)$	0.10	0.10	0.80
$\Pr(G\|B=2,E=1)$	0.00	0.10	0.90
$\Pr(G\|B=2,E=2)$	0.00	0.00	1.00

The decision diagram and syntax for $\Pr(A)$:



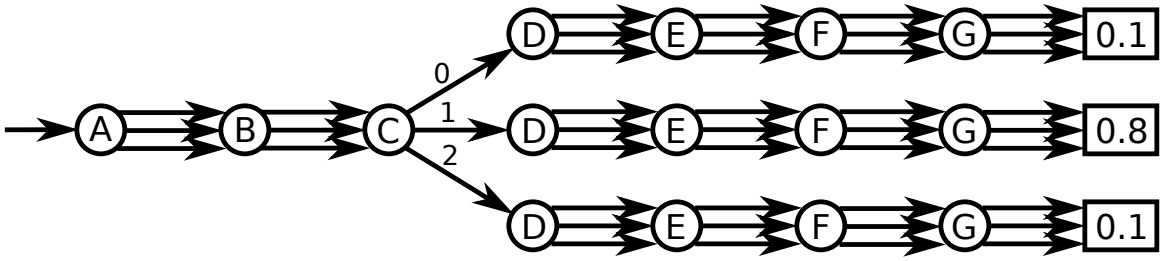

```
table 7
normal   s "A" 3  inactive s "B" 3  inactive s "C" 3  inactive s "D" 3
inactive s "E" 3  inactive s "F" 3  inactive s "G" 3
0
d 0.33  d 0.33  d 0.34
```

The decision diagram and syntax for Pr(B):



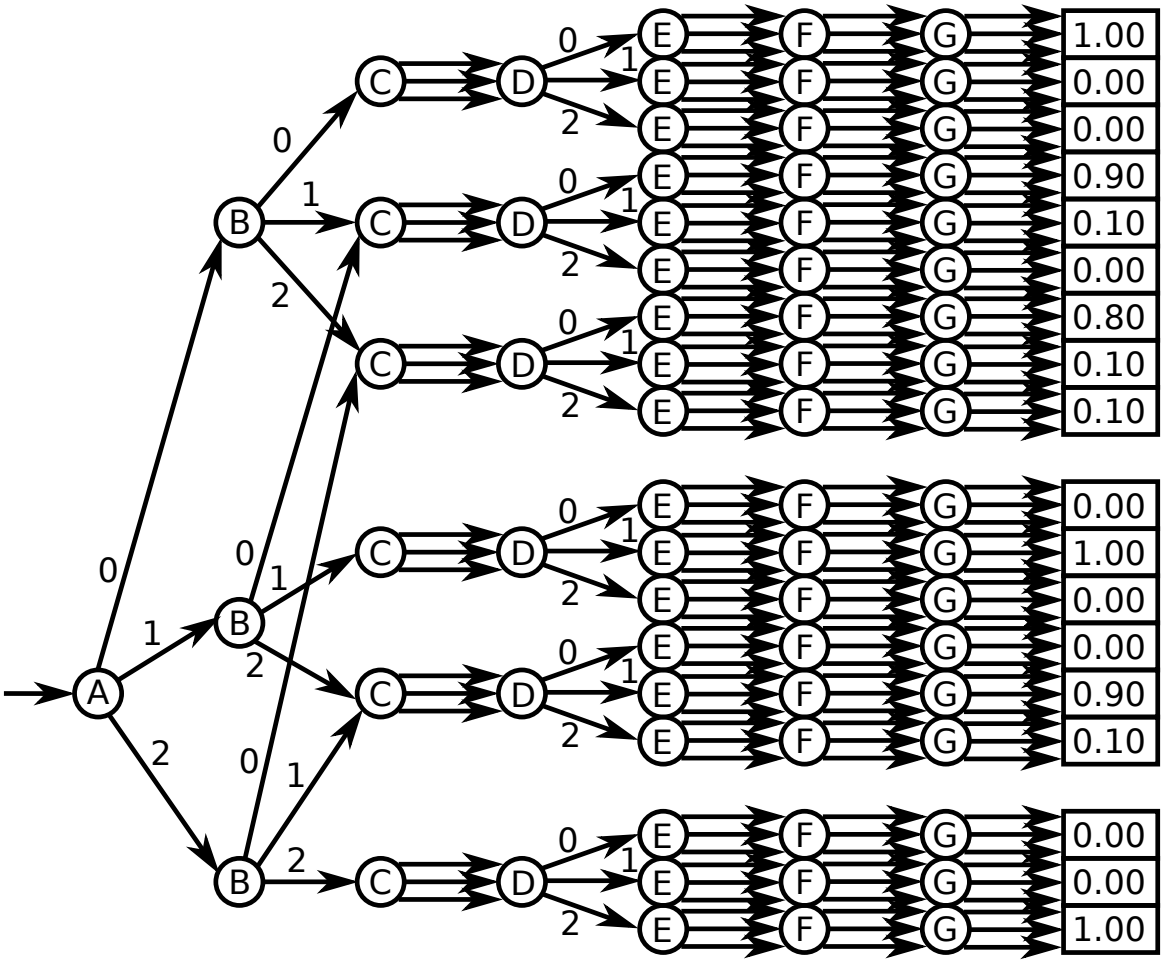

```
table 7
inactive s "A" 3  normal   s "B" 3  inactive s "C" 3  inactive s "D" 3
inactive s "E" 3  inactive s "F" 3  inactive s "G" 3
0
d 0.7  d 0.2  d 0.1
```

The decision diagram and syntax for Pr(C):




```
table 7
inactive s "A" 3  inactive s "B" 3  normal   s "C" 3  inactive s "D" 3
inactive s "E" 3  inactive s "F" 3  inactive s "G" 3
0
d 0.1  d 0.8  d 0.1
```

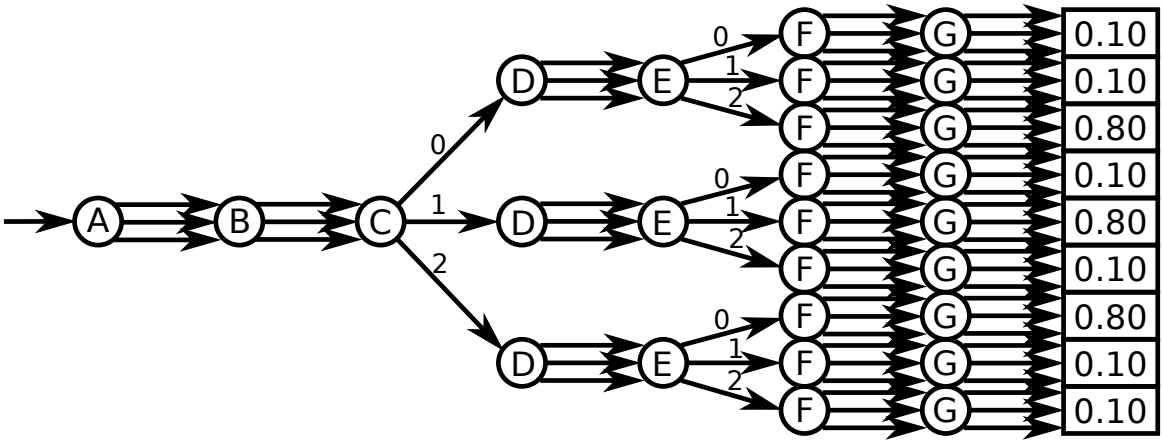
The decision diagram and syntax for $\Pr(D|A,B)$:




```
new s "A" <
    {A=0} new s "B" <
        {B=0} table 5
        inactive s "C" 3  normal   s "D" 3  inactive s "E" 3  inactive s "F" 3
        inactive s "G" 3
        0
        d 1.00  d 0.00  d 0.00 ,
        {B=1} table 5
        inactive s "C" 3  normal   s "D" 3  inactive s "E" 3  inactive s "F" 3
        inactive s "G" 3
```

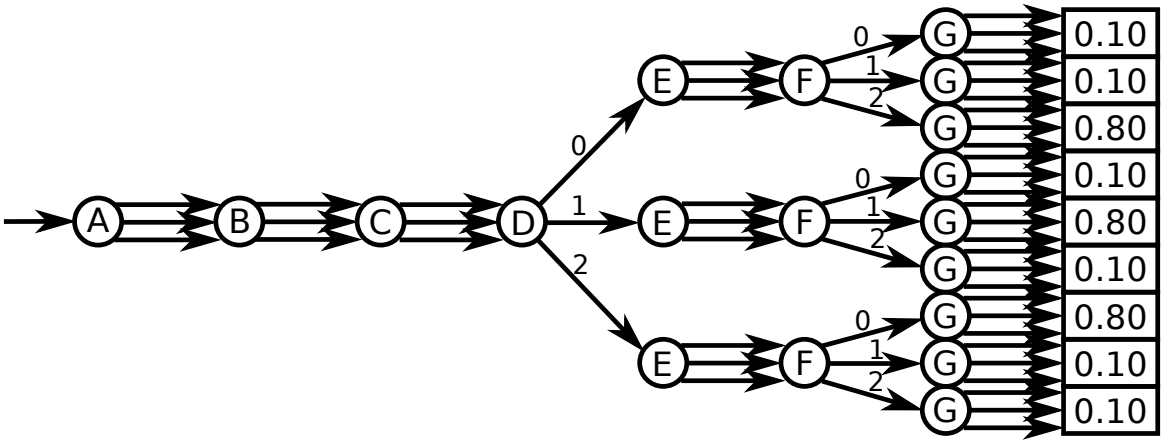
```
    0
    d 0.90  d 0.10  d 0.00 ,
    {B=2} table 5
    inactive s "C" 3  normal   s "D" 3  inactive s "E" 3  inactive s "F" 3
    inactive s "G" 3
    0
    d 0.80  d 0.10  d 0.10 > ,
{A=1} new s "B" <
    {B=0} addr ! 0 1 @ ,
    {B=1} table 5
    inactive s "C" 3  normal   s "D" 3  inactive s "E" 3  inactive s "F" 3
    inactive s "G" 3
    0
    d 0.00  d 1.00  d 0.00 ,
    {B=2} table 5
    inactive s "C" 3  normal   s "D" 3  inactive s "E" 3  inactive s "F" 3
    inactive s "G" 3
    0
    d 0.00  d 0.90  d 0.10 > ,
{A=2} new s "B" <
    {B=0} addr ! 0 2 @ ,
    {B=1} addr ! 1 2 @ ,
    {B=2} table 5
    inactive s "C" 3  normal   s "D" 3  inactive s "E" 3  inactive s "F" 3
    inactive s "G" 3
    0
    d 0.00  d 0.00  d 1.00 > >
```

The decision diagram and syntax for $\Pr(E|C)$:



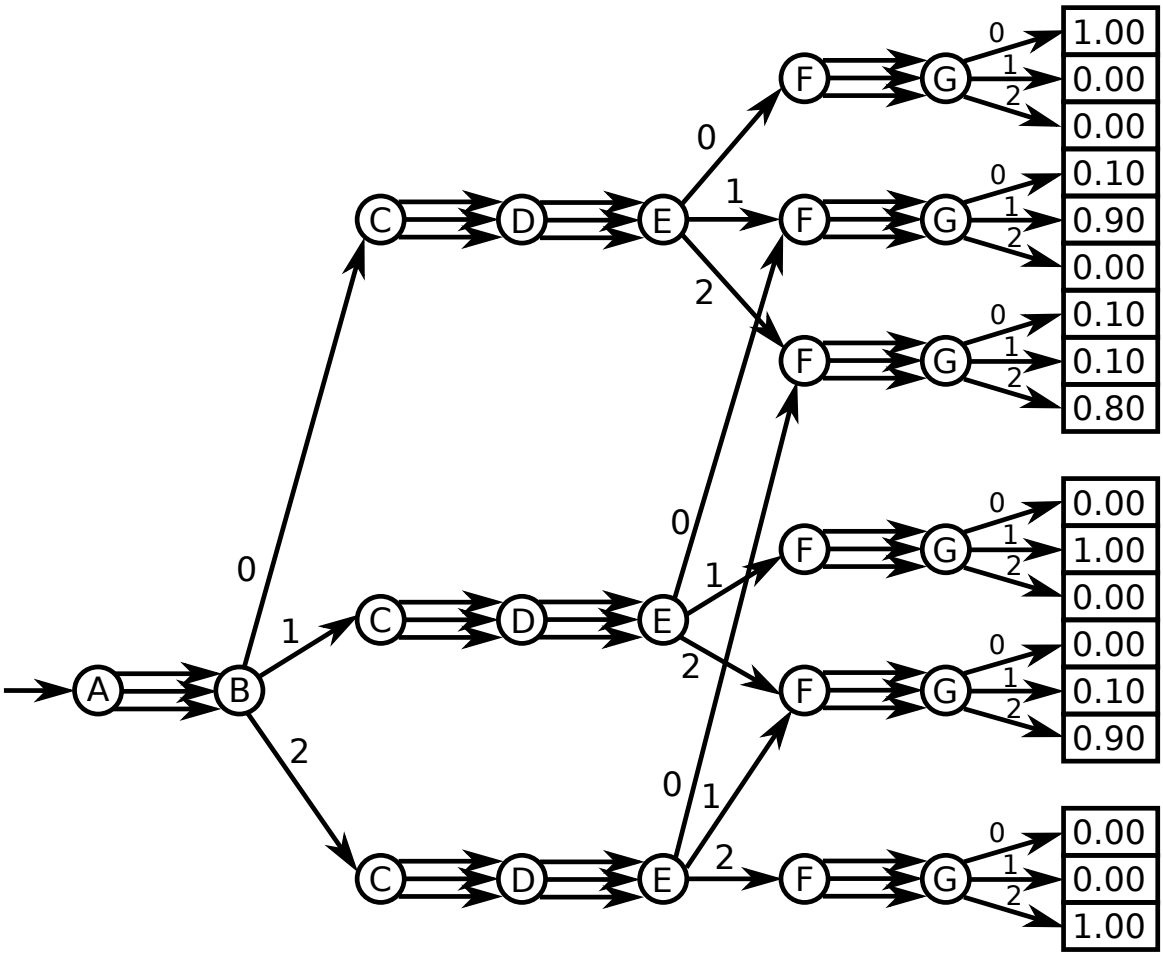

```
table 7
inactive s "A" 3  inactive s "B" 3  normal   s "C" 3  inactive s "D" 3
normal   s "E" 3  inactive s "F" 3  inactive s "G" 3
0
d 0.10  d 0.10  d 0.80   d 0.10  d 0.80  d 0.10   d 0.10  d 0.10  d 0.80
```

The decision diagram and syntax for $\Pr(F|D)$:




```
table 7
inactive s "A" 3  inactive s "B" 3  inactive s "C" 3  normal   s "D" 3
inactive s "E" 3  normal   s "F" 3  inactive s "G" 3
0
d 0.10  d 0.10  d 0.80   d 0.10  d 0.80  d 0.10   d 0.10  d 0.10  d 0.80
```

The decision diagram and syntax for $\Pr(G|B,E)$:




```
new s "A" < new s "B" <
    {B=0} new s "C" < new s "D" < new s "E" <
        {E=0} table 2
        inactive s "F" 3  normal   s "G" 3
        0
        d 1.00  d 0.00  d 0.00 ,
        {E=1} table 2
        inactive s "F" 3  normal   s "G" 3
        0
        d 0.10  d 0.90  d 0.00 ,
```

```
        {E=2} table 2
        inactive s "F" 3  normal   s "G" 3
        0
        d 0.10  d 0.10  d 0.80 > ,
    addr 0 @ , addr 0 @ > , addr 0 @ , addr 0 @ > ,
    {B=1} new s "C" < new s "D" < new s "E" <
        {E=0} addr ! 0 0 0 0 1 @ ,
        {E=1} table 2
        inactive s "F" 3  normal   s "G" 3
        0
        d 0.00  d 1.00  d 0.00 ,
        {E=2} table 2
        inactive s "F" 3  normal   s "G" 3
        0
        d 0.00  d 0.10  d 0.90 > ,
    addr 0 @ , addr 0 @ > , addr 0 @ , addr 0 @ > ,
    {B=2} new s "C" < new s "D" < new s "E" <
        {E=0} addr ! 0 0 0 0 2 @ ,
        {E=1} addr ! 0 1 0 0 2 @ ,
        {E=2} table 2
        inactive s "F" 3
        normal   s "G" 3
        0
        d 0.00  d 0.00  d 1.00 > ,
    addr 0 @ , addr 0 @ > , addr 0 @ , addr 0 @ > > ,
addr 0 @ , addr 0 @ >
```

D.8.1 Probabilistic Inference Example

This section will detail using the above data alongside the above instructions to extract data about the above noisy ternary logic circuit.

Consider a scenario where it is known that $A = 1$, $B = 0$, and $C = 0$. Of interest is the joint marginal probability distribution for the outputs F and G.

The contents of the instructions file `Test Files/1_instructions.txt` is:

```
{Applying the evidence}

copy_subgraph "Pr(A|evidence)" "Pr(A)"
condition "Pr(A|evidence)" 0 "Pr(A|evidence)" 0 1
condition "Pr(A|evidence)" 0 "Pr(A|evidence)" 1 0
condition "Pr(A|evidence)" 0 "Pr(A|evidence)" 2 0

copy_subgraph "Pr(B|evidence)" "Pr(B)"
condition "Pr(B|evidence)" 0 "Pr(B|evidence)" 0 1
condition "Pr(B|evidence)" 0 "Pr(B|evidence)" 1 0
condition "Pr(B|evidence)" 0 "Pr(B|evidence)" 2 0

copy_subgraph "Pr(C|evidence)" "Pr(C)"
condition "Pr(C|evidence)" 0 "Pr(C|evidence)" 0 1
condition "Pr(C|evidence)" 0 "Pr(C|evidence)" 1 0
condition "Pr(C|evidence)" 0 "Pr(C|evidence)" 2 0

copy_subgraph "Pr(D|evidence)" "Pr(D|A,B)"
condition "Pr(D|evidence)" 0 "Pr(D|evidence)" 0 1
condition "Pr(D|evidence)" 0 "Pr(D|evidence)" 1 0
```

```
condition "Pr(D|evidence)" 0 "Pr(D|evidence)" 2 0

copy_subgraph "Pr(E|evidence)" "Pr(E|C)"
condition "Pr(E|evidence)" 0 "Pr(E|evidence)" 0 1
condition "Pr(E|evidence)" 0 "Pr(E|evidence)" 1 0
condition "Pr(E|evidence)" 0 "Pr(E|evidence)" 2 0

copy_subgraph "Pr(F|evidence)" "Pr(F|D)"
condition "Pr(F|evidence)" 0 "Pr(F|evidence)" 0 1
condition "Pr(F|evidence)" 0 "Pr(F|evidence)" 1 0
condition "Pr(F|evidence)" 0 "Pr(F|evidence)" 2 0

copy_subgraph "Pr(G|evidence)" "Pr(G|B,E)"
condition "Pr(G|evidence)" 0 "Pr(G|evidence)" 0 1
condition "Pr(G|evidence)" 0 "Pr(G|evidence)" 1 0
condition "Pr(G|evidence)" 0 "Pr(G|evidence)" 2 0

{multiplying together the conditioned factors}

binary_operator "Pr(total)" * 1 "Pr(A|evidence)" "Pr(B|evidence)"
binary_operator "Pr(total)" * 1 "Pr(total)" "Pr(C|evidence)"
binary_operator "Pr(total)" * 1 "Pr(total)" "Pr(D|evidence)"
binary_operator "Pr(total)" * 1 "Pr(total)" "Pr(E|evidence)"
binary_operator "Pr(total)" * 1 "Pr(total)" "Pr(F|evidence)"
binary_operator "Pr(total)" * 1 "Pr(total)" "Pr(G|evidence)"

{marginalizing out the unnecessary variables}
```

```
marginalize "Pr(total)" 0 "Pr(total)" 3
marginalize "Pr(total)" 0 "Pr(total)" 4

{calculating and applying the normalization constant}

copy_subgraph "Z" "Pr(total)"
marginalize "Z" 1 "Z" 5
marginalize "Z" 1 "Z" 6

binary_operator "Pr(total)" / 1 "Pr(total)" "Z"

{print the posterior probabilities}

assign_node "probe" "Pr(total)" 7 0 0 0 0 0 0 0
print_data "probe"
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 0 1
print_data "probe"
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 0 2
print_data "probe"
print_string ""
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 1 0
print_data "probe"
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 1 1
print_data "probe"
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 1 2
print_data "probe"
```

```
print_string ""
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 2 0
print_data "probe"
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 2 1
print_data "probe"
assign_node "probe" "Pr(total)" 7 0 0 0 0 0 2 2
print_data "probe"
```

The contents of `Test Files/1_output.txt` is:

```
d 0.019000

d 0.017000

d 0.064000



d 0.032300

d 0.028900

d 0.108800



d 0.138700
```

```
d 0.124100
```

```
d 0.467200
```

This yields the following posterior probability distribution over F and G:

G	0	1	2
$\Pr(F = 0, G)$	0.0190	0.0170	0.0640
$\Pr(F = 1, G)$	0.0323	0.0289	0.1088
$\Pr(F = 2, G)$	0.1387	0.1241	0.4672

Appendix E

Copyright Information

This thesis includes some portions of IEEE copyrighted papers in which I am a co-author. No IEEE copyrighted paper has been included entirely. Only some portions of the IEEE copyrighted papers have been used. The papers are cited and IEEE copyright is acknowledged whenever applicable.