2018-11-20

Novel Methods for Reasoning with Uncertain Hard and Soft Data using Probabilistic and Belief Theoretic Methods

Rafael Camilo Nunez Sanchez
University of Miami, rafaelcns@gmail.com

UNIVERSITY OF MIAMI

NOVEL METHODS FOR REASONING WITH UNCERTAIN HARD AND SOFT DATA USING PROBABILISTIC AND BELIEF THEORETIC METHODS

By

Rafael Camilo Núñez Sánchez

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

December 2018

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

NOVEL METHODS FOR REASONING WITH UNCERTAIN HARD AND SOFT DATA USING PROBABILISTIC AND BELIEF THEORETIC METHODS

Rafael Camilo Núñez Sánchez

Approved:

Manohar N. Murthi, Ph.D.
Associate Professor of Electrical and Computer Engineering

Kamal Premaratne, Ph.D.
Professor of Electrical and Computer Engineering

Xiaodong Cai, Ph.D.
Professor of Electrical and Computer Engineering

Miroslav Kubat, Ph.D.
Associate Professor of Electrical and Computer Engineering

Otávio Bueno, Ph.D.
Professor of Philosophy

Guillermo J Prado, Ph.D.
Dean of the Graduate School

NÚÑEZ SÁNCHEZ, RAFAEL C. (Ph.D., Electrical and Computer Engineering)

Novel Methods for Reasoning with Uncertain Hard and Soft Data using Probabilistic and Belief Theoretic Methods (December 2018)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Associate Professor Manohar N. Murthi.
No. of pages in text. (124)

Effectively combining multiple and complementary sources of information is becoming one of the most promising paths for increased accuracy and more detailed analysis in numerous applications. Neuroscience, business analytics, military intelligence, and sociology are among the areas that could significantly benefit from properly processing diverse data sources. However, traditional methods for combining multiple sources of information are based on slow or impractical methods that rely either on vast amounts of manual processing or on suboptimal representations of data. Moreover, most of the existing methods are not well suited for dealing with the increasing amount of human-generated data. We introduce an analytical framework that allows automatic and efficient processing of both hard (e.g., physics-based sensors) and soft (e.g., human-generated) information, leading to enhanced decision-making in multisource environments. This framework is based on the Dempster-Shafer (DS) Theory of Evidence as the common language for data representation and inference. To model and track uncertainties in soft data, our framework introduces Uncertain Logic, a classically consistent first order logic environment. In addition, our framework defines a filtering and tracking environment for incorporating both hard and soft data, where the probability posterior can be decomposed into a product of combining functions over subsets of the state and measurement variables. This combining function approach offers a framework for the development and incorporation of more sophisticated uncertainty modeling and tracking/estimation models, and at the same time allows incorporating and enhancing existing Bayesian methods. Future work

is aimed at increasing the computational efficiency of the overall hard and soft data fusion framework.

To my family

Acknowledgements

I would like to express my deepest appreciation to my advisor, Professor Manohar Murthi, as well as to my co-advisor, Professor Kamal Premaratne, for their strong commitment to make me succeed in pursuing a doctoral degree. Their support, encouragement, and positively challenging research environment were fundamental for making this dissertation work both relevant and of the highest quality.

I would also like to extend my gratitude to Professors Otávio Bueno and Matthias Scheutz for their constant guidance and discussions on mathematical logic, as well as to Professors Xiadong Cai and Miroslav Kubat who, as members of the committee, contributed with valuable feedback to this work.

Many thanks to Professors Mohamed Abdel-Mottaleb and James Modestino for their support and for always having an open door for discussions of any kind, as well as to all the Faculty at the Department of Electrical and Computer Engineering.

Thanks should also go to my colleagues and fellow students at EB508, Doctors Thanuka Wickramaratne, Ranga Dabarera, Buddhika Samarakoon, Janith Heendeni, Tao Meng, Oscar Martinez, Zifang Huang, Dianting Liu, Qiusha Zhu, and Chao Chen, as well as to Lalintha Polpitiya, and May Zar Lin. They all contributed to making my studies and work at the University of Miami a very enriching and gratifying experience.

I also appreciate all the help and great attitude of the Electrical and Computer Engineering staff during my time as a student here: Michelle Perez, Angie Del-Llano, Rosamund Coutts, Kendra Parks, Ivelisse Reyes, Frances Nicholas, and Samantha Valea.

Just as important, I want to express my total gratitude to my wife, Sherin, for the love, patience, and support that allowed me to complete this dissertation. My daughters Natalia Lili and Eliza María have also been a great motivation for completing this dissertation. My mother, Constanza, my sisters Adriana and Claudia, and my aunt Katia, have always been supportive from the distance. Special thanks to my father, who is not with us today, for his guidance and support since the beginning of my professional career.

Rafael Camilo Núñez Sánchez

University of Miami

December 2018

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Background and Motivation

The focus of traditional data fusion systems is expanding beyond the use of physical sensors to observe, locate, and characterize physical targets to the combined use of physical and nonphysical sensors to observe the human landscape: individuals, groups, populations, organizations, and their interactions. In addition to refocusing from the physical to the human landscape, there is increasing interest in the utilization of nonphysical sensors including humans acting as observers (human as "soft sensors") and the use of other types of data available on the Web.

David L. Hall and John M. Jordan
Human-Centered Information Fusion, 2010 [3]

The excerpt above, extracted from Hall and Jordan's recent book, "Human-Centered Information Fusion" [3], succinctly summarizes the need for incorporating *soft* data (e.g., text from witness statements, blogs, newspapers) into modern data fusion systems. Successfully incorporating soft data into already existing (and usually sophisticated) *hard* data (i.e., generated by physical sensors) methods, could provide immediate solutions in many problem contexts, and significant improvements in numerous applications. For example, consider a surveillance scenario in which witness statements about suspicious behavior provide valuable soft data pertinent to the activity of criminals that must be combined with hard data from traditional surveillance sensors. Or consider a public health scenario in which patient statements, and doctor opinions must be combined with biomedical sensor hard data to detect and track the spread of illnesses.

By its nature, soft data in the form of text is more qualitative in nature, inherently possessing uncertainty with regards to semantics. In addition to the semantic uncertainty, the source (e.g., informant, blogger) of the text may not be reliable, and the text statement itself may not be credible. The first question that arises is, how should one parse and represent the soft data? Also, how does one combine inherently uncertain soft data text with hard data in an automated fusion, estimation, and tracking system? Clearly, there is a growing necessity for the development of new methods in fusion, estimation, and tracking that can incorporate and suitably combine heterogeneous forms of both hard and soft data.

1.1 Requirements and Challenges

The study and analysis of human-based information, as well as human interactions, has been traditionally addressed by social sciences, which includes, among other disciplines, linguistics, psychology, sociology, and political science. Some of the biggest challenges faced by these disciplines include abstracting relevant information to focus the attention to the analysis of the problem at hand, measuring and characterizing subjective information, understanding and describing reasons behind human behavior, among many others. Depending on the application, hard and soft data fusion would inherit most (if not all) of these challenges, making this type of fusion a heavily interdisciplinary task. In addition, hard and soft data fusion brings new challenges, such as finding appropriate representations for soft data, combining non-physical information systems with existing hard data processors, and addressing computational complexity. This work is focused on providing solutions to these new challenges, which are further described next.

1. **Data representation.** Soft data is naturally given in the form of speech or text inputs. However, these data formats are not well suited for an automated data processing system, as they are not in a standardized form. Computational linguistics addresses this problem, with the goal of preserving the largest possible amount of semantic information from natural language expressions. When building the corresponding representation in soft data fusion systems, our goal is exactly the same. In linguistics, First Order Logic (FOL) becomes relevant for addressing this requirement. Considering the advantages of FOL in preserving more semantic information compared to other soft data representations (e.g., RDF graphs) [4], and to the availability of natural language processing systems that automatically parse text information into logic sentences, we use FOL for data representation in our fusion solutions.

2. **Modeling and tracking uncertainties in soft data.** This requirement refers to the proper modeling of uncertainties in soft data, as well as tracking their evolution in inference processes. The more qualitative nature of soft data makes uncertainty management more challenging than in the case of hard data. Representing uncertainties using intervals, as enabled by Dempster-Shafer (DS) theory, is an intuitive and advantageous alternative to conventional probabilistic models for uncertainty management [5, 4].

3. **Robustness against conflicting evidence.** Note that conflicting evidence can easily lead to contradictions that make accurate reasoning impossible. This is actually not an unlikely event in fusion that incorporates soft data, as we can expected to receive large quantities of information from humans with various degrees of reliability or even with an intent for deception. The goal is to be able to process conflicting evidence without relying on drastic solutions (e.g., discarding both correct and incorrect evidence indiscriminately). Fusion operators within the DS theory framework such as the Conditional Fusion Equation [6]

provide various degrees of robustness against conflicting evidence. The robustness against conflicting evidence, as well as its strength as an uncertainty modeling system for soft data reasoning, make DS theory our primary mathematical framework for the new methods introduced in this manuscript.

4. **Suitable combination of existing hard data processors with both existing and new soft data processing systems.** While natural language processing can convert text into uncertainty-augmented first order logic, the question of how to incorporate such soft data models into estimation and tracking remains an important fundamental issue. An important challenge in terms of the impact that soft data fusion can have in enhancing existing (hard data-based) fusion systems, is the ability to combine or extend Bayesian filtering with soft data processing capabilities.

5. **Reduced computational complexity.** The requirement is critical, as it is well known that logic reasoning systems are computationally demanding. Then, alternatives for reducing complexity must be considered. One of these alternatives is to solve a satisfiability problem in which, instead of using inference rules (e.g., Modus Ponens) to derive conclusions and their associated uncertainties, we find the logical values (or uncertainties) that are unknown for the propositions in a set of logic formulas (in our case, this set of formulas is the FOL representation of the soft data) such that the full set of logic formulas (i.e., our logic model) is true or satisfiable.

1.2 Application Scenarios

The range of application of soft and hard data fusion methods is vast, involving virtually every scenario where humans have traditionally played a key role either in producing or analyzing information. To better illustrate the impact and benefits of the work described in this dissertation, we can consider the following sampling of potential applications.

- **Detection and Tracking.** A recent trend motivated by irregular/asymmetric warfare cases considers the problem of incorporating non conventional sources such as human sources for tracking applications [7]. The emergence of technologies such as smartphones provides humans with means to rapidly share what they can see to a centralized system. The fundamental hardware is then already available for using humans as "soft" sources in detection and tracking systems. When combined with hard data fusion systems, soft-data enhanced tracking systems could provide more accurate information in difficult scenarios such as urban environments. In addition, humans could also provide knowledge (e.g., in the form of if-then rules) that can aid in prediction of events in a tracking scenario. As will be seen later in this manuscript, our new methods could be directly used in detection and tracking applications.

- **Situation Awareness.** Situation awareness is the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future [8]. Incorporation of automated soft data processing into situation awareness can be valuable in a wide array of applications, including search and rescue operations, battlefield reconnaissance, surveillance, politics, etc. For example, Twitter updates could be used to assess the status after an emergency (e.g., earthquake), as the inhabitants of a region of interest can provide on-the-ground information, including expressions of fear, requests for help, and disasters's impact on the community [9]. Manually extracting information from social media would be a daunting task, making the introduction of a reliable system that can accurately fuse the available soft (and perhaps hard) data an important need.

- **Crowdsourcing.** Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call [10]. It allows the power of the crowd to accomplish tasks that were once the province of just a specialized few. Successful examples of crowdsourcing include Wikipedia, Waze traffic application, and YouTube, among others. These and new applications of crowdsourcing could benefit from more efficient ways of categorizing and processing knowledge that is expressed in natural language, such as the ones presented in this manuscript.

- **Question Answering Systems.** Question answering systems aim at automatically answering questions that are posed by humans in natural language. Unlike existing information retrieval systems (e.g., search engines), which simply render a list of matching documents, the main objective of question answering systems is to retrieve answers to questions rather than full documents or best matching passages, as most information retrieval systems currently do [11]. Existing answering systems such as Wolfram Alpha, as well as IBM Watson, are already fully functional systems able to parse natural language and either search or compute answers based on database searches or numerical operations. However, there is a need for addressing more complex types of questions or, in general, questions that may extend beyond the scope of what is already available in a database (e.g., why-questions). In addition, these systems could also benefit from more accurate tracking of imperfections or uncertainties in both the input data, as well as uncertainties in the possible answers.

- **Human-Robot Interaction.** Human-robot interactions require robots to be capable of dealing with spoken natural language dialogues [12]. In addition to accounting for imperfections in the speech processing and natural language parsing, the robot must be able to cope with various types of errors, inaccuracies, and imperfections of human data, as well as indirect speech acts (e.g., dealing with expressions like "I would like a coffee" instead of a direct order "get me a coffee"). A robust soft data processing framework such as the one being developed in this research work could provide significant improvements to interfaces and decision systems within the human-robot interaction scope.

- **Medical Diagnosis.** Medical diagnosis is an important task that should be performed as accurately and efficiently as possible. As with any human driven process, it is inherently influenced by biases. If a doctor has an assumed diagnosis, she will immediately begin searching for further evidence that her assumption can be validated, missing other potential diagnoses. Additionally, if the doctor begins searching by symptoms, while these may be accurate, the order or weight given to any one symptom will give a bias toward related diagnosis when in fact, there may be a symptom not given any credit and thus not included in the analysis [13]. We then need an approach that minimizes human bias and considers all relevant and irrelevant data in determining a diagnosis. With an automated medical diagnosis system, doctors could be presented with multiple potential diagnoses based on all of the patient's current and past details. Such a system could be designed using the solutions introduced in this manuscript.

1.3 Related Research

Two research areas are directly related to our work on hard and soft data fusion. The first area deals with modeling uncertainty in logic reasoning. This area has been studied for several decades now, but the challenges introduced by soft data fusion require further improvements to the existing solutions. The second area can be named directly as hard and soft data fusion. This is a younger research topic, in which our efforts are focused on foundational work.

1.3.1 Modeling Uncertainty in Logic Reasoning

Research on the area has traditionally targeted a number of distinct goals, such as investigating the source and meaning of uncertainty, enriching of logic systems with appropriated formalisms for uncertainty management (e.g., semantics, axioms), and creating appropriate models and operators to quantify the propagation of uncertainty in reasoning and inference problems. The latter is more related to the uncertain logic work presented in this manuscript. However, it is necessary to understand advancements on the other areas to contextualize and properly apply the methods introduced herein.

Relevant foundational work on analyzing the source and representation of uncertainty in logic systems can be found in [14]. In this work, the author introduces two different approaches to giving semantics to first-order logics of probability, the first one incorporating probability in the domain (for problems involving statistical information), and the second one assigning probabilities to possible worlds. This work is extended in [15], where the author further discusses the use of a "possible-worlds" framework to represent and reason about uncertainty. Then, quantification of the uncertainty is accomplished by assigning a probability distribution to the possible worlds. In addition, the author discusses the importance of considering time in the inference process, i.e., possible words should describe states at each time point of interest. The work in [16] provides insight on how to process and combine data-driven (e.g., information obtained from observed events) and knowledge-driven (e.g., information provided by domain experts) using different logic systems.

In addition to first-order logic, uncertain representations of logic systems have been extended to other types of logic. For example, the work in [17] introduces a multi-agent epistemic logic able to represent and merge partial beliefs of multiple agents. This logic system is based on possibility theory [18], and enhances epistemic logic with parametric models to obtain lower bounds on the degree of belief of agents.

Similarly, an axiomatization of a modal logic using fuzzy sets and DS belief functions for measuring probabilities of modal necessity is presented in [19].

When addressing quantification and propagation of uncertainty in logic reasoning systems, one of the most important approaches is probabilistic logic [20]. Probabilistic logic provides a generalization of logic in which the truth values of sentences are probability values (between 0 and 1). A related approach, possibilistic logic [21], defines mechanisms (based on possibility theory) to associate classical logic formulas with weights. These weights represent lower bounds of necessity degrees. To efficiently address the computational complexity of larger problems, two new probability-enhanced methods have been recently introduced, namely, MLNs [22] and PSL [23]. MLNs combine FOL and Markov networks as undirected graphical models and assign weights to FOL formulas whose truth values are represented through probability values. PSL also uses undirected graphical models to represent templates for FOL formulas, and represents truth values as a number in the interval $[0, 1]$. PSL restricts FOL formulas to conjunctive bodies. It relies on the Lukasiewicz t-norm and its corresponding co-norm to model AND and OR operations. Both MLN and PSL are statistical methods that attempt to find uncertainty parameters that ensure satisfiability of the logical models specified by the users. Their definition, however, does not insist on being consistent with classical logic, as is our goal with ULP, and their accuracy may be compromised when the amount of training data is small.

Other approaches that extend logic reasoning to address uncertain scenarios are many-valued and fuzzy logics. Many-valued logics [24] do not restrict the number of truth values of propositions to two. The interpretation of the truth values depends on the actual application. Fuzzy logic can be seen as a type of many-valued logic. Fuzzy logic is based on the theory of fuzzy sets [25]. In fuzzy logic, the imprecision in probabilities is modeled through membership functions defined on the sets of possible probabilities and utilities.

Regarding the use of intervals as means of representing uncertainty, it appears in several methods, such as possibility theory [26] and DS theory. The latter, in addition, incorporates a rigorous methodology for assigning probabilistic measures based on available evidence [27]. Given the direct relation that exists between DS theory and probability (DS belief and plausibility measures correspond precisely to probabilistic inner and outer measures [27]), it is possible to simplify DS models to probabilistic models. Considering these advantages, a number of researchers have studied the relation of DS theory and logic. In [28], DS theory is formulated in terms of propositional logic, enabling certain logic reasoning operations in the DS framework. Insight into the relationship between DS theory and probabilistic logic is presented in [28]. A belief-function logic that uses DS models and operations to quantify and estimate uncertainty of logic formulas is introduced in [29]. This logic system allows non-zero belief assignments to the empty set, relies on Dempster's combination rule as the method for quantifying the propagation of uncertainty, and is used in deduction systems where the logic formulas are in Skolemized normal conjunctive form. An application of this system for inference is described in [30]. Further analysis on DS-based logic is presented in [31]. A detailed study on uncertain implication rules is in [32]. This latter work, however, is not focused on ensuring consistency with classical logic, but on modeling causal probabilistic relations.

In spite of existing research to provide logic with uncertainty modeled by DS, efforts to date can be improved by ensuring consistency with classical logic and reducing the number of assumptions needed for the logic systems to work. For example, most of the existing methods are based on Dempster's Combination Rule, which, as it is shown in this manuscript, is not necessarily well suited for logical reasoning. In addition, inference processes could benefit from eliminating the condition that logic formulas need to be expressed in normal conjunctive form or as implication rules, as well as eliminating the need for allowing non-zero belief assignments to the empty set in a DS model.

1.3.2 Hard and Soft Data Fusion

Due to the novelty of the hard and soft data fusion area, existing work is less abundant and more recent than the above described uncertainty management in logic systems. The work in [33] used a random set approach to fuse natural language propositions from a very restricted spatial vocabulary, and demonstrated efficacy in an experiment involving the detection and location of a fixed target. The work in [7] focused on tracking, and took a related approach, similarly imposed severe limitations on the soft data, in this case with regards to semantics. In [34], the authors capture soft data using probability and First Order Logic (FOL) to model associations between hard and soft variables, extending the framework of Markov Logic Networks. In [35] the author introduces a particle filter for tracking in an urban scenario given soft binary data. The work in [36] focused on fusion architectures, using 2nd-degree uncertainty models and hybrid Bayesian/fuzzy-logic inference schemes. Our work in [37, 38, 39, 6, 4] has focused on utilizing Dempster-Shafer (DS) belief theory, developing new methods for modeling and fusing heterogeneous soft/hard data sources with pervasive uncertainty, leading to developments in estimating source and evidence reliability/credibility, sensor consensus estimation, modeling/fusion of imperfect FOL, and computational efficiency. All of these previous papers have contributed various attempts at grappling with the nature of soft and hard data fusion and estimation. In this dissertation we propose an approach that complements these papers, allowing to efficiently (i.e., scalable solutions) process soft data, and to embed existing (and usually sophisticated) hard data processors.

1.4 Key Contributions

The academic contributions in this dissertation can be grouped in three main areas, namely: 1) Uncertain Logic Processing; 2) combining functions for hard and soft data fusion; and 3) computational efficiency improvements for hard and soft data fusion.

1. **Uncertain Logic Processing.** We introduce *Uncertain Logic Processing* (ULP), a DS theoretic approach for first order logic operations. ULP provides support for handling variables and quantifiers, in addition to fundamental logic operations (i.e., $\neg, \wedge, \vee$). This framework allows systematic generation of mass assignments based on uncertain first order logic formulas. Furthermore, by using appropriate fusion operators, higher-level applications are possible within this framework, such as inference and resolution based on uncertain data models.

2. **Combining functions for hard and soft data fusion.** We propose a framework for Bayesian estimation/tracking with soft and hard data in which the posterior is decomposed into a product of combining functions that suitably partition both the tracking state and the measurement variables, as well as their corresponding soft and hard components. This partition engenders a principled approach to estimation and prediction, and enables the integration of different subsystems when results of these subsystems are expressed through probability distributions. In addition, the proposed framework allows one to dynamically incorporate domain knowledge into inference/estimation systems. Although our technology could be used to create completely new fusion systems (potentially using all the richness of DS theory), our emphasis on combining with Bayesian frameworks is fed by our interest on allowing an easy integration with existing solutions. By following this approach, we target a faster acceptance and integration of our soft data enhanced fusion solutions.

3. **Computational efficiency improvements for hard and soft data fusion.** We introduce a computationally efficient algorithm for the processing of soft data in systems of hard and soft data fusion. To address both complexity and robustness against conflicting evidence, we aim at solving the satisfiability

problem associated with an uncertain logic model, and formulate this problem as a convex optimization problem. This significantly reduces the complexity of the problem and, by defining a cost function that minimizes the uncertainty of conflicting evidence, provides an increased level of robustness. Combined with efficient Bayesian processors for hard data processing, the proposed method can significantly improve the performance of existing methods for hard and soft data fusion.

Documentation and dissemination of the contributions in these areas have been addressed by the following publications:

1. **R. C. Núñez**, T. Wickramarathne, K. Premaratne, M. N. Murthi, S. Kübler, M. Scheutz, and M. Pravia, *Credibility assessment and inference for fusion of hard and soft information,* in 2nd International Conference on Cross- Cultural Decision Making: Focus 2012 (also in Advances in Design for Cross- Cultural Activities, 1, pp. 96-105, Eds: D. D. Schmorrow, D. M. Nicholson, CRC Press, 2013), July 2012.

2. **R. C. Núñez**, M. Scheutz, K. Premaratne, and M. N. Murthi, *Modeling Uncertainty in First-Order Logic: A Dempster-Shafer Theoretic Approach*, in 8th International Symposium on Imprecise Probability: Theories and Applications, Compiégne, France, July 2013.

3. **R. C. Núñez**, R. Dabarera, M. Scheutz, O. Bueno, K. Premaratne, and M. N. Murthi, *DS-Based Uncertain Implication Rules for Inference and Fusion Applications*, in 16th International Conference on Information Fusion, Istanbul, Turkey, July 2013.

4. **R. C. Núñez**, B. Samarakoon, K. Premaratne, and M. N. Murthi, *Hard and soft data fusion for joint tracking and classification/intent-detection*, in 16th International Conference on Information Fusion, Istanbul, Turkey, July 2013.

5. R. Dabarera, **R. C. Núñez**, K. Premaratne, and M. N. Murthi, *Dynamics of Belief Theoretic Agent Opinions Under Bounded Confidence*, in 17th International Conference on Information Fusion, Salamanca, Spain, 2014.

6. **R. C. Núñez**, M. N. Murthi, and K. Premaratne, *Efficient Computation of DS-Based Uncertain Logic Operations and its Application to Hard and Soft Data Fusion*, in 17th International Conference on Information Fusion, Salamanca, Spain, 2014.

7. T. Williams, **R. C. Núñez**, G. Briggs, M. Scheutz, K. Premaratne, and M. N. Murthi, *A Dempster-Shafer Theoretic Approach to Understanding Indirect Speech Acts*, Advances in Artificial Intelligence – IBERAMIA 2014, Santiago de Chile, Chile, 2014.

8. **R. C. Núñez**, M. N. Murthi, K. Premaratne, M. Scheutz, and O. Bueno *Uncertain Logic Processing: logic-based inference inference and reasoning using Dempster-Shafer models*, International Journal of Approximated Reasoning, Volume 95, April 2018.

1.5 Organization of this Dissertation

The reminder of this manuscript continues as follows. In Chapter 2 we introduce Uncertain Logic Processing (ULP), our core reasoning engine for soft data processing. In Chapter 3 we lay out the foundations of a DS-based graphical model that can be used for efficient reasoning under the ULP framework. In Chapter 4 we introduce a new method for hard and soft data fusion, which is based on expressing joint probability distributions as a product of combining functions. In Chapter 5 we illustrate the use and application of the proposed methods through several examples. Finally, Chapter 6 concludes the proposal and discusses avenues for future research.

CHAPTER 2

Reasoning with Soft Data using DS-based Logic Models

The ability of reasoning in the presence of uncertainty is a requirement whose importance keeps growing in science and engineering. Addressing this requirement for logic inference and reasoning systems is particularly important given the increasing number of uncertainty sources in this type of systems. Think, for example, of a logic-based Artificial Intelligence (AI) system that relies on a human-provided knowledge base for inference tasks. How certain can we be on the accuracy of the information present in this knowledge base? How can we account for lack of reliability of people's entries in the knowledge base? What about lack of information, or incomplete information in the knowledge base? Like this one, multiples scenarios arise in applications and models that involve the collection of physical data or experts' knowledge.

Due to its established position as a fundamental and very powerful tool for knowledge representation and reasoning, First Order Logic (FOL) has been gradually enriched to handle imperfections in real-life data (see Chapter 1.3.1 above). Some approaches include fuzzy logic and probabilistic logic [40]. Although useful in some applications, these approaches are sometimes limited by the way they model uncertainty, or simply by the complexity of the problem formulation. Extensions of these approaches could be strengthened by adding more flexibility in assigning probabilities (e.g., through intervals) and a more rigorous method of assigning probability/uncertainty measures (e.g., one that does not require defining priors or membership functions).

Originally introduced in [41, 5] and formally described in [42], *Uncertain Logic Processing* (ULP) is a suitable solution for addressing these issues. ULP is an extension of first order logic into Demster-Shafer (DS) theory. Consider, for example, an expression of the form: $\exists x : \ \varphi(x)$, with uncertainty $[\alpha, \beta]$, where $\varphi(x)$ is a logic predicate that depends on the variable x. The ULP framework allows us to model this sentence, and to combine it with similar ones in order to solve various inference problems. When $\alpha = \beta$, ULP renders probabilistic results. When $\alpha = \beta \in \{0, 1\}$, ULP converges to first order logic. ULP can also be used as an adaptive many-valued logic system, where the quantization varies depending on the granularity defined by the input data and/or knowledge models. Unlike existing DS models for logic that, in general, cannot guarantee logic consistency with classical logic for a plurality of logic constructs, uncertain logic preserves this consistency, and can grow to incorporate logic rules and properties without loss of uncertainty measures. By preserving this consistency, it is possible to seamlessly move between the logic and DS domains, and to incorporate both the strength of first order logic for information representation, inference, and resolution, and the strength of DS for representing and manipulating uncertainty in the data. In this chapter, we describe the fundamentals of ULP, one of the main contributions of this dissertation work.

2.1 Dempster-Shafer Theory

DS Theory is defined for a discrete set of elementary events related to a given problem. This set is called the *Frame of Discernment* (FoD). In general, a FoD is defined as $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$, and has a finite cardinality $N = |\Theta|$. Elements (or singletons) $\theta_i \in \Theta$ represent the lowest level of discernible information. The power set of Θ is defined as a set containing all the possible subsets of Θ, *i.e.*, $2^{\Theta} = \{A : A \subseteq \Theta\}$. The cardinality of the power set of Θ is 2^N. Next we introduce some basic definitions of DS Theory, as required for building uncertain logic models. For additional details on DS Theory, we refer the reader to [40, 43].

2.1.1 Basic Belief Assignment and Focal Sets

A Basic Belief Assignment (BBA) or *mass assignment* is a mapping $m_\Theta(\cdot) : 2^\Theta \to [0, 1]$ such that: $\sum_{A \subseteq \Theta} m_\Theta(A) = 1$ and $m_\Theta(\emptyset) = 0$. The BBA measures the support assigned to $A \subseteq \Theta$.

Masses in DS theory can be assigned to any singleton (e.g., $\{\theta_1\}, \{\theta_N\}$) or non-singleton (e.g., $\{\theta_1, \theta_2\}$, $\{\theta_1, \theta_3\}$, $\{\theta_1, \theta_2, \theta_3\}$) proposition.

The subsets A such that $m(A) > 0$ are referred to as focal sets of the BBA. A belief function is called Bayesian if each focal set in Θ is a singleton.

The set of focal elements is the core $\mathcal{F}_\Theta$. The triple $\{\Theta, \mathcal{F}_\Theta, m_\Theta(\cdot)\}$ is referred to as *Body of Evidence* (BoE).

The state of complete ignorance is represented by the *vacuous* BBA, which is defined as:

$$m(A) = 1_\Theta \equiv \begin{cases} 1 & \text{if } A = \Theta; \\ 0 & \text{if } A \subset \Theta. \end{cases}$$

2.1.2 Belief, Plausibility, and Uncertainty

The belief and plausibility functions are associated to a BBA m, and are often used as a convenient interpretation of belief. When focal elements are constituted of singletons only, the BBA, belief, and plausibility, all reduce to a probability assignment.

Given a BoE $\{\Theta, \mathcal{F}, m\}$, the *belief* function $\mathrm{Bel} : 2^\Theta \to [0, 1]$ is defined as:

$$\mathrm{Bel}_\Theta(A) = \sum_{B \subseteq A} m_\Theta(B).$$

$\mathrm{Bel}(A)$ represents the total belief that is committed to A without also being committed to its complement A^C.

The *plausibility* function $\mathrm{Pl} : 2^{\Theta} \to [0,1]$ is defined as: $\mathrm{Pl}_{\Theta}(A) = 1 - \mathrm{Bel}_{\Theta}(A^C)$. It corresponds to the total belief that does not contradict A.

The *uncertainty* of A is: $[\mathrm{Bel}_{\Theta}(A), \mathrm{Pl}_{\Theta}(A)]$.

2.1.3 Vacuous Extension and Marginalization

A BBA $m_{\Theta}(\cdot)$, which is defined on a FoD Θ, can be extended to a larger domain $\Theta' \supseteq \Theta$ via the *vacuous extension*:

$$m_{\Theta \uparrow \Theta'}(B) = \begin{cases} m_{\Theta}(A) & \text{if } B = A^{\Theta \uparrow \Theta'}; \\ 0 & \text{otherwise,} \end{cases}$$

where $A^{\Theta \uparrow \Theta'}$ is the projection of A into the domain Θ'.

Similarly, a BBA $m_{\Theta'}(\cdot)$, which is defined on a FoD Θ', can be projected into a coarser domain $\Theta \subseteq \Theta'$ via *marginalization*:

$$m_{\Theta' \downarrow \Theta}(A) = \sum_{B : B^{\downarrow \Theta'} = A} m_{\Theta'}(B).$$

2.1.4 Combination Rules

Information from distinct sources can be fused using combination (or fusion) rules. One of the most widely used rules is Dempster's Combination Rule (DCR). Other combination rules include the Conditional Fusion Equation (CFE) [6], Yager's rule [44], and Inagaki's unified combination rule [45], among many others. Next we present the first two as they are the more relevant for the presentation of the ULP models below. Definitions of ULP operators could be easily extend to incorporate other combination rules.

2.1.4.1 Dempster's Combination Rule (DCR)

For two focal sets $C \subseteq \Theta$ and $D \subseteq \Theta$ such that $B = C \cap D$, and two BBA's $m_j(\cdot)$ and $m_k(\cdot)$, the combined $m_{jk}(B)$ is given by:

$$m_{jk}(B) = \frac{1}{1 - K_{jk}} \sum_{C \cap D = B; B \neq \emptyset} m_j(C)\, m_k(D), \tag{2.1}$$

where $K_{jk} = \sum_{C \cap D = \emptyset} m_j(C)\, m_k(D) \neq 1$ is referred to as the *conflict* between the two BBAs; $K_{jk} = 1$ identifies two totally conflicting BBAs for which DCR-based fusion cannot be carried out.

2.1.4.2 Conditional Fusion Equation (CFE)

A combination rule that is robust when confronted with conflicting evidence is based on the *Conditional Fusion Equation (CFE)* [37]. For two identical FoDs, CFE-based fusion is defined by [6]:

$$m(B) = \sum_{i=1}^{M} \sum_{A_i \in \mathcal{A}_i} \gamma_i(A_i)\, m_i(B|A_i), \tag{2.2}$$

where $\sum_{i=1}^{M} \sum_{A_i \in \mathcal{A}_i} \gamma_i(A_i) = 1$. Here $\mathcal{A}_i = \{A \in \mathcal{F}_i : \mathrm{Bel}_i(A) > 0\},\ i = 1, \ldots, M$.

The conditional masses above are computed using Fagin-Halperns' Rule of Conditioning [46], which can be summarized as follows. Given a BoE $\{\Theta, \mathcal{F}, m\}$ and $A \in \hat{\mathcal{F}}$, with $\hat{\mathcal{F}} = \{A \subseteq \Theta : \mathrm{Bel}(A) > 0\}$, the *conditional belief* $\mathrm{Bel}(B|A) : 2^{\Theta} \to [0,1]$ and *conditional plausibility* $\mathrm{Pl}(B|A) : 2^{\Theta} \to [0,1]$ assigned to $B \subseteq \Theta$ are:

$$\mathrm{Bel}(B|A) = \frac{\mathrm{Bel}(A \cap B)}{\mathrm{Bel}(A \cap B) + \mathrm{Pl}(A \setminus B)};$$
$$\mathrm{Pl}(B|A) = \frac{\mathrm{Pl}(A \cap B)}{\mathrm{Pl}(A \cap B) + \mathrm{Bel}(A \setminus B)}.$$

2.1.5 Quantifying the Quality of Fusion Results

To help quantify the degree of degeneracy of BBAs as the fusion models above are applied, we introduce an ambiguity measure λ of a BBA; $0 \leq \lambda \leq 1$; with some important characteristics, namely: 1) $\lambda \rightarrow 0$ as the BBA degenerates (i.e., as the uncertainty grows); 2) $\lambda \rightarrow 1$ as the BBA represents a more exact model (i.e., the BBA gets close to a "perfect" model where $\alpha = \beta \in \{1, 0\}$); and 3) $\lambda \rightarrow 0$ as the values of α and $1 - \beta$ get closer to each other.

Definition 1 (Ambiguity measure) *Let $\Theta = \{x_1, x_2, \ldots, x_n\}$ be a FoD, and let m be a BBA defined on Θ. Then, an ambiguity measure λ is defined as:*

$$\lambda = 1 + \sum_{x \in \Theta} P_m(x) \log \left(P_m(x) \right), \tag{2.3}$$

where $P_m(x)$ is the probability of the event x occurring. $P_m(x)$ is obtained from a DS-to-probability transformation applied to the BBA m. ■

Note that this ambiguity measure is similar to the measure introduced in [47]. However, Definition 1 does not rely on the pignistic transformation. This gives us flexibility to select the transformation that renders an ambiguity measure that satisfies the characteristics mentioned above. From several probability transformations available (see, for example, [48], [49], [50], [51], and [52]), we have found that, unlike the others, the plausibility transformation [52] preserves these characteristics. For dichotomous mass functions (such as the true/false $\{x, \overline{x}\}$ events in ULP models), the plausibility transformation is defined as:

$$\mathrm{P\_Pl}(x) = \tfrac{1}{\eta}\mathrm{Pl}(x) \qquad \text{and} \qquad \mathrm{P\_Pl}(\overline{x}) = \tfrac{1}{\eta}\mathrm{Pl}(\overline{x}), \tag{2.4}$$

with $\eta = \mathrm{Pl}(x) + \mathrm{Pl}(\overline{x})$. Then, the ambiguity measure λ becomes:

$$\lambda = 1 + \mathrm{P\_Pl}(x) \log_2[\mathrm{P\_Pl}(x)] + \mathrm{P\_Pl}(\overline{x}) \log_2[\mathrm{P\_Pl}(\overline{x})]. \tag{2.5}$$

Figure 2.1 shows how λ changes as a function of the uncertainty parameters $[\alpha, \beta]$ that characterize a dichotomous BBA.

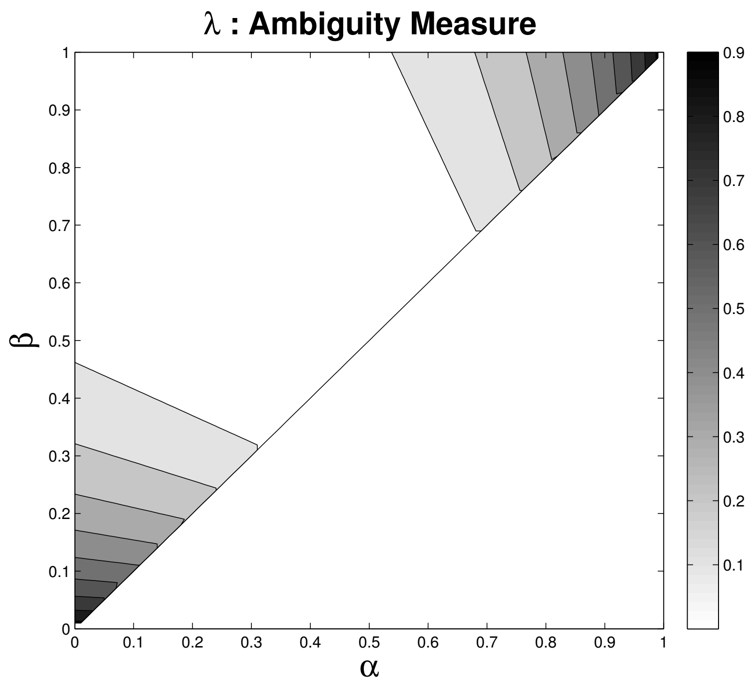


Figure 2.1: Ambiguity measure λ as a function of the uncertainty parameters $[\alpha, \beta]$ that characterize a dichotomous BBA. A higher value of λ indicates that the BBA provides stronger evidence supporting particular focal elements in the corresponding FoD.

2.2 From Propositional Logic to Uncertain Logic Processing (ULP)

2.2.1 Propositional and Predicate Logic

To introduce terminology and notation employed in our presentation of uncertain logic, let us first recall that, from propositional logic, a *proposition* is statement that could take a truth-value. Examples of propositions could be "the apple is red" or "a tall man". Propositions are typically represented by lower case greek letters (e.g., φ). A proposition can be either modified or combined with other proposition using connectives. In this manuscript we only consider the following connectives: $\wedge$ (and), $\vee$ (or), $\neg$ (not), and $\implies$ (implies).

Through *inference*, a group of propositions (called *premises*) are used to derive conclusions. Inference is typically a multi-step process. Each step in inference must be sanctioned by an acceptable rule of inference.

Predicate logic allows us to look into the structure of phrases that propositional logic treats as "black boxes" denoted by letters [53]. All sentences in predicate logic are strings of characters arranged according to rules of grammar. For example, if we want to express the relation a is above b using predicate logic, we can use `Above(a, b)`. If we want to express this relation using the corresponding propositional logic expression, we can make $\varphi =$ "`Above`" and obtain $\varphi(a, b)$ [2].

Note that, in the example above, we are allowing φ to have arguments. These arguments could be either *constants* or *variables*. In general, we assume that both propositions and predicates can have arguments, and that these arguments belong to sets that are finite.

In general, when referring to propositional and predicate logic, we follow the conventions and definitions provided in [2] and [53].

2.2.2 First-Order Logic

First Order Logic introduces, on top of propositional and predicate logic designations, the use of quantifiers. Quantified sentences provide a more flexible way of talking about all objects (i.e., elements) in our universe of discourse or asserting a property of an individual object identifying that object. There are two types of quantifiers: the *universal quantifier* ($\forall$) and the *existential quantifier* ($\exists$).

A universally quantified sentence is formed by combining the universal quantifier $\forall$, a variable x, and any simpler sentence φ, as follows: $\forall x\ \varphi(x)$. The intended meaning is that the sentence φ is true, no matter what object the variable x represents.

An existentially quantified sentence is formed by combining the existential quantifier $\exists$, a variable x, and any simpler sentence φ as follows: $\exists x\ \varphi(x)$. The intended meaning is that the sentence φ is true, for at least one object in the universe of discourse.

2.2.3 Uncertain Logic Processing

ULP is an extension of FOL that allows one to deal with expressions whose truth is uncertain. The level of uncertainty is modeled with DS theory, and is bounded in the range $[0, 1]$. In our presentation of ULP, we define the basic operators and symbols enumerated for propositional logic above (i.e., $\neg, \wedge, \vee$, and $\implies$), and incorporate the symbols and quantifiers defined for FOL in Section 2.2.2, namely $\forall$ and $\exists$.

As an extension of FOL, we must also define the set of objects that ULP deals with, i.e., its domain. Furthermore, we need means for specifying exactly which objects are referred to by constants and variables in ULP expressions. For the purpose of the description herein, these basic components are defined as follows.

Definition 2 (Domain and Interpretation of a Generic ULP Model) *Let* $D = \{d_1, d_2, \ldots, d_n\}$ *be a non-empty set of individuals. Then, we define an interpretation function* I *as a function that maps an arbitrary variable* x *into an element* $d \in D$.

Furthermore, the interpretation function I maps uncertainty intervals $[\alpha, \beta]$ in ULP expressions into properly defined DS models (i.e., into BBA or mass functions). ■

Definition 3 (Uncertain FOL Expressions) *Consider a quantifier-free first-order formula $\varphi(x)$ from a finite set of formulas Φ in some first-order language $\mathcal{L}$, with x being the only free variable in φ.*[1] *Then, an uncertain first-order logic expression relates the uncertainty associated with the truth of $\varphi(x)$ as:*

$$\varphi(x), \text{ with uncertainty } [\alpha, \beta], \tag{2.6}$$

where $[\alpha, \beta]$ refers to the corresponding uncertainty interval, $0 \leq \alpha \leq \beta \leq 1$, and x is interpreted over individuals in $D = \{d_1, \ldots, d_n\}$, with $n \geq 1$, i.e., $I(x) \in D$. The uncertain logic expression (2.6) can be abbreviated as $\varphi(x)_{[\alpha,\beta]}$. ■

The uncertainty interval $[\alpha, \beta]$ in Definition 3 indicates the support that we have for the expression $\varphi(x)$ being true or false. In addition, this interval can be used to characterize the level of ignorance that we have on the event $\varphi(x)$ being either true or false. In particular, the value of α quantifies the evidence or *belief* that we have on $\varphi(x)$ being true; β accounts for the *plausibility* of $\varphi(x)$ being true; and $\beta - \alpha$ accounts for the level of ignorance that we have on the event $\varphi(x)$ being either true or false. This concept also applies to groundings of ULP expressions, which are defined next.

Definition 4 (Grounded FOL Expressions) *Consider an uncertain first-order logic expression $\varphi(x)_{[\alpha,\beta]}$ for $I(x) \in D = \{d_1, \ldots, d_n\}$, $n \geq 1$. The grounding of this expression for an individual $d_i \in D$, $i \in \{1, \ldots, n\}$, is represented by $(\varphi_I(I(x) = d_i))_{[\alpha,\beta]}$. When no confusion can arise, we may use the alternative notations $(\varphi_I(I(x) = d_i))_{[\alpha,\beta]} \equiv (\varphi(I(x) = d_i))_{[\alpha,\beta]} \equiv (\varphi(x/d_i))_{[\alpha,\beta]} \equiv (\varphi(d_i))_{[\alpha,\beta]}$.*

[1] *We assume a single free variable for ease of description. However, extending the definition to any number of finite variables in φ is straightforward. This extension will be used later in this manuscript, as new operators are introduced.*

Note that the uncertainty of this grounding can be indicated explicitly (i.e., $\varphi(d_i)$, with uncertainty $[\alpha, \beta]$), or appended as a subindex of the grounded expression (e.g., $\varphi(d_i)_{[\alpha,\beta]}$). ■

To properly quantify the support that we have for each of the three possible events in $\varphi(d_i)$ (i.e., "true", "false", "do not know if either true/false"), we need to build a DS model for the expression $\varphi(d_i)_{[\alpha,\beta]}$. This model must be defined on a proper FoD. A proper FoD for the expression $\varphi(d_i)_{[\alpha,\beta]}$ is defined next. Its extension for the expression $\varphi(x)$ is straightforward.

Definition 5 (Basic FoD) *Given a grounded logic expression $(\varphi(d))_{[\alpha,\beta]}$, with $d \in D$, a FoD is defined as:*

$$\Theta_{\varphi(d)} = \{\varphi(d), \overline{\varphi(d)}\}, \tag{2.7}$$

where the first element (i.e., $\varphi(d)$) represents the event in which $\varphi(d)$ is true, and the second element (i.e., $\overline{\varphi(d)}$) represents the event in which $\varphi(d)$ is false. ■

An alternative definition of the basic FoD that may be appropriate for some applications appears in [41]. This alternative definition creates $\Theta_{\varphi(d)}$ as the cross product of $\Theta_{\varphi(d)}$ with a true–false set $\{\mathbf{1}, \mathbf{0}\}$, that is to say, $\Theta_{\varphi(d)} = \varphi(d) \times \{\mathbf{1}, \mathbf{0}\} = \{\varphi(d) \times \mathbf{1}, \varphi(d) \times \mathbf{0}\} = \{\varphi(d), \overline{\varphi(d)}\}$. Although not formally rigorous like the definitions in this manuscript, the alternative definition in [41] provides an intuitive way of generating FoDs and propagating uncertainties through DS-based logic inference.

When considering grounded logic expressions where more than one element of D becomes relevant, the basic FoD must be extended. An alternative is using Cartesian products of basic FoDs to obtain proper FoDs for this scenario. As an example, the comprehensive FoD, which is defined next, considers all the elements of D in a single FoD.

Definition 6 (Comprehensive FoD) *Given a set of grounded logic expressions* $\{(\varphi(d_1))_{[\alpha_1,\beta_1]}, (\varphi(d_2))_{[\alpha_2,\beta_2]}, \ldots, (\varphi(d_n)))_{[\alpha_n,\beta_n]}\}$, *we define* $\Theta_{\varphi(D)}$ *to be the Cartesian product of all basic frames of discernment* $\Theta_{\varphi(d_i)} = \{\varphi(d_i), \overline{\varphi(d_i)}\}$, $i = 1, 2, \ldots, n$, *i.e.,* $\Theta_{\varphi(D)} = \{\{\varphi(d_1), \overline{\varphi(d_1)}\} \times \ldots \times \{\varphi(d_n), \overline{\varphi(d_n)}\}\}$. ■

Based on the FoDs in Definitions 5 and 6, we can model uncertain FOL expressions in the DS theoretic framework as follows.

Definition 7 (DS Theoretic Model for Uncertain Logic Expressions) *Consider an uncertain logic expression* $\varphi(d)_{[\alpha,\beta]}$, *with* $d \in D$. *Then, a DS theoretic model that would capture the uncertain information in this logic expression is the following mass assignment:*

$$\begin{aligned} \varphi(d): \quad m(\varphi(d)) &= \alpha; \\ m(\overline{\varphi(d)}) &= 1 - \beta; \\ m(\Theta_{\varphi(d)}) &= \beta - \alpha, \end{aligned} \tag{2.8}$$

defined over the basic FoD $\Theta_{\varphi(d)} = \{\varphi(d), \overline{\varphi(d)}\}$. *When no confusion can arise, we may use the following notation, which shortens the parameter of the mass function by defining* φ *as a subindex of the mass function:*

$$\begin{aligned} \varphi(d): \quad m_\varphi(d) &= \alpha; \\ m_\varphi(\bar{d}) &= 1 - \beta; \\ m_\varphi(\Theta) &= \beta - \alpha. \end{aligned} \tag{2.9}$$

■

Note that, since the DS model for uncertain logic expressions is defined over a dichotomous FoD (i.e., the FoD has only two elements), it can be completely characterized by an uncertainty interval. Then, in the following, we may use $m(\varphi(d)) = [\alpha, \beta]$ as a shorter notation for the DS model in Definition 7.

Based on the structure of the logic propositions, as well as on their arguments, different cases arise:

1) ***Case 1* (negation)**: Recall that the basic uncertain logic expression has the form: $\varphi(d)$. Its corresponding negation is $\neg\varphi(d)$. A DS model for this negation is introduced in Section 2.3 below.

2) ***Case 2* (expressions without variables)**: We could have expressions of the form

$$\varphi(d_i), \text{with } 1 \leq i \leq n.$$

In this case, the argument of the proposition φ is a constant. A DS model for $\varphi(d_i)$ is introduced in Section 2.3 below. Also, note that this case is representative of more complex expressions whose only argument is d_i, such as

$$\begin{aligned} &\varphi_1(d_i) \wedge \varphi_2(d_i) \wedge \ldots \wedge \varphi_M(d_i), \\ \text{or} \quad &\varphi_1(d_i) \vee \varphi_2(d_i) \vee \ldots \vee \varphi_M(d_i). \end{aligned}$$

Uncertain logic models for these cases are introduced in Section 2.3.

3) ***Case 3* (expressions with one variable)**: We could have logic expressions of the form:

$$\begin{aligned} &\varphi_1(x) \wedge \varphi_2(x) \wedge \ldots \wedge \varphi_M(x), \\ \text{or} \quad &\varphi_1(x) \vee \varphi_2(x) \vee \ldots \vee \varphi_M(x). \end{aligned}$$

Modeling this type of expression in uncertain logic may require the introduction of quantifiers. This is done in Section 2.5 below.

4) ***Case 4* (expressions with more than one variable)**: We could have expressions of the form

$$\varphi_1(x_i) \wedge \varphi_2(x_j),$$

with $1 \leq i \leq k$, $1 \leq j \leq k$, and $i \neq j$. In this case, we could have uncertainty associated with the full expression $\varphi_1(x_i) \wedge \varphi_2(x_j)$, as well as with individual propositions $\varphi_1(x_i)$ and $\varphi_2(x_j)$. Uncertain logic for this case is introduced in Section 2.3. Note that this case is representative of more complex expressions such as

$$\begin{aligned} & \varphi_1(x_{i_1}) \wedge \varphi_2(x_{i_2}) \wedge \ldots \wedge \varphi_M(x_{i_M}), \\ \text{or} \quad & \varphi_1(x_{i_1}) \vee \varphi_2(x_{i_2}) \vee \ldots \vee \varphi_M(x_{i_M}), \end{aligned}$$

with $1 \leq i_j \leq k$, and $j = 1, 2, \ldots, m$. Modeling this type of expression in uncertain logic may require the introduction of quantifiers. This is done in Section 2.5 below.

5) ***Other Cases*** : Although this manuscript emphasizes descriptions of models that address Cases 1 - 4, it is easy to see that those cases could be easily extended to incorporate more complex scenarios. For example, consider the expression $\varphi_1(x_i) \wedge \varphi_2(x_i, x_j)$, or the expression $\varphi(\{d_i, d_j\})$ (non-singletons). These expressions could be modeled in uncertain logic by extending the models corresponding to Cases 3 and 4.

2.3 Basic Operators: NOT, AND, and OR

The AND and OR operators are, together with the logical negation, the basic operators in classical logic. This is also the case in ULP, as any other operator can be defined using combinations of these three basic operators. In this section we introduce the basic operators for ULP, which are defined using generic DS fusion operations. Later in this document, we show how to select appropriate fusion operators to obtain a desired behavior. For example, one may be interested in attaining consistency with classical logic, or one may want to relax this condition and ensure consistency with a paraconsistent logic. The first case (i.e., classically consistent logic) is described in Section 2.4; the latter is out of the scope of this work and is a matter of further research.

2.3.1 Uncertain Logic Negation

The negation operation in ULP is based on the definition of set complement as it applies to DS models (see [54], for example). Using this approach, we define the complement of a BBA, and then define the logical not of this BBA based on said complement. This is described next.

Definition 8 (Complementary BBA) *Consider a basic FoD* $\Theta_{\varphi(d)} = \{\varphi(d), \overline{\varphi(d)}\}$*, and a BBA* $m_\varphi(\cdot)$ *defined as:*

$$m_\varphi(d) = \alpha; \quad m_\varphi(\overline{d}) = 1 - \beta; \quad m_\varphi(\Theta) = \beta - \alpha. \tag{2.10}$$

A complementary BBA for (2.10) is given by [54]:

$$m^c_\varphi(d) = 1 - \beta;\ m^c_\varphi(\overline{d}) = \alpha;\ m^c_\varphi(\Theta) = \beta - \alpha. \tag{2.11}$$

■

Based on the complementary BBA, we can define an uncertain logic negation as follows.

Definition 9 (Logical Not in ULP) *Consider an uncertain logic expression* $\varphi(d)_{[\alpha,\beta]}$*. Also, consider its corresponding DS model, which is defined by (2.9). Then, the ULP-negation of* $\varphi(d)_{[\alpha,\beta]}$ *denoted* $\neg(\varphi(d)_{[\alpha,\beta]})$ *is defined as* $(\neg\varphi(d))_{[1-\beta,1-\alpha]}$*. We utilize the complementary BBA corresponding to (2.9) as the DS theoretic model for* $\neg\varphi(d)$*, i.e.,*

$$\begin{aligned} \neg\varphi(d): \quad m^c_\varphi(d) &= 1 - \beta; \\ m^c_\varphi(\overline{d}) &= \alpha; \\ m^c_\varphi(\Theta) &= \beta - \alpha. \end{aligned} \tag{2.12}$$

■

It is clear that the complementary BBA associated with $\varphi_{[\alpha,\beta]}(x)$ is $\neg\varphi_{[1-\beta,1-\alpha]}(x)$.

2.3.2 Uncertain Logic AND/OR

Definition 10 (Logical AND & OR in ULP) *Suppose that we have M propositions $(\varphi_i(d))_{[\alpha_i,\beta_i]}, i = \{1, 2, \ldots, M\}$. Their corresponding ULP models are $\varphi_i(d):\ m_{\varphi_i}(d) = \alpha_i;\ m_{\varphi_i}(\overline{d}) = 1 - \beta_i;\ m_{\varphi_i}(\Theta_{\varphi_i(d)}) = \beta_i - \alpha_i$, for $i = 1, 2, \ldots, M$. We propose to utilize the following DS theoretic models for the logical AND and OR of these propositions:*

$$\begin{aligned} &\bigwedge_{i=1}^{M} \varphi_i(d):\quad m(\cdot) = \bigcap_{i=1}^{M} m_{\varphi_i}(\cdot); \\ \text{and}\qquad &\bigvee_{i=1}^{M} \varphi_i(d):\quad m(\cdot) = \left(\bigcap_{i=1}^{M} m^c_{\varphi_i}(\cdot)\right)^c, \end{aligned} \tag{2.13}$$

where $\bigcap$ denotes an appropriate fusion operator. ■

In the remainder of this manuscript, we may prefer to use the notation $\bigwedge_{i=1}^{M} m_i(\cdot)$ and $\bigvee_{i=1}^{M} m_i(\cdot)$ instead of $\bigwedge_{i=1}^{M} \varphi_i(\cdot)$ and $\bigvee_{i=1}^{M} \varphi_i(\cdot)$, respectively, to emphasize that we refer to DS models in uncertain logic and not to conventional propositional logic.

Remarks:

1) The definition of the conjunction in Definition 10 explicitly includes its corresponding BBA. This BBA is obtained from the direct application of the ULP extension of the definition of conjunction, that is to say, from defining $\varphi_1(d) \vee \varphi_2(d) \vee \ldots \vee \varphi_M(d) \equiv \neg(\neg\varphi_1(d) \wedge \neg\varphi_2(d) \wedge \ldots \wedge \neg\varphi_M(d))$.

2) Based on Definition 10, we can define the AND/OR operators for unquantified expressions $\varphi_i(x)$, with uncertainty $[\alpha_i, \beta_i]$, $i = 1, \ldots, M$, as: $\bigwedge_{i=1}^{M} \varphi_i(x):\ m(\cdot) = \bigcap_{i=1}^{M} m_{\varphi_i}(\cdot)$; and $\bigvee_{i=1}^{M} \varphi_i(x) : m(\cdot) = \left(\bigcap_{i=1}^{M} m^c_{\varphi_i}(\cdot)\right)^c$, for the AND and OR operations, respectively.

3) A model similar to the one in Definition 10 can be obtained for the case of AND/OR operations of a set of expressions $\{\varphi(x_i)\}$ with uncertainty $[\alpha_i, \beta_i]$,

$x_i \in \{x_1, x_2, \ldots, x_k\}$. In this case: $\bigwedge_{i=1}^{k} \varphi(x_i)\ :\ \ m(\cdot) = \bigcap_{i=1}^{k} m_\varphi(\cdot)$, and $\bigvee_{i=1}^{k} \varphi(x_i)\ :\ \ m(\cdot) = \left(\bigcap_{i=1}^{k} m_\varphi^c(\cdot)\right)^c$. This case represents AND/OR models applied to the truthfulness of elements $\{x_i\}$ satisfying a property φ, whereas (2.13) analyzes the case of x satisfying multiple properties $\{\varphi_i\}$.

4) Definition 10 also extends to AND/OR operations of set of expressions $\{\varphi(d_i)\}$ with uncertainty $[\alpha_i, \beta_i]$, $d_i \in \{d_1, d_2, \ldots, d_n\}$. In this case: $\bigwedge_{i=1}^{n} \varphi(d_i)\ :\ \ m(\cdot) = \bigcap_{i=1}^{n} m_\varphi(\cdot)$, and $\bigvee_{i=1}^{n} \varphi(d_i)\ :\ \ m(\cdot) = \left(\bigcap_{i=1}^{n} m_\varphi^c(\cdot)\right)^c$.

2.3.3 Other Uncertain Logic Operators

It is possible to extend the ULP operators described above and create new operators. As an example, consider implication rules. An uncertain logic implication can be defined by extending the (classical) definition for the implication rule based on AND/OR operators to the uncertain logic framework. The classical logic definition is: Given two statements $\varphi_1(\cdot)$ and $\varphi_2(\cdot)$, an implication rule has the property:

$$\begin{aligned} \varphi_1(d_i) \implies \varphi_2(d_j) &\equiv \neg\varphi_1(d_i) \vee \varphi_2(d_j) \\ &\equiv \neg\left(\varphi_1(d_i) \wedge \neg\varphi_2(d_j)\right), \end{aligned}$$

where $d_i, d_j \in D$. Now we define an uncertain implication rule as follows.

Definition 11 (Uncertain Implication Rule) *Consider an antecedent $\varphi_1(d_i)$ and a consequent $\varphi_2(d_j)$, with uncertainty intervals $[\alpha_{\varphi_1(d_i)}, \beta_{\varphi_1(d_i)}]$ and $[\alpha_{\varphi_2(d_j)}, \beta_{\varphi_2(d_j)}]$, respectively. Furthermore, suppose that said uncertainty is represented via the DS theoretic models $m_1(\cdot)$ and $m_2(\cdot)$ over the FoDs $\Theta_{\varphi_1(d_i)}$ and $\Theta_{\varphi_2(d_j)}$, respectively. Then, the implication rule $\varphi_1(\cdot) \implies \varphi_2(\cdot)$ is taken to have the following DS model:*

$$m_{\varphi_1 \to \varphi_2}(\cdot) = (m_1^c \vee m_2)(\cdot) = (m_1 \wedge m_2^c)^c(\cdot), \tag{2.14}$$

over the FoD $\Theta_{\varphi_1(d_i)} \times \Theta_{\varphi_2(d_j)}$. ■

2.4 Attaining Consistency with Classical Logic

Recall that ULP operators can be tuned to satisfy particular properties. This can be accomplished by selecting an appropriate fusion operator $\bigcap$ in (2.13). Note that this operator directly impacts the behavior of logic operations, as many logic operations are derived from the fundamental AND and OR operators. This fusion operator must be then selected carefully to obtain the particular properties that we wish to obtain in an uncertainty measuring and tracking system.

One fundamental configuration that we may wish to attain for ULP is one that is consistent with classical logic. In this section we analyze the selection of the appropriate fusion operator for this case. To ensure consistency with classical logic, we are interested in a fusion operator that allows the logic operators to satisfy the following properties:

(a) **Double negation.** Given a proposition $\varphi(d)$, the BBA corresponding to its double-negation is the same model as the one associated with $\varphi(d)$. By definition of the negation in ULP (see Definition 9), ULP already satisfies $\neg\neg\varphi(d) = \varphi(d)$).

(b) **De Morgan's Laws.** $(\varphi_1(d) \vee \varphi_2(d))$ and $\neg(\neg\varphi_1(d) \wedge \neg\varphi_2(d))$ have identical DS theoretic models. Also, $(\varphi_1(d) \wedge \varphi_2(d))$ and $\neg(\neg\varphi_1(d) \vee \neg\varphi_2(d))$ have identical DS theoretic models. By definition of the conjunction and disjunction in ULP (see Definition 10), ULP already satisfies the De Morgan's laws.

(c) **Idempotency.** Uncertain logic AND and OR operators are idempotent, i.e.: $\varphi(d)_{[\alpha,\beta]} \wedge \varphi(d)_{[\alpha,\beta]} = \varphi(d)_{[\alpha,\beta]} \vee \varphi(d)_{[\alpha,\beta]} = \varphi(d)_{[\alpha,\beta]}$.

(d) **Commutativity.** Uncertain logic AND and OR operators are commutative, i.e.: $\varphi_1(d) \wedge \varphi_2(d) = \varphi_2(d) \wedge \varphi_1(d)$, and $\varphi_1(d) \vee \varphi_2(d) = \varphi_2(d) \vee \varphi_1(d)$.

(e) **Associativity.** Uncertain logic AND and OR operators must be associative. i.e.: $\varphi_1(d) \wedge [\varphi_2(d) \wedge \varphi_3(d)] = [\varphi_1(d) \wedge \varphi_2(d)] \wedge \varphi_3(d)$, and $\varphi_1(d) \vee [\varphi_2(d) \vee \varphi_3(d)] = [\varphi_1(d) \vee \varphi_2(d)] \vee \varphi_3(d)$.

(f) **Distributivity.** Uncertain logic AND and OR operators must be distributive, i.e.: $\varphi_1(d) \wedge [\varphi_2(d) \vee \varphi_3(d)] = [\varphi_1(d) \wedge \varphi_2(d)] \vee [\varphi_1(d) \wedge \varphi_3(d)]$, and $\varphi_1(d) \vee [\varphi_2(d) \wedge \varphi_3(d)] = [\varphi_1(d) \vee \varphi_2(d)] \wedge [\varphi_1(d) \vee \varphi_3(d)]$.

(g) **Consistent Boolean Models.** In the absence of uncertainty, ULP models converge to those of Boolean logic. That is to say, if $\alpha, \beta \in \{0, 1\}$, and $\alpha = \beta$, inference results in ULP render intervals that are either $[0, 0]$ or $[1, 1]$, and that correspond to the false/true truth value assignments that would be obtained using Boolean logic.

(h) **Consistent Probabilistic Models.** In probabilistic scenarios (i.e., where $\alpha = \beta$ for every uncertainty interval $[\alpha, \beta]$), ULP inference renders probabilistic models.

(i) **Uniqueness of Model.** Given propositions $\varphi_1(d)_{[\alpha_1,\beta_1]}$ and $\varphi_2(d)_{[\alpha_2,\beta_2]}$, with $\alpha_1 \neq \alpha_2$ and $\beta_1 \neq \beta_2$ the AND and OR operations satisfy $\varphi_1(d) \wedge \varphi_2(d) \neq \varphi_1(d) \vee \varphi_2(d)$.

Next we show that ULP does not satisfy all these properties when we use DRC. Then we present the CFE as a better alternative for classically consistent ULP.

2.4.1 DCR-Based Uncertain Logic

Let us consider the two-propositions (*i.e.*, $M = 2$) case. Table 2.1 contains the DCR-based logical AND and OR operations for this case. Notice that the mass assignments for the AND operation (*i.e.*, $\varphi_1(x) \wedge \varphi_2(x)$) are exactly the same as the ones obtained for the OR operation (*i.e.*, $\varphi_1(x) \vee \varphi_1(x)$). Having identical models for both AND and OR operators suggests that, although DCR may work as a fusion operator for certain operations, it does not render models that satisfy important properties for all the logical operations defined above. More particularly, DCR-based uncertain logic does not satisfy the "uniqueness of the model" property (i) mentioned earlier at the beginning of this Section. As an alternative, we propose using a more appropriate fusion strategy, such as the CFE, which is analyzed next.

Table 2.1: DCR-Based Logical AND and OR. In both cases, the masses should be normalized by $1 - K$, with $K = 1 - \sum_{A \in \mathcal{F}} m(A) = \alpha_1(1-\beta_2) + (1-\beta_1)\alpha_2$. Note that the DS models for AND and OR are identical, which suggests that DCR is not an appropriate fusion operator for classically consistent logic operations.

Focal Set	$\varphi_1(x) \wedge \varphi_2(x)$	$\varphi_1(x) \vee \varphi_2(x)$
x	$\alpha_1\beta_2 + (\beta_1 - \alpha_1)\alpha_2$	$\alpha_1\beta_2 + (\beta_1 - \alpha_1)\alpha_2$
$\overline{x}$	$(1-\beta_1)(1-\alpha_2) + (\beta_1 - \alpha_1)(1-\beta_2)$	$(1-\beta_1)(1-\alpha_2) + (\beta_1 - \alpha_1)(1-\beta_2)$
$\Theta_{\varphi_1,\varphi_2,x}$	$(\beta_1 - \alpha_1)(\beta_2 - \alpha_2)$	$(\beta_1 - \alpha_1)(\beta_2 - \alpha_2)$

2.4.2 CFE-Based Uncertain Logic

Recall (from Section 2.1.4.2) that CFE-based fusion requires the definition of a set of coefficients γ_i. We define the Logic Consistent (LC) strategy for the definition of the CFE coefficients as follows.

Definition 12 (Logic Consistent (LC) Strategy) *Consider the case $M = 2$ in (2.13), and let us define $\underline{\alpha} = \min(\alpha_1, \alpha_2)$; $\underline{\beta} = \min(\beta_1, \beta_2)$; $\overline{\alpha} = \max(\alpha_1, \alpha_2)$; $\overline{\beta} = \max(\beta_1, \beta_2)$; $\delta_1 = \beta_1 - \alpha_1$; $\delta_2 = \beta_2 - \alpha_2$; $\underline{\delta} = \underline{\beta} - \underline{\alpha}$; and $\overline{\delta} = \overline{\beta} - \overline{\alpha}$. Then select the CFE coefficients as follows:*

$$\gamma_1(d) = \gamma_2(d) \equiv \gamma(d); \qquad \gamma_1(\overline{d}) = \gamma_2(\overline{d}) \equiv \gamma(\overline{d});$$
$$\textit{and} \quad \gamma_1(\Theta) = \gamma_2(\Theta) \equiv \gamma(\Theta),$$

where $\gamma(d)$, $\gamma(\overline{d})$, and $\gamma(\Theta)$ are given by:

a. Logical AND:

$$
\begin{aligned}
&\textit{If } \delta_1+\delta_2 \neq 0:\\
&\qquad \gamma(d) = \frac{\underline{\alpha}(\beta_1+\beta_2)-\underline{\beta}(\alpha_1+\alpha_2)}{2(\delta_1+\delta_2)};\\
&\qquad \gamma(\overline{d}) = \frac{1}{2}-\frac{\underline{\beta}(2-\alpha_1-\alpha_2)-\underline{\alpha}(2-\beta_1-\beta_2)}{2(\delta_1+\delta_2)};\\
&\qquad \gamma(\Theta) = \frac{\underline{\delta}}{\delta_1+\delta_2}.\\
&\textit{If } \delta_1+\delta_2 = 0:\\
&\qquad \gamma(d) = \frac{\underline{\alpha}-\gamma(\Theta)\,(\alpha_1+\alpha_2)}{2};\\
&\qquad \gamma(\overline{d}) = \frac{(1-\underline{\alpha})-\gamma(\Theta)\,(2-\alpha_1-\alpha_2)}{2};\\
&\qquad \gamma(\Theta) \quad \textit{is arbitrary in the interval } [0,1].
\end{aligned}
$$

b. Logical OR:

$$
\begin{aligned}
&\textit{If } \delta_1+\delta_2 \neq 0:\\
&\qquad \gamma(d) = \frac{1}{2}-\frac{\overline{\beta}(2-\alpha_1-\alpha_2)-\overline{\alpha}(2-\beta_1-\beta_2)}{2(\delta_1+\delta_2)};\\
&\qquad \gamma(\overline{d}) = \frac{\overline{\alpha}(\beta_1+\beta_2)-\overline{\beta}(\alpha_1+\alpha_2)}{2(\delta_1+\delta_2)};\\
&\qquad \gamma(\Theta) = \frac{\overline{\delta}}{\delta_1+\delta_2}.\\
&\textit{If } \delta_1+\delta_2 = 0:\\
&\qquad \gamma(d) = \frac{\overline{\alpha}-\gamma(\Theta)\,(\alpha_1+\alpha_2)}{2};\\
&\qquad \gamma(\overline{d}) = \frac{(1-\overline{\alpha})-\gamma(\Theta)\,(2-\alpha_1-\alpha_2)}{2};\\
&\qquad \gamma(\Theta) \quad \textit{is arbitrary in the interval } [0,1].
\end{aligned}
$$

■

Based on the LC Strategy, we can define the CFE-based AND/OR operators, as well as implication rules, as follows.

2.4.2.1 CFE-Based AND and OR Operators

Given two uncertain propositions $\varphi_1(d)_{[\alpha_1,\beta_1]}$ and $\varphi_2(d)_{[\alpha_2,\beta_2]}$, the LC strategy renders the following BBA for the AND operation (see proof in Appendix A):

$$\begin{aligned}\varphi_1(d) \wedge \varphi_2(d): \quad & m(d) &&= \underline{\alpha}; \\ & m(\overline{d}) &&= 1 - \underline{\beta};\ \text{and} \\ & m(\Theta_{(\varphi_1 \wedge \varphi_2)(d)}) &&= \underline{\beta} - \underline{\alpha},\end{aligned} \tag{2.15}$$

with $\underline{\alpha} = \min(\alpha_1, \alpha_2)$ and $\underline{\beta} = \min(\beta_1, \beta_2)$. Similarly, when used for the OR operation, the LC strategy renders the following BBA:

$$\begin{aligned}\varphi_1(d) \vee \varphi_2(d): \quad & m(d) &&= \overline{\alpha}; \\ & m(\overline{d}) &&= 1 - \overline{\beta};\ \text{and} \\ & m(\Theta_{(\varphi_1 \vee \varphi_2)(d)}) &&= \overline{\beta} - \overline{\alpha},\end{aligned} \tag{2.16}$$

with $\overline{\alpha} = \max(\alpha_1, \alpha_2)$ and $\overline{\beta} = \max(\beta_1, \beta_2)$.

It can be proven that CFE-based fusion using the LC strategy satisfies the properties (a)-(g) above. As mentioned above, properties (a) and (b) are satisfied by the basic definitions of ULP models. Proofs for properties (c)-(f) and (h) can be found in [41, 42], and are presented in Appendix B. Property (h) can be easily proven as follows: Consider uncertainty parameters defined as $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$. Let us denote $\varphi_1(d)_{[\alpha_1,\beta_1]}$ and $\varphi_2(d)_{[\alpha_2,\beta_2]}$ as $\varphi(d)_{[\alpha_1]}$ and $\varphi(d)_{[\alpha_2]}$, respectively. We then get

$$\begin{aligned}\varphi(d)_{[\alpha_1]} \wedge \varphi(d)_{[\alpha_2]} &= \varphi(d)_{[\underline{\alpha}]}; \\ \varphi(d)_{[\alpha_1]} \vee \varphi(d)_{[\alpha_2]} &= \varphi(d)_{[\overline{\alpha}]},\end{aligned} \tag{2.17}$$

where $\underline{\alpha} = \min(\alpha_1, \alpha_2)$ and $\overline{\alpha} = \max(\alpha_1, \alpha_2)$. Property (g) can be easily proved by building a truth table based on the models for (h). Finally, property (i) is satisfied by the models (2.15) and (2.16) above. ULP models for multiple propositions can be found in Section 2.4.2.3.

2.4.2.2 CFE-Based Implication

In the case of logic implications, the model in Definition 11 renders the following BBA:

$$\begin{aligned}
&\varphi_1(d_i) \implies \varphi_2(d_j): \\
&\qquad m_{\varphi_1\to\varphi_2}(d_i \times \Theta_{\varphi_2,d_j}) &&= \tfrac{1}{2}\alpha_R; \\
&\qquad m_{\varphi_1\to\varphi_2}(\Theta_{\varphi_1,d_i} \times d_j) &&= \tfrac{1}{2}\alpha_R; \\
&\qquad m_{\varphi_1\to\varphi_2}(\overline{d_i} \times \Theta_{\varphi_2,d_j}) &&= \tfrac{1}{2}(1-\beta_R); \\
&\qquad m_{\varphi_1\to\varphi_2}(\Theta_{\varphi_1,d_i} \times \overline{d_j}) &&= \tfrac{1}{2}(1-\beta_R); \\
&\qquad m_{\varphi_1\to\varphi_2}(\Theta_{\varphi_1,d_i} \times \Theta_{\varphi_2,d_j}) &&= \beta_R - \alpha_R,
\end{aligned} \tag{2.18}$$

with $\alpha_R = \max(1-\beta_1, \alpha_2)$ and $\beta_R = \max(1-\alpha_1, \beta_2)$. Note that α_R and β_R define the uncertainty interval $[\alpha_R, \beta_R]$ of the implication rule. This interval is obtained from projecting the BBA defined in (2.18) into the true-false components of the original BoEs, which we label as $\{\mathbf{1}, \mathbf{0}\}$, for ease of notation, as follows:

$$\begin{aligned}
m_{\varphi_1\to\varphi_2}(\mathbf{1}) &= m_{\varphi_1\to\varphi_2}(d_i \times \Theta_{\varphi_2,d_j}) \\
&\quad + m_{\varphi_1\to\varphi_2}(\Theta_{\varphi_1,d_i} \times d_j) \quad = \alpha_R; \\
m_{\varphi_1\to\varphi_2}(\mathbf{0}) &= m_{\varphi_1\to\varphi_2}(\overline{d_i} \times \Theta_{\varphi_2,d_j}) \\
&\quad + m_{\varphi_1\to\varphi_2}(\Theta_{\varphi_1,d_i} \times \overline{d_j}) \quad = 1-\beta_R; \\
m_{\varphi_1\to\varphi_2}(\{\mathbf{1},\mathbf{0}\}) &= m_{\varphi_1\to\varphi_2}(\Theta_{\varphi_1,d_i} \times \Theta_{\varphi_2,d_j}) = \beta_R - \alpha_R.
\end{aligned} \tag{2.19}$$

The $\mathbf{1}$ component corresponds to the event "$\varphi_1 \to \varphi_2$", and the $\mathbf{0}$ component corresponds to the event "$\overline{\varphi_1 \to \varphi_2}$". Note that, in the Boolean case (i.e., $\alpha_1 = \beta_1 \in \{\mathbf{1}, \mathbf{0}\}$ and $\alpha_2 = \beta_2 \in \{\mathbf{1}, \mathbf{0}\}$), the CFE-based uncertain implication rule converges to the conventional logic result (see Table 2.2).

Table 2.2: CFE-Based implication with Boolean arguments. Uncertainty parameters are defined so that they represent complete certainty on the truth (or falsity) of each proposition. In this case, there is complete certainty of the truth of the output model (as occurs with classical logic.)

Parameters		**Uncertainty of the rule**
$[\alpha_1, \beta_1]$	$[\alpha_2, \beta_2]$	$[\alpha_R, \beta_R]$
$[0, 0]$	$[0, 0]$	$[1, 1]$
$[0, 0]$	$[1, 1]$	$[1, 1]$
$[1, 1]$	$[0, 0]$	$[0, 0]$
$[1, 1]$	$[1, 1]$	$[1, 1]$

2.4.2.3 Uncertain Logic Operators for Multiple Propositions

The CFE-based uncertain logic AND operator generalizes as follows for the general case of n propositions:

$$\begin{aligned}
&\bigwedge_{i=1}^{n} \varphi(d_i) : \\
&m(\ \theta_{i^-} \times d_i \times \theta_{i^+}\ ) = \tfrac{1}{n}\underline{\alpha}_{1:n} \\
&\qquad + \tfrac{1}{n}\left(\underline{\beta}_{1:n} - \underline{\alpha}_{1:n}\right)\alpha_i, \forall i; \\
&m(\ \theta_{i^-} \times \overline{d}_i \times \theta_{i^+}\ ) = \tfrac{1}{n}\left(1 - \underline{\beta}_{1:n}\right) \\
&\qquad + \tfrac{1}{n}\left(\underline{\beta}_{1:n} - \underline{\alpha}_{1:n}\right)(1 - \beta_i), \forall i; \\
&m(\ \Theta_{\varphi,D}\ ) = \tfrac{1}{n}\left(\underline{\beta}_{1:n} - \underline{\alpha}_{1:n}\right) \\
&\qquad \left(\textstyle\sum_{i=1}^{n}(\beta_i - \alpha_i)\right),
\end{aligned} \tag{2.20}$$

with:

$$\theta_{i^-} = \Theta_{\varphi,d_1} \times \Theta_{\varphi,d_2} \times \ldots \times \Theta_{\varphi,d_{i-1}};$$

$$\theta_{i^+} = \Theta_{\varphi,d_{i+1}} \times \ldots \times \Theta_{\varphi,d_{n-1}} \times \Theta_{\varphi,d_n},$$

and

$$\underline{\alpha}_{1:n} = \underline{\alpha}_{1,2,\ldots,n} = \min(\alpha_1, \alpha_2, \ldots, \alpha_N);$$

$$\underline{\beta}_{1:n} = \underline{\beta}_{1,2,\ldots,n} = \min(\beta_1, \beta_2, \ldots, \beta_N).$$

Similarly, the logical OR operator for n propositions generalizes as follows:

$$\begin{aligned}
&\bigvee_{i=1}^{n} \varphi(d_i) : \\
&\quad m(\ \theta_{i^-} \times d_i \times \theta_{i^+}\ ) \ = \ \tfrac{1}{n}\overline{\alpha}_{1:n} \\
&\qquad\qquad\qquad\qquad + \tfrac{1}{n}\left(\overline{\beta}_{1:n} - \overline{\alpha}_{1:n}\right)\alpha_i, \forall i; \\
&\quad m(\ \theta_{i^-} \times \overline{d}_i \times \theta_{i^+}\ ) \ = \ \tfrac{1}{n}\left(1 - \overline{\beta}_{1:n}\right) \\
&\qquad\qquad\qquad\qquad + \tfrac{1}{n}\left(\overline{\beta}_{1:n} - \overline{\alpha}_{1:n}\right)(1-\beta_i), \forall i; \\
&\qquad\qquad m(\ \Theta_{\varphi,D}\ ) = \ \tfrac{1}{n}\left(\overline{\beta}_{1:n} - \overline{\alpha}_{1:n}\right) \\
&\qquad\qquad\qquad\qquad \left(\textstyle\sum_{i=1}^{n}(\beta_i - \alpha_i)\right),
\end{aligned} \tag{2.21}$$

with:

$$\overline{\alpha}_{1:n} = \overline{\alpha}_{1,2,\ldots,n} = \max(\alpha_1, \alpha_2, \ldots, \alpha_n);$$
$$\overline{\beta}_{1:n} = \overline{\beta}_{1,2,\ldots,n} = \max(\beta_1, \beta_2, \ldots, \beta_n).$$

Simpler models are obtained for the expressions $\bigwedge_{i=1}^{M} \varphi_i(x)$ and $\bigvee_{i=1}^{M} \varphi_i(x)$:

$$\begin{aligned}
&\bigwedge_{i=1}^{M} \varphi_i(x) : \\
&\qquad m(x) = \underline{\alpha}_{1:M} + \tfrac{1}{M}\left(\underline{\beta}_{1:M} - \underline{\alpha}_{1:M}\right)\left(\textstyle\sum_{i=1}^{M}\alpha_i\right); \\
&\qquad m(\overline{x}) = \left(1 - \underline{\beta}_{1:M}\right) \\
&\qquad\qquad + \tfrac{1}{M}\left(\underline{\beta}_{1:M} - \underline{\alpha}_{1:M}\right)\left(\textstyle\sum_{i=1}^{M}(1-\beta_i)\right); \\
&\quad m(\Theta_{\varphi_{1:M},x}) = \tfrac{1}{M}\left(\underline{\beta}_{1:M} - \underline{\alpha}_{1:M}\right)\left(\textstyle\sum_{i=1}^{M}(\beta_i - \alpha_i)\right),
\end{aligned} \tag{2.22}$$

and

$$
\begin{aligned}
\bigvee_{i=1}^{M} \varphi_i(x) : & \\
m(x) &= \overline{\alpha}_{1:M} + \tfrac{1}{M}\left(\overline{\beta}_{1:M} - \overline{\alpha}_{1:M}\right)\left(\textstyle\sum_{i=1}^{M} \alpha_i\right); \\
m(\overline{x}) &= \left(1 - \overline{\beta}_{1:M}\right) \\
&\quad + \tfrac{1}{M}\left(\overline{\beta}_{1:M} - \overline{\alpha}_{1:M}\right)\left(\textstyle\sum_{i=1}^{M}(1 - \beta_i)\right); \\
m(\Theta_{\varphi_{1:M},x}) &= \tfrac{1}{M}\left(\overline{\beta}_{1:M} - \overline{\alpha}_{1:M}\right)\left(\textstyle\sum_{i=1}^{M}(\beta_i - \alpha_i)\right).
\end{aligned} \tag{2.23}
$$

2.4.3 ULP as a Model for Capturing Variable Granularities of Many-Valued Logics

ULP naturally adapts to different quantizations in the uncertainty of the evidence. For example, if all the uncertainty intervals $[\alpha, \beta]$ in the premises are such that $\alpha, \beta \in \{0, 1\}$, then the uncertainty results provided by ULP will be also in $\{0, 1\}$. In general, if uncertainties of the premises are quantized in steps of $1/n$, with $n = 2, 3, \ldots$, then the result will also be quantized by this step size.

To illustrate this property, consider an n-ary logic in which we have the propositions φ_i and φ_j, with $i, j \in \{0, 1, \ldots, n\}$ and $n \geq 1$, such that the uncertainty of the premises is probabilistic and modeled as follows:

$$
\varphi_i, \text{ with uncertainty } \frac{i}{n}; \text{ and } \varphi_j, \text{ with uncertainty } \frac{j}{n}. \tag{2.24}
$$

Now consider an uncertain implication rule $\varphi_i \implies \varphi_j$. Given (2.24), the uncertainty of this implication is given by:

$$
\varphi_i \implies \varphi_j, \text{ with uncertainty } \max(1 - \frac{i}{n}, \frac{j}{n}). \tag{2.25}
$$

Note that the resulting uncertainty in the implication rule is a multiple of $1/n$. It can be shown that the same behavior occurs for NOT, AND, and OR operations. Hence, the ULP framework can be used to model an n-ary (i.e., many-valued) logic, where we can have from coarse ($n = 2$, Boolean logic) to infinitely small quantizations (when

$n \to \infty$). Moreover, we can increase the granularity dynamically during an inference process, right at the moment when new evidence introduces a refined granularity. That is to say, ULP can be used to adaptively adjust the quantization of the uncertain models according to the input data.

2.5 Quantifiers in Uncertain Logic

Existential and universal quantifiers extend the propositional logic framework described above into a First-Order Logic (FOL) environment. For ULP operations, these quantifiers are defined as described next.

2.5.1 Existential Quantifiers

Existential quantifiers are used to express that a sentence is true for at least one object of the universe of discourse. Given a logic predicate $\varphi(x)$, where x can take any value in $D = \{d_1, d_2, \ldots\}$, a corresponding existentially quantified sentence is defined as: $\exists x \;\; \varphi(x)$. We define existential quantifiers in ULP as follows.

Definition 13 (Existential Quantifier in ULP) *Consider the statement:*

$$(\exists x \;\; \varphi(x))_{[\alpha,\beta]}, \tag{2.26}$$

where $[\alpha, \beta]$ *refers to the corresponding uncertainty interval,* $0 \leq \alpha \leq \beta \leq 1$, *and* x *is interpreted over elements in* $D = \{d_1, d_2, \ldots, d_n\}$, *with* $n \geq 1$. *Let us define an FoD over the domain* D *as* $\Theta_{\varphi,D} = \Theta_{\varphi,d_1} \times \Theta_{\varphi,d_2} \times \ldots \times \Theta_{\varphi,d_n}$. *Then, we define the DS theoretic model for (2.26) as:*

$$\bigvee_{i=1}^{n} (\varphi(d_i))_{[\alpha_i,\beta_i]}, \tag{2.27}$$

over the FoD $\Theta_{\varphi,D}$, subject to the constraint:

$$m(\mathbf{1}) = \sum_{i=1}^{n} m_\varphi(d_i) = \alpha;$$
$$m(\mathbf{0}) = \sum_{i=1}^{n} m_\varphi(\overline{d_i}) = 1 - \beta;$$
$$m(\Theta_{\varphi,\Theta_D}) = \beta - \alpha, \tag{2.28}$$

where we have adopted the notation d_i and $\overline{d_i}$ to represent the sets $\{\ldots \times \Theta_{\varphi,d_{i-2}} \times \Theta_{\varphi,d_{i-1}} \times \varphi(d_i) \times \Theta_{\varphi,d_{i+1}} \times \Theta_{\varphi,d_{i+2}} \times \ldots\}$, *and* $\{\ldots \times \Theta_{\varphi,d_{i-2}} \times \Theta_{\varphi,d_{i-1}} \times \overline{\varphi(d_i)} \times \Theta_{\varphi,d_{i+1}} \times \Theta_{\varphi,d_{i+2}} \times \ldots\}$, *respectively.* ■

Remarks:

1) Definition 13 is based on the substitutional conception of quantifiers, according to which $\exists$ is treated as a generalization of the disjunction [24] grounded over the finite set $D = \{d_1, d_2, \ldots, d_n\}$.

2) When using the CFE-based ULP model described above, the constraint (2.28) is satisfied if the uncertainty of at least one of the propositions $\varphi(d_i)$ in (2.27) is $[\alpha, \beta]$, and the uncertainty of every other proposition is $[0, 0]$ (or, in general, $[\alpha_j, \beta_j]$, with $\alpha_j \leq \alpha$, $\beta_j \leq \beta$, and $i \neq j$), then the DS model corresponding to (2.27) is equivalent to the DS model corresponding to (2.26) when the OR operations are computed as indicated by Definitions 10 and 12.

2.5.2 Universal Quantifiers

Universal quantifiers are used to express that a sentence is true for every object of the universe of discourse. Given a logic predicate $\varphi(x)$, where x can take any value in $D = \{d_1, d_2, \ldots\}$, a corresponding universally quantified sentence is defined as: $\forall x \;\; \varphi(x)$. We define universal quantifiers in ULP as follows.

Definition 14 (Universal Quantifier in ULP) *Consider the statement:*

$$(\forall x \quad \varphi(x))_{[\alpha,\beta]}, \tag{2.29}$$

where $[\alpha, \beta]$ refers to the corresponding uncertainty interval, $0 \leq \alpha \leq \beta \leq 1$, and x is interpreted over elements in $D = \{d_1, d_2, \ldots, d_n\}$, with $n \geq 1$. Let us define an FoD over the domain D as $\Theta_{\varphi,D} = \Theta_{\varphi,d_1} \times \Theta_{\varphi,d_2} \times \ldots \times \Theta_{\varphi,d_n}$. Then, we define the DS theoretic model for (2.29) as:

$$\bigwedge_{i=1}^{n} (\varphi(d_i))_{[\alpha_i,\beta_i]}, \tag{2.30}$$

over the FoD $\Theta_{\varphi,D}$, subject to the constraint:

$$\begin{aligned} m(\mathbf{1}) &= \sum\nolimits_{i=1}^{n} m_{\varphi}(d_i) = \alpha; \\ m(\mathbf{0}) &= \sum\nolimits_{i=1}^{n} m_{\varphi}(\overline{d_i}) = 1 - \beta; \\ m(\Theta_{\Theta_{\varphi,\Theta_D}}) &= \beta - \alpha. \end{aligned} \tag{2.31}$$

where we have adopted the notation d_i and $\overline{d_i}$ to represent the sets $\{\ldots \times \Theta_{\varphi,d_{i-2}} \times \Theta_{\varphi,d_{i-1}} \times \varphi(d_i) \times \Theta_{\varphi,d_{i+1}} \times \Theta_{\varphi,d_{i+2}} \times \ldots\}$, and $\{\ldots \times \Theta_{\varphi,d_{i-2}} \times \Theta_{\varphi,d_{i-1}} \times \overline{\varphi(d_i)} \times \Theta_{\varphi,d_{i+1}} \times \Theta_{\varphi,d_{i+2}} \times \ldots\}$, respectively. ■

Remarks:

1) Definition 14 is based on the substitutional conception of quantifiers, according to which $\forall$ is treated as a generalization of the conjunction [24] grounded over the finite set $D = \{d_1, d_2, \ldots, d_n\}$. This definition is justified as long as the conjunction satisfies both associativity and commutativity, as is the case of the CFE-based ULP disjunction described above.

2) When using the CFE-based ULP model described above, the constraint (2.31) is satisfied if the uncertainty of every proposition $\varphi(d_i)$ in (2.30) is $[\alpha, \beta]$.

3) Although an infinite number of solutions satisfy (2.31), a useful solution (e.g., for universal instantiation on inference) is given by $m_\varphi(d_i) = \alpha$; $m_\varphi(\overline{d_i}) = 1 - \beta$; and $m_\varphi(\{d_i, \overline{d_i}\}) = \beta - \alpha, i = 1, 2, \ldots, n$. This solution can be proven by applying idempotency to the AND operator.

2.6 Semantics: Encapsulating uncertain information in FOL

In classical logic there are two truth-values, "true" and "false". An expression that is true for all interpretations is called a tautology ("$\top$"). An expression that is not true for any interpretation is a contradiction ("$\perp$"). Two expressions are semantically equivalent if they take on the same truth value for all interpretations.

In ULP we extend these definitions. The truth value of an expression corresponds to the support that is projected into the true-false FoD, $\Theta_{t-f} = \{\mathbf{1}, \mathbf{0}\}$. A BBA (2.8) defined by $[\alpha, \beta] = [1, 1]$ corresponds to the classical logical truth. A BBA (2.8) defined by $[\alpha, \beta] = [0, 0]$ corresponds to the classical logical falsehood.

The notions of tautology and contradiction in ULP are extended following an approach similar to that in [55]. In particular, given a generic dichotomous BBA ψ characterized by the uncertainty interval $\sigma = [\alpha, \beta]$, we define a σ-tautology as $\top_\sigma \equiv \psi \vee \neg\psi$, and a σ-contradiction as $\perp_\sigma \equiv \psi \wedge \neg\psi$. It follows that $\top \equiv \top_{\sigma=[1,1]}$, and $\perp \equiv \perp_{\sigma=[0,0]}$.

2.7 Inference in ULP

When using the LC strategy for CFE-based ULP, inference in ULP shares the fundamental principles of classical logic. Inheriting inference rules and algorithms from classical logic is possible due to the definition of ULP as an algebra that reproduces the core properties of classical logic. Due to the extensive number of methods for logic inference, the scope of this section is limited to the introduction of some of the most fundamental inference rules, along with some basic examples that illustrate ULP inference. For an extended definition of these rules and their application for inference in the context of classical logic, we refer the reader to [2].

Inference and resolution using ULP can be done by simply combining a set of uncertain logic predicates, and extracting information from a consolidated BoE. The set of uncertain logic predicates to be combined should be enough to describe the information that should be used for inference. The process for selecting the required logic predicates could be similar to typical approaches in logic, such as resolution trees.

2.7.1 Inference Rules in ULP

2.7.1.1 Modus Ponens (MP)

Modus Ponens (MP) rule states that, whenever the sentences $\varphi \implies \psi$ and φ have been established, then we can infer the sentence ψ as well. MP extends to ULP as follows. Consider:

$$\begin{aligned} \varphi_1(d_1)&, \text{ with uncertainty } [\alpha_1, \beta_1]; \\ \varphi_2(d_2)&, \text{ with uncertainty } [\alpha_2, \beta_2]; \text{ and} \\ \varphi_1(d_1) \implies \varphi_2(d_2)&, \text{ with uncertainty } [\alpha_R, \beta_R]. \end{aligned} \tag{2.32}$$

Then, given the uncertain premises $\varphi_1(d_1) \implies \varphi_2(d_2)$ and $\varphi_1(d_1)$, MP allows us to infer the uncertain expression $\varphi_2(d_2)$. Note that if the uncertainty parameters $[\alpha_2, \beta_2]$ are unknown, their value should be obtained by using the definition of uncertain implication rules in Section 2.4.2.2 above. Based on (2.18), if we know a model $m_{\varphi_1 \to \varphi_2}$ for the implication rule, as well as a model m_{φ_1} for the antecedent, we could obtain a model for an unknown consequent $m_2(\cdot)$. In this case:

$$\alpha_2 = \begin{cases} \alpha_R, & \text{if } \alpha_R > 1 - \beta_1; \\ [0, \alpha_R], & \text{if } \alpha_R = 1 - \beta_1; \text{ and} \\ \text{no solution}, & \text{otherwise}, \end{cases} \tag{2.33}$$

and

$$\beta_2 = \begin{cases} \beta_R, & \text{if } \beta_R > 1 - \alpha_1; \\ [0, \beta_R], & \text{if } \beta_R = 1 - \alpha_1; \text{ and} \\ \text{no solution}, & \text{otherwise}. \end{cases} \tag{2.34}$$

Some important observations:

- Given a pair of uncertainty intervals $[\alpha_1, \beta_1]$ and $[\alpha_R, \beta_R]$, it is not always possible to infer anything about a model for the consequent (i.e., $m_2(\cdot)$). This is consistent with the application of the Modus Ponens (MP) rule, according to which, if we know $\varphi_1 \implies \varphi_2$, and also that φ_1 is true, then we can infer that φ_2 is true. However, if we know that φ_1 is false (i.e., $\neg\varphi_1$ is true), we cannot say anything about the truth value of φ_2.

- Furthermore, once we have enough support in $m_1(\cdot)$ to infer something regarding $m_2(\cdot)$, the only conclusion that we can provide regarding the uncertainty of $m_2(\cdot)$ is that said uncertainty is $m_R(\cdot)$. That is, after we have gathered a certain amount of evidence regarding the truth of the antecedent, getting more evidence is not going to affect the confidence we have on the truth of the consequent (i.e., α_2 is bounded by α_R, and β_2 is bounded by β_R).

- When $\alpha_R = 1 - \beta_1$ and $\beta_R = 1 - \alpha_1$, an infinite number of solutions exist for $[\alpha_2, \beta_2]$. We could use the minimum commitment criterion to decide $\alpha_2 = 0$ and $\beta_2 = \beta_R$.

To better understand MP in ULP, consider an example where $\alpha_1 = \beta_1 = \alpha_2 = \beta_2 = 1$. In this case, we obtain $\alpha_R = \beta_R = 1$. Furthermore, given the $\varphi_1 \implies \varphi_2$ and

φ_1, then we can infer φ_2 with uncertainty $[\alpha_2 = \beta_2] = [1, 1]$. This case represents a scenario with no uncertainty.

Now consider a scenario where there is uncertainty in the rule, in such a way that $[\alpha_R, \beta_R] = [0.5, 1.0]$, and assume that we have a model for the uncertainty of φ_1 such that $\alpha_1 = \beta_1 = 1$. Then, MP allows us to infer φ_2, with the uncertainty $[\alpha_2, \beta_2]$ obtained from the equations $\alpha_R = \max(1-\beta_1, \alpha_2)$ and $\beta_R = \max(1-\alpha_1, \beta_2)$. Solving these equations we obtain $\alpha_2 = 0.5$ and $\beta_2 = 1$.

2.7.1.2 Modus Tolens (MT)

This rule states that, if we know that $\varphi \implies \psi$, then we can infer $\neg\varphi$ if we believe that ψ is false. MT extends to uncertain logic as follows. Assume that the uncertainty on each of the expressions involved in MP are defined by (2.32). Then, given the uncertain premises $\varphi_1(x) \implies \varphi_2(y)$ and $\neg\varphi_2$, MT allows us to infer the uncertain expression $\neg\varphi_1(y)$. As with MP above, if the uncertainty parameters $[\alpha_2, \beta_2]$ are unknown, their value should be obtained by applying the methodology introduced in Section 2.3.

2.7.1.3 Other Rules of Inference

Uncertain logic can be extended by incorporating new rules of inference that already exist in conventional logic inference. Some examples of new rules of inference are: AND elimination (AE), AND introduction (AI), universal instantiation (UI), and existential instantiation (EI). The definition of these rules of inference is straightforward based on their definition for conventional logic, and is not included in this manuscript. For a description of these rules of inference, we refer the reader to [2].

2.7.2 Examples

In this Section we illustrate the use and application of the DS-based uncertain logic framework described above. Additional examples with applications on human-robot interaction, tracking, and intent-detection can be found in [5] and [56], and in Chapter 5 below.

2.7.2.1 Example 1

Consider the following problem, originally introduced in [2]. We know that horses are faster than dogs and that there is a greyhound that is faster than every rabbit. We know that Harry is a horse and that Ralph is a rabbit. We also know that greyhounds are dogs and that our speed relationship is transitive. Then:

$$\forall x\ \forall y\ \ \text{Horse}(x) \wedge \text{Dog}(y) \Rightarrow \text{Faster}(x,y) \tag{2.35a}$$

$$\exists y\ \ \text{Greyhound}(y) \wedge (\forall z\ \ \text{Rabbit}(z) \Rightarrow \text{Faster}(y,z)) \tag{2.35b}$$

$$\forall y\ \ \text{Greyhound}(y) \Rightarrow \text{Dog}(y) \tag{2.35c}$$

$$\forall x\ \forall y\ \forall z\ \ \text{Faster}(x,y) \wedge \text{Faster}(y,z) \Rightarrow \text{Faster}(x,z) \tag{2.35d}$$

$$\text{Horse}(\text{Harry}) \tag{2.35e}$$

$$\text{Rabbit}(\text{Ralph}). \tag{2.35f}$$

Using these logic statements, it can be inferred that Harry is faster than Ralph (i.e., Faster(Harry, Ralph)) [2].

Now, let us introduce uncertain logic operations by assuming that the logic premise (2.35a) is uncertain, with uncertainty $[\alpha_1, \beta_1]$, and that there is no uncertainty in premises (2.35b)-(2.35f). This represents some uncertainty in the sentence "horses are faster than dogs", which may occur if we consider cases such as sick or old horses compared to healthy dogs. The steps that are used for inferring Faster(Harry, Ralph), as well as the uncertainty in each of the steps of this process are in Table 2.3. It is easy to verify that, if $\alpha_1 = \beta_1 = 1$.

The initial steps in the inference process are simply the reproduction of (2.35a)-(2.35f) as premises 1 to 6. Steps 7 to 13 can be obtained from applying EI, AI, UI, and

Table 2.3: Steps followed for the inference of the sentence Faster(Harry, Ralph) based on the premises defined in (2.35). The uncertainty is obtained from applying uncertain logic definitions and rules to the example described in Section 2.7.2.1.

	Logic Formula	**Premises & Rule**	**Uncertainty**
1	$\forall x\ \forall y$ Horse$(x) \wedge$ Dog$(y) \Rightarrow$ Faster(x, y)	Δ	$[\alpha_1, \beta_1]$
2	$\exists y$ Greyhound$(y) \wedge (\forall z$ Rabbit$(z) \Rightarrow$ Faster$(y, z))$	Δ	$[\alpha_2, \beta_2] = [1, 1]$
3	$\forall y$ Greyhound$(y) \Rightarrow$ Dog(y)	Δ	$[\alpha_3, \beta_3] = [1, 1]$
4	$\forall x\ \forall y\ \forall z$ Faster$(x, y) \wedge$ Faster$(y, z) \Rightarrow$ Faster(x, z)	Δ	$[\alpha_4, \beta_4]$
5	Horse(Harry)	Δ	$[\alpha_5, \beta_5]$
6	Rabbit(Ralph)	Δ	$[\alpha_6, \beta_6] = [1, 1]$
7	Greyhound(Greg) $\wedge (\forall z$ Rabbit$(z) \Longrightarrow$ Faster(Greg, z))	2, EI	$[\alpha_7, \beta_7] = [1, 1]$
8	Greyhound(Greg)	7, AE	$[\alpha_8, \beta_8] = [1, 1]$
9	$\forall z$ Rabbit$(z) \Longrightarrow$ Faster(Greg, z)	7, AE	$[\alpha_9, \beta_9] = [1, 1]$
10	Rabbit(Ralph) $\Longrightarrow$ Faster(Greg, Ralph)	9, UI	$[\alpha_{10}, \beta_{10}] = [1, 1]$
11	Faster(Greg, Ralph)	10, 6, MP	$[\alpha_{11}, \beta_{11}] = [1, 1]$
12	Greyhound(Greg) $\Longrightarrow$ Dog(Greg)	3, UI	$[\alpha_{12}, \beta_{12}] = [1, 1]$
13	Dog(Greg)	12, 8, MP	$[\alpha_{13}, \beta_{13}] = [1, 1]$
14	Horse(Harry) $\wedge$ Dog(Greg) $\Longrightarrow$ Faster(Harry, Greg)	1, UI	$[\alpha_{14}, \beta_{14}] = [\alpha_1, \beta_1]$
15	Horse(Harry) $\wedge$ Dog(Greg)	5, 13, AI	$[\alpha_{15}, \beta_{15}] = [\alpha_5, \beta_5]$
16	Faster(Harry, Greg)	14, 15, MP	$[\alpha_{16}, \beta_{16}]$ obtained from solving $\begin{cases} \alpha_{14} = \max(1 - \beta_{15}, \alpha_{16}) \\ \beta_{14} = \max(1 - \alpha_{15}, \beta_{16}) \end{cases}$
17	Faster(Harry, Greg) $\wedge$ Faster(Greg, Ralph) $\Longrightarrow$ Faster(Harry, Ralph)	4, UI	$[\alpha_{17}, \beta_{17}] = [\alpha_4, \beta_4]$
18	Faster(Harry, Greg) $\wedge$ Faster(Greg, Ralph)	16, 11, AI	$[\alpha_{18}, \beta_{18}] = [\alpha_{16}, \beta_{16}]$
19	Faster(Harry, Ralph)	17, 18, MP	$[\alpha_{19}, \beta_{19}]$ obtained from solving $\begin{cases} \alpha_{17} = \max(1 - \beta_{18}, \alpha_{19}) \\ \beta_{17} = \max(1 - \alpha_{18}, \beta_{19}) \end{cases}$

MP rules to premises 2 to 6. In our initial example (only the first premise is uncertain), the uncertainty in premises 2 to 6 is $[\alpha_i, \beta_i] = [1, 1], i = 2, 3, \ldots, 6$. Uncertain logic operations become relevant in steps 14 to 19. For example, the uncertainty in premise 16 is obtained from solving the system of equations shown in the corresponding row in Table 2.3. This system of equations is derived from the models (2.33) and (2.34). As a consequence, any change in the uncertainty $[\alpha_1, \beta_1]$ directly affects $[\alpha_{16}, \beta_{16}]$. Figure 2.2 illustrates the result in a probabilistic scenario. Note that, for us to be able to conclude "Faster(Harry, Ralph)" given the initial uncertainty, α_4 must be larger than α_1. Similar results can be further verified by modifying uncertainties on the premises, whose values can be computed as indicated in Table 2.3.

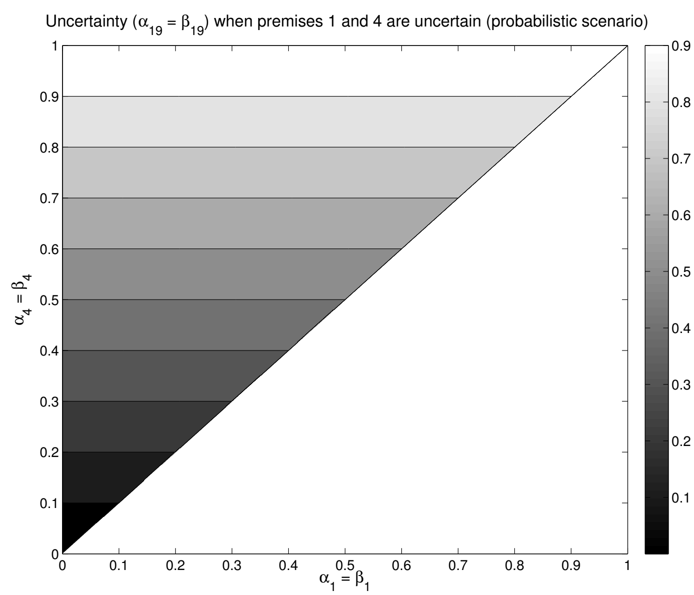


Figure 2.2: Uncertainty in Premise 19 of Table 2.3.

2.7.2.2 Example 2

Consider the following problem, extracted from [2]. The law says that it is a crime to sell an unregistered gun. Red has several unregistered guns, and all of them were purchased from Lefty. Based on these premises, can we derive the conclusion that

Lefty is a criminal? Moreover, how is this conclusion affected if these premises become uncertain? Table 2.4 contains the FOL representation of these premises, as well as a derivation of the desired conclusion. Note that, if all the rules and premises are true (i.e., Boolean scenario with true clauses, which are represented by an uncertainty interval $[1, 1]$), the conclusion effectively shows that Lefty is a criminal. However, an uncertainty interval starts growing in this conclusion when the premises lose certainty. We can track such uncertainty used through uncertain logic computations. The fourth column in Table 2.4 shows how uncertainty is tracked in this problem. Note how in the Boolean scenario the conclusion matches that of classical logic. Also, Note that, if the inputs (clauses 1-3) are probabilistic, the result is probabilistic too. Figure 2.3 illustrates a more general uncertain logic scenario, where the lower bound of the conclusion's uncertainty (i.e., α_{10}) is a function of the input arguments α_1 and β_2. By looking at this figure, we can identify a region (defined by $\alpha_1 + \beta_2 \leq 1$) for which we cannot derive the uncertainty of the conclusion. This is expected, given that, as mentioned above, we cannot always infer anything from a MP rule when we have not enough evidence for the antecedent. In addition, in the figure we can see how α_{10} increases as the evidence of the input premises increases (i.e., α_1 and β_2 increase).

2.8 Satisfiability in ULP

Section 2.7 shows how to use ULP for tracking uncertainties using a traditional inference approach. This approach, however, may not be well suited for some applications. For example, applications that rely on large amounts of data are prone to suffer scalability problems if a traditional inference scheme is followed. Then, alternatives for reducing complexity must be considered.

One of these alternatives is to solve a satisfiability problem in which, instead of using inference rules (e.g., MP) to derive conclusions and their associated uncertainties, we find the logical values (or uncertainties) that are unknown for the propositions in

Table 2.4: Premises and inference process for the derivation of the conclusion "Is Lefty a criminal?". The example and inference process are introduced in [2]. Uncertainty modeling and tracking is done using Uncertain Logic models.

	Logic Expression	Inference Rule	Uncertainty Interval
1	$\forall x \forall y \forall z \quad (\texttt{Sold}(x, y, z) \wedge \texttt{Unregistered}(y)) \implies \texttt{Criminal}(x)$	Δ	$[\alpha_1, \beta_1]$
2	$\exists y \quad \texttt{Owns}(\text{Red}, y) \wedge \texttt{Unregistered}(y)$	Δ	$[\alpha_2, \beta_2]$
3	$\forall y \quad (\texttt{Owns}(\text{Red}, y) \wedge \texttt{Unregistered}(y)) \implies \texttt{Sold}(\text{Lefty}, y, \text{Red})$	Δ	$[\alpha_3, \beta_3]$
4	$\texttt{Owns}(\text{Red}, \text{Gatling}) \wedge \texttt{Unregistered}(\text{Gatling})$	EI, 2	$[\alpha_4, \beta_4] = [\alpha_2, \beta_2]$
5	$(\texttt{Owns}(\text{Red}, \text{Gatling}) \wedge \texttt{Unregistered}(\text{Gatling})) \implies \texttt{Sold}(\text{Lefty}, \text{Gatling}, \text{Red})$	UI, 3	$[\alpha_5, \beta_5] = [\alpha_3, \beta_3]$
6	$\texttt{Sold}(\text{Lefty}, \text{Gatling}, \text{Red})$	MP, 5, 4	$[\alpha_6, \beta_6]$, with: $\alpha_6 = \begin{cases} \alpha_3, & \text{if } \alpha_3 > 1 - \beta_2; \\ 0, & \text{if } \alpha_3 = 1 - \beta_2; \\ \text{no solution}, & \text{otherwise;} \end{cases}$ $\beta_6 = \begin{cases} \beta_3, & \text{if } \beta_3 \leq 1 - \alpha_2; \\ \text{no solution}, & \text{otherwise.} \end{cases}$
7	$\texttt{Unregistered}(\text{Gatling})$	AE, 4	$[\alpha_7, \beta_7] = [\alpha_2, \beta_2]$
8	$(\texttt{Sold}(\text{Lefty}, \text{Gatling}, \text{Red}) \wedge \texttt{Unregistered}(\text{Gatling})) \implies \texttt{Criminal}(\text{Lefty})$	UI, 1	$[\alpha_8, \beta_8] = [\alpha_1, \beta_1]$
9	$\texttt{Sold}(\text{Lefty}, \text{Gatling}, \text{Red}) \wedge \texttt{Unregistered}(\text{Gatling})$	AI, 6, 7	$[\alpha_9, \beta_9] = [\min(\alpha_2, \alpha_6), \min(\beta_2, \beta_6)]$
10	$\texttt{Criminal}(\text{Lefty})$	MP, 8, 9	$[\alpha_{10}, \beta_{10}]$, with: $\alpha_{10} = \begin{cases} \alpha_1, & \text{if } \alpha_1 > 1 - \beta_9; \\ 0, & \text{if } \alpha_1 = 1 - \beta_9; \\ \text{no solution}, & \text{otherwise;} \end{cases}$ $\beta_{10} = \begin{cases} \beta_1, & \text{if } \beta_1 \leq 1 - \alpha_9; \\ \text{no solution}, & \text{otherwise.} \end{cases}$

a set of logic formulas such that the full set of logic formulas (i.e., our logic model) is true.

Formally, the satisfiability problem consists in determining whether there exists a variable assignment such that every formula in a group of logic formulas (i.e., the model) evaluates to true [57]. This concept can be extended into ULP. In this case, instead of focusing on true/false variable assignments, we need to find the uncertainty intervals that satisfy a formula or a set of formulas. This is equivalent to finding a possible interpretation for the unknown uncertainties in a ULP model. Although this formulation departs from the traditional satisfiability formulation which identifies

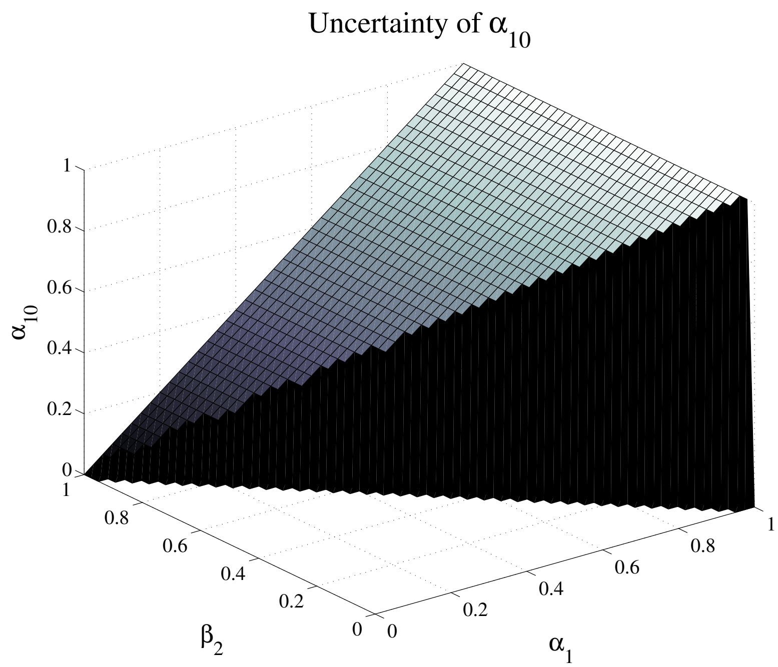


Figure 2.3: Lower bound of the uncertainty of the conclusion α_{10} of Table 2.4.

possible variable assignments, equivalent variable assignments in ULP can be found by identifying those grounded propositions with no uncertainty (i.e., with uncertainty interval $[1, 1]$).

It has been demonstrated that the SAT problem is NP complete. However, there are instances in many diverse areas for which this problem can be reformulated with simpler and efficient algorithms [57]. As it will become evident later in this section, this is also true for the case of ULP satisfiability, enabling scalable solutions for reasoning with ULP.

The satisfiability problem for ULP models can be formulated as an optimization problem, as follows.

2.8.1 ULP Satisfiability as an Optimization Problem

Consider a set Φ of uncertain propositions containing:

$$\begin{aligned} \{ \quad & \varphi_1(\cdot), \text{ with uncertainty } [\alpha_{\varphi_1}, \beta_{\varphi_1}], \\ & \varphi_2(\cdot), \text{ with uncertainty } [\alpha_{\varphi_2}, \beta_{\varphi_2}], \\ \ldots, \quad & \varphi_l(\cdot), \text{ with uncertainty } [\alpha_{\varphi_l}, \beta_{\varphi_l}] \quad \}, \end{aligned}$$

whose uncertainty intervals are known. Let us call these propositions the evidence. Also consider a set Ψ of uncertain propositions containing the elements:

$$\begin{aligned} \{ \quad & \psi_1(\cdot), \text{ with uncertainty } [\alpha_{\psi_1}, \beta_{\psi_1}], \\ & \psi_2(\cdot), \text{ with uncertainty } [\alpha_{\psi_2}, \beta_{\psi_2}], \\ \ldots, \quad & \psi_m(\cdot), \text{ with uncertainty } [\alpha_{\psi_m}, \beta_{\psi_m}] \quad \}, \end{aligned}$$

whose uncertainty intervals are unknown. Both evidence and unknown propositions are components of a ULP model made of n logic expressions:

$$\begin{aligned} & F_1 : f_1(\Phi, \Psi), \text{ with uncertainty } [\alpha_{F_1}, \beta_{F_1}], \\ & F_2 : f_2(\Phi, \Psi), \text{ with uncertainty } [\alpha_{F_2}, \beta_{F_2}], \quad \ldots, \\ & F_n : f_n(\Phi, \Psi), \text{ with uncertainty } [\alpha_{F_n}, \beta_{F_n}], \end{aligned}$$

where $f_1, f_2, \ldots, f_n$ are logic formulas. Without loss of generality, assume that these formulas are disjunctions of a subset of propositions in Φ and Ψ. For example, a ULP model could be defined as:

$$\begin{aligned} & F_1 : \varphi_1 \vee \psi_1, && \text{with uncertainty } [\alpha_{F_1}, \beta_{F_1}], \\ & F_2 : \varphi_2 \vee \varphi_3 \vee \psi_1, && \text{with uncertainty } [\alpha_{F_2}, \beta_{F_2}], \text{ and} \\ & F_3 : \varphi_3 \vee \psi_2, && \text{with uncertainty } [\alpha_{F_3}, \beta_{F_3}]. \end{aligned} \tag{2.36}$$

Furthermore, let us denote as $\alpha_{\Phi \in F_j}$ and $\alpha_{\Psi \in F_j}$ the set of beliefs (i.e., lower bound of the uncertainty intervals) of the propositions in Φ and Ψ, respectively, that are part of the formula $F_j, j = 1, 2, \ldots, n$. Similarly, let us denote as $\beta_{\Phi\in F_j}$ and $\beta_{\Psi\in F_j}$ the set of plausibilities (i.e., upper bound of the uncertainty intervals) of the propositions in Φ and Ψ, respectively, that are part of the formula F_j. For example, in the model defined by (2.36), the sets $\alpha_{\Phi\in F_j}$, $\beta_{\Phi\in F_j}$, $\alpha_{\Psi\in F_j}$, $\beta_{\Psi\in F_j}$, with $j = 1, 2, 3$, are:

$$
\begin{aligned}
\alpha_{\Phi\in F_1} &= \{\alpha_{\varphi_1}\}, & \beta_{\Phi\in F_1} &= \{\beta_{\varphi_1}\},\\
\alpha_{\Phi\in F_2} &= \{\alpha_{\varphi_2}, \alpha_{\varphi_3}\}, & \beta_{\Phi\in F_2} &= \{\beta_{\varphi_2}, \beta_{\varphi_3}\},\\
\alpha_{\Phi\in F_3} &= \{\alpha_{\varphi_3}\}, & \beta_{\Phi\in F_3} &= \{\beta_{\varphi_3}\},\\
\alpha_{\Psi\in F_1} &= \{\alpha_{\psi_1}\}, & \beta_{\Psi\in F_1} &= \{\beta_{\psi_1}\},\\
\alpha_{\Psi\in F_2} &= \{\alpha_{\psi_1}\}, & \beta_{\Psi\in F_2} &= \{\beta_{\psi_1}\},\\
\alpha_{\Psi\in F_3} &= \{\alpha_{\psi_2}\}, \quad \text{and} & \beta_{\Psi\in F_3} &= \{\beta_{\psi_2}\}.
\end{aligned}
$$

Then, the satisfiability problem in ULP can be defined as finding the uncertainty intervals $[\alpha_{\psi_i}, \beta_{\psi_i}]$ that solve the following optimization problem:

$$\underset{\{[\alpha_{\psi_i},\beta_{\psi_i}]\}}{\text{minimize}} \quad \sum_{j=1}^{n}(\alpha_{F_j} - \hat{\alpha}_{F_j})^2 + (\beta_{F_j} - \hat{\beta}_{F_j})^2 \tag{2.37a}$$

subject to

for all $j \in \{1, 2, \ldots, n\}$:

$$\hat{\alpha}_{F_j} = \max(\alpha_{\Phi\in F_j}, \alpha_{\Psi\in F_j}); \tag{2.37b}$$

$$\hat{\beta}_{F_j} = \max(\beta_{\Phi\in F_j}, \beta_{\Psi\in F_j}); \tag{2.37c}$$

$$0 \leq \hat{\alpha}_{F_j} \leq \hat{\beta}_{F_j} \leq 1; \text{ and} \tag{2.37d}$$

$$0 \leq \alpha_{\psi_i} \leq \beta_{\psi_i} \leq 1; i \in \{1, \ldots, m\}. \tag{2.37e}$$

It is important to note the following:

- The *cost function* in (2.37a) is based on an l_2 norm. However, the definition of the cost function should not be considered restricted to l_2 norms. Other norms could be used to enforce particular properties of the solution.

- The *constraints* (2.37b) and (2.37c) model the uncertainty of an OR operation in CFE-based ULP, as defined in Section 2.4.2 above. The use of a disjunctive model simplifies the formulation of the optimization problem. Similar constraints could be constructed to model logic formulas that involve other logic operators, or, alternatively, any other logic expression could be converted into a disjunctive model. Constraints (2.37d) and (2.37e) condition the uncertainty intervals to be defined as closed sets, consistent with DS theory definitions.

- When the solution of the optimization problem renders a cost function equal to zero, the logic model is satisfiable. In this case, there may be an infinite number of solutions to the optimization problem, and an additional step should be added to the reasoning process for enforcing minimal commitment in the output intervals (i.e., delivering the most conservative interval allocation).

When the cost function in the solution does not evaluate to zero, the model is not satisfiable. In this case it is possible, however, to identify particular formulas that are making the problem unsatisfiable (by identifying the non-zero components in the cost function), and either discard them from the model, or properly weight the evidence to ensure satisfiability.

Also note that the optimization problem (2.37) is nonlinear. This problem, however, can be converted into a convex optimization problem if in each of the logic expressions $F_1, F_2, \ldots, F_n$ there is *at most* one proposition whose uncertainty is unknown. Under this assumption, the optimization problem (2.37) can be reformulated as:

$$\underset{\{[\alpha_{\psi_i},\beta_{\psi_i}]\}}{\text{minimize}} \quad \sum_{j=1}^{n}(\alpha_{F_j} - \hat{\alpha}_{F_j})^2 + (\beta_{F_j} - \hat{\beta}_{F_j})^2 \tag{2.38a}$$

subject to

for all $j \in \{1, 2, \ldots, n\}$:

$$0 \leq \hat{\alpha}_{F_j} \leq \hat{\beta}_{F_j} \leq 1; \tag{2.38b}$$

$$\hat{\alpha}_{F_j} = \alpha_{\Psi \in F_j}; \quad \hat{\beta}_{F_j} = \beta_{\Psi \in F_j}; \tag{2.38c}$$

$$\hat{\alpha}_{F_k} \geq \alpha_{\Phi \in F_j}; \quad \hat{\beta}_{F_k} \geq \beta_{\Phi \in F_j}; \tag{2.38d}$$

$$0 \leq \alpha_{\psi_i} \leq \beta_{\psi_i} \leq 1; i \in \{1, \ldots, m\}. \tag{2.38e}$$

As a convex optimization problem, the complexity of this algorithm is significantly less than that of its corresponding nonlinear problem. This makes this approach valuable in scenarios with hundreds to thousands of logic formulas.

2.8.2 Example: MLNs and Optimization-based ULP

Consider the scenario described in [58], in which a knowledge base contains the following three rules:

1. "Friends of friends are friends";
2. "Smoking causes cancer"; and
3. "If two people are friends and one smokes, then so does the other".

As defined in [58], these rules could be expressed in first order logic as:

1. $\forall x \forall y \forall z$ $\texttt{Friends}(x, y) \wedge \texttt{Friends}(y, z) \implies \texttt{Friends}(x, z)$;
2. $\forall x$ $\texttt{Smokes}(x) \implies \texttt{Cancer}(x)$; and
3. $\forall x \forall y$ $\texttt{Friends}(x, y) \wedge \texttt{Smokes}(x) \implies \texttt{Smokes}(y)$,

respectively. Note that these types of rules would be rarely completely true (or false) in real-life problems [58]. However, if we are able to quantify and attach uncertainty measures to these rules, then we would have a valuable set of rules able to support meaningful automated reasoning. When using probabilities as the uncertainty measure, problems like the one in this example can be modeled (and solved) using MLNs. They could also be solved using ULP to take advantage of the increased degree of freedom and logic-consistency properties introduced in this paper.

Let us enhance this set of rules with a set of uncertain logic expressions and rules, as shown in Table 2.5. This set of logic expressions is applied on a domain of people defined as $\Theta_p = \{\text{Ivan}, \text{John}, \text{Katherine}, \text{Lars}, \text{Michael}, \text{Nick}\}$. For ease of explanation, friendship relations have been assumed to be perfectly known (i.e., with no uncertainty), with uncertainty intervals $[1, 1]$. The knowledge base and set of evidence are propotitionalized (i.e., grounded) over the domain Θ_p, along with the grounded propositions `Smokes(·)` and `Cancer(·)`. The propotitionalized knowledge base and evidence sets are further processed to convert them into conjunctive normal form, rendering a total of 438 propositions. Then, we automatically formulate the ULP convex optimization model (2.38) for this example and use it for answering questions regarding the uncertainty of particular logic propositions.

Let us consider first the effect that changing the uncertainty of rule F_3 has in the uncertainty of the proposition `Smokes()`. Figure 2.4 shows the uncertainty of this proposition as it applies to Katherine when the uncertainty of F_3 changes. In the figure, nine uncertainty intervals are considered for F_3, namely $[\alpha_{F3}, \beta_{F3} \in \{[1.00, 1.00], [0.75, 1.00], [0.50, 1.00], [0.50, 0.75], [0.50, 0.50], [0.25, 0.50], [0.00, 0.50], [0.00, 0.25], [0.00, 0.00]\}$. These uncertainty intervals represent various uncertainty conditions, including F_3 being completely true to completely false, as well as the uncertainty of F_3 being modeled with both probabilistic and DS models. This figure also shows the corresponding probabilities computed with MLNs (using Alchemy 1.0 [59]). Since MLNs rules require setting a weight for each logic expression, we set

Table 2.5: Set of logic formulas (i.e., rules) and evidence for the example of Section 2.8.2. For simplicity, the duals of evidence expressions to E_1, E_2, E_3, and E_4 are not shown in the table. These duals enforce an assumption of symmetry of a friendship relation (i.e., for two subjects d_1, d_2, the uncertainty of `Friends`(d_1, d_2) is the same of `Friends`(d_2, d_1).

Knowledge base of uncertain formulas/rules		
	Logic Expression	**Uncertainty**
F_1	$\forall x \forall y \forall z$ `Friends`(x, y) $\wedge$`Friends`$(y, z) \implies$ `Friends`(x, z)	$[\alpha_{F1}, \beta_{F1}] = [1, 1]$
F_2	$\forall x$ `Smokes`$(x) \implies$ `Cancer`(x)	$[\alpha_{F2}, \beta_{F2}]$
F_3	$\forall x \forall y$ `Friends`$(x, y) \wedge$ `Smokes`(x) $\implies$ `Smokes`(y)	$[\alpha_{F3}, \beta_{F3}]$

Evidence		
	Logic Expression	**Uncertainty**
E_1	`Friends( Ivan, John )`	$[\alpha_{E1}, \beta_{E1}] = [1, 1]$
E_2	`Friends( Katherine, Lars )`	$[\alpha_{E2}, \beta_{E2}] = [1, 1]$
E_3	`Friends( Michael, Nick )`	$[\alpha_{E3}, \beta_{E3}] = [1, 1]$
E_4	`Friends( Ivan, Michael )`	$[\alpha_{E4}, \beta_{E4}] = [1, 1]$
E_5	`Smokes( Ivan )`	$[\alpha_{E5}, \beta_{E5}]$
E_6	`Smokes( Nick )`	$[\alpha_{E6}, \beta_{E6}]$

the weights of each proposition using the pignistic probability of its corresponding uncertainty interval. We can observe the following:

- The uncertainty interval of the proposition `Smokes(Katherine)` resembles the uncertainty interval of rule F_3. This is expected given that, as described in Section 2.7 above, the uncertain model of implication rules (as the one in F_3) is bounded by the uncertainty of the rule.

- The pignistic probability (BetP curve in the figure) provides a probabilistic solution based on the ULP models.

- Unless there is complete certainty of the truthfulness of F_3, the probabilistic result rendered by an MLN model quickly drops to a value close to 0.4. Although for most of the intervals considered in the figure this probability value falls outside of the uncertainty interval provided by ULP, this result is the best an MLN can provide given its underlying probabilistic model. Recall that, unlike ULP, the probabilistic model of MLNs cannot naturally model the uncertainty of the negation of F_3. In addition, by definition, the MLN model does not ensure consistency with a classical logic model and there is no direct correspondence between a weight in a logic formula and a probability value. With these limitations, the best solution that the MLN can provide for the proposition F_3 is close to 0.5 (i.e., both events "Katherine smokes" and "Katherine does not smoke" are almost equally likely).

Let us now focus on the effect of changing the uncertainty of the rule F_2, and assume that F_3 is always true (with uncertainty $[1, 1]$). Figure 2.5) shows the uncertainty of the proposition `Cancer(Ivan)` as a function of the uncertainty interval $[\alpha_{F2}, \beta_{F2}]$. As in the analysis of the rule F_3, the following nine uncertainty intervals are considered for F_3: $[\alpha_{F2}, \beta_{F2}] \in \{[1.00, 1.00], [0.75, 1.00], [0.50, 1.00], [0.50, 0.75], [0.50, 0.50], [0.25, 0.50], [0.00, 0.50], [0.00, 0.25], [0.00, 0.00]\}$. Note how, from interval $[1.00, 1.00]$ down to interval $[0.50, 0.50]$ the uncertainty of `Cancer(Ivan)` follows the uncertainty

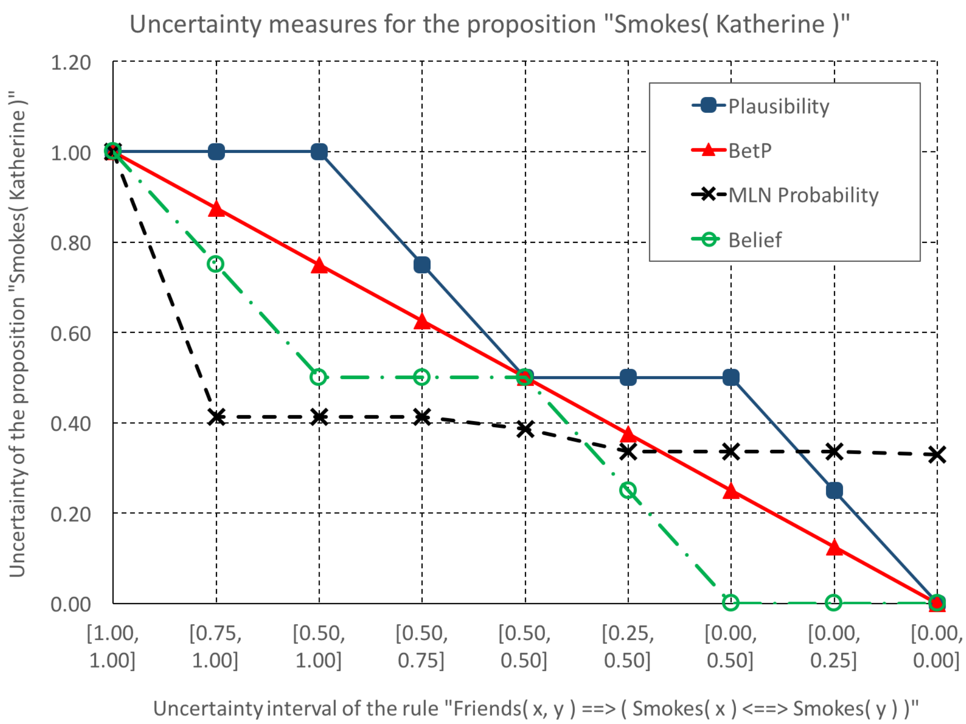


Figure 2.4: Uncertainty of the proposition `Smokes(Kate)` as a function of the uncertainty interval $[\alpha_{F3}, \beta_{F3}]$, which indicates the uncertainty of the logic formula F_3 in the example of Section 2.8.2. The figure shows the belief and plausibility of `Smokes(Kate)` using ULP. In addition, the figure shows a mapping of the resulting ULP model into a probabilistic model using the pignistic transformation (labeled as BetP in the figure), as well as a probabilistic estimate rendered by an MLN model. Note how the uncertainty of `Smokes(Kate)` rendered by ULP follows the behavior of the uncertainty of F_3, and provides information even if the uncertainty of the rule indicates that F_3 is false.

of the rule F_2. However, when there is insufficient evidence to support a satisfiable model a conclusion cannot be made on the uncertainty interval of `Cancer(Ivan)` in ULP. In this case, the uncertainty interval becomes $[0, 1]$ which indicates complete ignorance on the event `Cancer(Ivan)` being true or false. Note that rendering an uncertainty interval $[0, 1]$ in this case is consistent with the MP model (2.33) and (2.34): In the case of insufficient evidence to support a conclusion, the model is not satisfiable and a conservative solution $[0, 1]$ is output. Finally, the figure shows that the pignistic probabilistic transformation (i.e., BetP in the figure) renders results that are close to the output delivered by MLNs in this scenario.

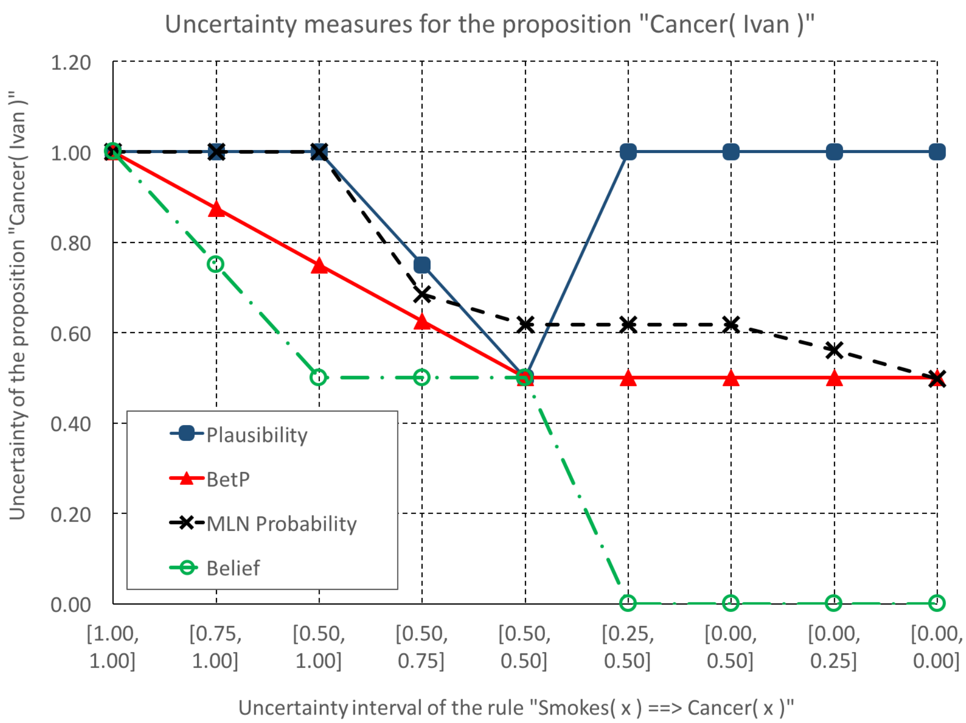


Figure 2.5: Uncertainty of the proposition `Cancer(Ivan)` as a function of the uncertainty interval $[\alpha_{F2}, \beta_{F2}]$, which indicates the uncertainty of the logic formula F_2 in the example of Section 2.8.2. The figure shows the belief and plausibility of `Smokes(Kate)` using ULP. In addition, the figure shows a mapping of the resulting ULP model into a probabilistic model using the pignistic transformation (labeled as BetP in the figure), as well as a probabilistic estimate rendered by an MLN model. Note how the uncertainty of `Cancer(Ivan)` rendered by ULP follows the behavior of the uncertainty of F_2. When there is not enough evidence to support an assignment of an uncertainty interval of `Cancer(Ivan)`, the ULP model renders as a result the interval that represents complete ignorance, namely $[0, 1]$.

CHAPTER 3

DS-Based Undirected Graphical Models

Graphical models are a popular framework for compact representation of a joint probability distribution over a large number of interdependent variables, and for efficient reasoning about such a distribution [60]. Graphical models define probability distributions in terms of a directed or undirected graph. The nodes in the graph are identified with random variables, and joint probability distributions are defined by taking products over functions defined on connected subsets of nodes. By exploiting the graph-theoretic representation, the formalism provides general algorithms for computing marginal and conditional probabilities of interest. Moreover, the formalism provides control over the computational complexity associated with these operations [61].

In this chapter we extend the concept of graphical models to the Demspter-Shafer domain. In particular, we lay up the foundations of a DS-based graphical model framework which is compatible with ULP reasoning. This framework inherits the advantages of traditional probabilistic graphical models. Furthermore, it enhances them by providing the ability for reasoning with uncertainty intervals. The graphical model formalism introduced in this chapter is an initial formulation of a well-founded DS-based graphical model theory, whose further development is matter of future work.

3.1 Probabilistic Graphical Models

In general, a graphical model has two components: 1) A graph whose nodes are the random variables and whose edges connect variables that interact directly (variables that are not directly connected are conditionally independent given some combination of the other variables); and 2) a set of functions (called factors) defined over a subset of the random variables [60]. A factor $f(\mathbf{x})$ is a function over a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_k\}$ such that $f(\mathbf{x}) \geq 0, \forall \mathbf{x}$ in the domain of $\mathbf{X}$. It is not required that $f(\mathbf{x}) \leq 1$, i.e., factors are not required to be probability distributions. Given a complete joint assignment $\mathbf{x}$ to the variables in $\mathbf{X}$, a joint distribution is defined by:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i f_i(\mathbf{x}), \tag{3.1}$$

where f_i represents all the factors of the form f defined by the graph, and $Z = \sum_{\mathbf{x}'} \prod_i f_i(\mathbf{x}')$ is a normalization constant.

Directed graphical models (also known as Bayesian networks), are typically used to represent causal or asymmetric interactions amongst a set of random variables. A directed edge from variable X_i to variable X_j in the graph (which must be acyclic) is used to indicate that X_i directly influences X_j. The probability distribution that a directed graphical model defines is:

$$p(\mathbf{x}) = p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | \mathbf{x}_i^p), \tag{3.2}$$

where $\mathbf{x}_i^p$ represents the set of parents of node X_i. Note that (3.2) is a conditional probability distribution over a node given its parents in the graph.

Undirected graphical models, or Markov Networks, are useful for representing distributions over variables where there is no natural directionality to the influence of one variable over another and where the interactions are more symmetric [61]. Let G be the undirected graph over the random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ corresponding to a Markov network, and let C denote the set of cliques (i.e., complete subgraphs) of G. Then, the probability distribution represented by the Markov network factorizes as follows:

$$p(\mathbf{x}) = p(x_1, x_2, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} f_c(\mathbf{x}), \tag{3.3}$$

where $f_c(\mathbf{x})$ are the factors over a complete subgraph of G, and $Z = \sum_X \prod_{c \in C} f_c(\mathbf{x})$ is the normalization constant.

3.2 Undirected Graphical Models for Probabilistic Extensions of Logic Reasoning

Undirected graphical models have been recently considered for the creation of probabilistic extensions of logic reasoning systems. The reason is that the undirected graph approach is more general when causal structure is unknown. Undirected graphs are used as the substrate of Markov Logic Networks (MLN) [22] and Probabilistic Soft Logic (PSL) [23], two recent approaches for providing logic inference with probabilistic reasoning. These two approaches are briefly described next.

3.2.1 Markov Logic Networks

A Markov Logic Network is a set of weighted first order logic clauses. It defines a Markov network, with a feature corresponding to each ground clause. The weight of each feature is the weight of the corresponding first order clause. If $\mathbf{x}$ is a possible assignment of truth values to all ground atoms, then

$$p(\mathbf{x}) = \frac{1}{Z} \exp \sum_i w_i n_i(\mathbf{x}), \tag{3.4}$$

where w_i is the weight of the ith clause, $n_i(\mathbf{x})$ is its number of true groundings in $\mathbf{x}$, and $Z = \sum_x \exp \sum_i w_i n_i(\mathbf{x})$ is a normalization constant. MLN weights can be learned generatively using pseudo-likelihood [22] or discriminatively using a variety of techniques [62]. MLN structure can be learned using a form of inductive logic programming [63].

Once the weights are learned, inference in MLN is done based on the function:

$$p(\mathbf{x}) = \frac{1}{Z} \exp \sum_i w_i f_i(\mathbf{x}), \tag{3.5}$$

where $f_i(\mathbf{x}) = 1$ if the ith clause is true, and 0 otherwise.

Inference in MLNs is carried out using a weighted satisfiability solver, or through more computationally efficient such as Gibbs sampling or MC-SAT. Recent research on MLNs has rendered Tractable Markov Logic (TML) [64]. TML defines a subset of MLNs, and uses probabilistic class and part hierarchies to control complexity. It relies on sum-product networks [65] for efficient computation of the normalization constant.

3.2.2 Probabilistic Soft Logic

Probabilistic Soft Logic (PSL) is a framework for probabilistic reasoning in relational domains [23]. As MLNs, PSL uses first order logic as a template language to specify undirected graphical models where ground atoms correspond to random variables and first order formulas encode dependencies among these variables and induce the features of the graphical mode. A key difference with MLN is, however, that PSL relaxes the Boolean truth values of MLNs to continuous truth values in the interval $[0, 1]$. In addition, PSL restricts the syntax of first order formulas to that of rules with conjunctive bodies. These two differences allow to solve inference problems in PSL as convex optimization problems in continuous space, enabling a more efficient inference that what MLNs can do [23].

Logical operations within the PSL framework are based on the Lukasiewicz t-norm and its corresponding co-norm for the AND and OR operations, respectively. Given an interpretation I, the formulas for the AND, OR, and negation in PSL are as follows:

$$
\begin{aligned}
\varphi_1 \;\tilde{\wedge}\; \varphi_2 &= \max(0, I(\varphi_1) + I(\varphi_2) - 1), \\
\varphi_1 \;\tilde{\vee}\; \varphi_2 &= \min(I(\varphi_1) + I(\varphi_2), 1), \\
\tilde{\neg}\; \varphi_1 &= 1 - I(\varphi_1),
\end{aligned} \tag{3.6}
$$

where the ˜ symbol indicates a relaxation from the Boolean domain.

A PSL program induces a distribution over possible interpretations I. Let R be the set of all ground rules that are instances of a rule in the program and only mentions atoms in a set l of interest. The probability density function $p(I)$ is:

$$p(I) = \frac{1}{Z} \exp\left(-\sum_{r \in R} \lambda_r (d_r(I))^q\right), \tag{3.7}$$

where $Z = \int_I \frac{1}{Z} \exp - \sum_{r \in R} \lambda(d_r(I))^q$ is a normalization constant, λ_r is the weight of the rule r, $d_r(I)$ is a distance-to-satisfaction metric, and $q \in \{1, 2\}$ provides a choose of two different loss functions. $d_r(I)$ is defined for an implication rule $\varphi_1 \implies \varphi_2$ as $d_r(I) = \max(0, I(\varphi_1) - I(\varphi_2))$. Weight learning in PSL can be done using maximum likelihood estimation. However, this problem is often intractable and its solution currently relies on approximated algorithms.

Inference in PSL targets two tasks: 1) Most Probable Explanation (MPE), which is inferring most likely values for a set of propositions given values of the remaining propositions as evidence; and 2) computing marginal distributions. MPE inference can be obtained by maximizing the density function $p(I)$ in (3.7), which is equivalent to minimizing the summation in the exponent, subject to both the evidence and any additional constraints. This optimization problem can be solved in polynomial time. The computation of marginal probabilities refers to calculating the probability that an atom l_i takes a value from a given interval $[l, u]$, i.e., finding $p(l \leq I(l_i) \leq u)$. This second problem is more complex, and to solve it, marginal distributions are approximated by collecting a histogram of sampled points following Markov Chain Monte Carlo schemes.

3.3 Graphical Models for Uncertain Logic Processing

3.3.1 General Framework

We propose the formulation of undirected graphical models in the DS domain based on the graph model defined for MLNs, which are described in in [58, 66]. A Markov network models the joint distribution of a set of random variables $X = (X_1, X_2, \ldots, X_n) \in \Xi$ as follows:

$$P(X = x) = \frac{1}{Z} \prod_j \phi_j(x_{\{j\}}), \tag{3.8}$$

where $x_{\{j\}}$ is the state of the j-th clique, ϕ_k is called a potential function for clique j, and Z is a normalization constant such that $Z = \sum_{x \in \Xi} \prod_j \phi_j(x_{\{j\}})$. A log-linear equivalent formulation is described by (3.5), i.e., by: $P(X = x) = \frac{1}{Z} \exp\left(\sum_j w_j f_j(x)\right)$, where f_j is called a "feature" function of the state j, and w_j is a weight. A feature may be any real-valued function of the state [22].

A Markov Logic Network L (which we use as the general framework for DS-based graphical models) is defined as a set of pairs (Λ_j, w_j), where Λ_j is a formula in first order logic (as defined above) and w_j is a real number. Together with any required set of constraints C, the pairs (Λ_j, w_j) define a Markov network $M_{\{L,C\}}$ (e.g., models (3.8) and (3.5) above), as follows:

1) $M_{\{L,C\}}$ contains one binary node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground predicate is true, and 0 otherwise.

2) $M_{\{L,C\}}$ contains one feature for each possible grounding of each formula Λ_j in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_j associated with F_j in L.

In its original formulation, an MLN can be seen as a template for constructing Markov networks. Based on models (3.8) and (3.5) as well as on items 1) and 2) above, the probability distribution over possible worlds x specified by the ground Markov network $M_{\{L,C\}}$ is given by:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_j w_j n_j(x)\right) = \frac{1}{Z} \prod_j \phi_j(x_{\{j\}})^{n_j(x)}, \tag{3.9}$$

where $n_j(x)$ is the number of true groundings of Λ_j in x, $x_{\{j\}}$ is the state (truth values) of the predicates appearing in Λ_j, and $\phi_j(x_{\{j\}}) = \exp(w_j)$.

3.3.2 A Feature Function for DS-based Graphical Models

We can relax the conditions 1) and 2) described in Section 3.3.1 above in such a way that, for each logic expression Λ_j, we quantify uncertainty intervals $[\alpha_j, \beta_j]$ instead of groundings of the expression. Furthermore, we can define features as non-negative real numbers. In this case, we introduce a feature function $\mathcal{L}$ that captures information embedded in the uncertainty intervals. The feature related to Λ_j as $\mathcal{L}_j$. Note that, in a traditional MLN (as introduced in [22]), the feature function f is defined as a count on the number of times logic expression Λ_j is found to be true in a knowledge base. This count is assumed to be proportional to the probability of Λ_j being true. In our extension to the DS framework, the feature function becomes the support for Λ_j being true or not being false.

We can further express the feature $\mathcal{L}_j$ as $\mathcal{L}_j = \mathcal{L}_j^{\{\mathbf{1}\}} + \mathcal{L}_j^{\{\mathbf{0}\}}$, which emphasizes the notion that $\mathcal{L}_j$ captures a consolidated probabilistic metric for the events "Λ_j is true" and "Λ_j is not false", respectively. The most natural definition for $\mathcal{L}_j^{\{\mathbf{1}\}}$ and $\mathcal{L}_j^{\{\mathbf{0}\}}$ is obtained directly from the uncertainty interval $[\alpha_j, \beta_j]$. Then, in the following, we let $\mathcal{L}_j^{\{\mathbf{1}\}} = \alpha_j$ and $\mathcal{L}_j^{\{\mathbf{0}\}} = \beta_j$.

When we consider the use of the new feature function $\mathcal{L}$, (3.5) becomes:

$$\begin{aligned} P(X=x) &= \frac{1}{Z}\exp\left(\sum_j w_j f_j(x_j)\right) \\ &= \frac{1}{Z}\exp\left(\sum_j w_j \mathcal{L}_j(x_j)\right) \\ &= \frac{1}{Z}\exp\left(\sum_j w_j\left(\mathcal{L}_j^{\{\mathbf{1}\}}(x_j)+\mathcal{L}_j^{\{\mathbf{0}\}}(x_j)\right)\right), \end{aligned} \tag{3.10}$$

where $x_j = \{[\alpha_1,\beta_1],[\alpha_2,\beta_2],\ldots,[\alpha_{M_j},\beta_{M_j}]\}$, the uncertainty intervals of the propositions that make up Λ_j. Some important remarks:

- The model described by (3.10) captures uncertainty intervals from propositions x_j and logic expressions Λ_j, and renders a probability measurement $P(X=x)$. Hence, this model can be seen as a probabilistic transformation of the DS-based logic model (the comparison of this transformation with existing probabilistic transformations for DS models is a matter of future work).

- Equation (3.10) can be also expressed as:

$$\begin{aligned} P(X=x) &= \frac{1}{Z}\exp\sum_j w_j\left(\mathcal{L}_j^{\{\mathbf{1}\}}(x_j)+\mathcal{L}_j^{\{\mathbf{0}\}}(x_j)\right) \\ &= \frac{1}{Z}\exp\sum_j \left(w_j\mathcal{L}_j^{\{\mathbf{1}\}}(x_j)+w_j\mathcal{L}_j^{\{\mathbf{0}\}}(x_j)\right) \\ &= \frac{1}{Z}\exp\sum_j \left(w_j\mathcal{L}_j^{\{\mathbf{1}\}}(x_j)+w_j\mathcal{L}_j^{\{\mathbf{0}\}}(x_j)\right) \\ &= \left[\frac{1}{Z_1}\exp\left(\sum_j w_j\mathcal{L}_j^{\{\mathbf{1}\}}(x_j)\right)\right]\times\left[\frac{1}{Z_2}\exp\left(\sum_j w_j\mathcal{L}_j^{\{\mathbf{0}\}}(x_j)\right)\right] \\ &= P^{\{\mathbf{1}\}}(X=x)\times P^{\{\mathbf{0}\}}(X=x), \end{aligned} \tag{3.11}$$

  with $Z = Z_1 Z_2$. In this case, $P^{\{\mathbf{1}\}}(X=x)$ could be seen as quantifying the event in which all the Λ_j expressions are true, while $P^{\{\mathbf{0}\}}(X=x)$ quantifies the event in which the Λ_j expressions are false. If the condition $0 \leq P^{\{\mathbf{1}\}}(X =$

$x) \leq 1 - P^{\{1\}}(X = x) \leq 1$ is maintained, this formulation could lead to an uncertainty interval on the logic model that is being captured by the graph.

Using the feature function $\mathcal{L}$, the problem of finding the most probable world (or maximum satisfiability) becomes:

$$\arg\max \frac{1}{Z} \exp\left(\sum_j w_j \mathcal{L}_j(x)\right). \tag{3.12}$$

Note that this problem is equivalent to:

$$\arg\max \sum_j w_j \mathcal{L}_j(x). \tag{3.13}$$

Furthermore, given that $\mathcal{L}_j = \mathcal{L}_j^{\{1\}} + \mathcal{L}_j^{\{1\}} = \alpha_j + \beta_j$, the objective function in (3.13) is maximal when $\alpha_j = \beta_j = 1, \forall j$. That is to say, the objective function is maximal when all the logic expressions Λ_j hold true. This would lead to $P(X = x) = 1$ in (3.10). Similarly, the objective function in (3.13) is minimal when $\alpha_j = \beta_j = 0, \forall j$. In this case, the interpretation is that all the logic expressions Λ_j are false. This is an expected behavior for a model that captures probabilities (or uncertainty intervals) of logic models encoded in graphical models.

3.3.3 Pseudo-likelihood Operators for Building DS-based Feature Functions

The feature function $\mathcal{L}$ introduced in Section 3.3.2 provides a suitable definition that properly and advantageously captures uncertainty information from logic expressions Λ_j for its use within graphical model frameworks. Due to the ability of this feature function for consistently increasing or decreasing the probability of logic expressions Λ_j being true depending on the amount of evidence embedded in the uncertainty interval, we could see this feature function as a pseudo-likelihood operator (recall that likelihood operators are used in statistics for estimating parameters or expected outcomes).

The problem now becomes the definition of $\mathcal{L}$ (as well as functions $\mathcal{L}^{\{\mathbf{1}\}}$ and $\mathcal{L}^{\{\mathbf{0}\}}$) as a function of the uncertainty intervals of the propositions that are part of the logic expressions Λ_j. A possible definition that is advantageous because it leads to a convex formulation of a ULP satisfiability problem is described next.

3.3.3.1 Likelihood of an ULP Formula Being True

Consider an uncertain logic expression of the form:

$$\Lambda : \quad \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_M, \tag{3.14}$$

with M being the number of uncertain propositions in the logic conjunction Λ, propositions φ_i having uncertainty intervals $[\alpha_i, \beta_i]$, and $i = 1, 2, \ldots, M$. When using the CFE as the fusion operator, the BBA of the conjunction of the M uncertain propositions above can be computed as:

$$\begin{aligned} m(\bigwedge_{i=1}^{M} \varphi_i) = [\quad & \Gamma^{\{\mathbf{1}\}}\mathbf{a} + \mathbf{\Gamma}^{\{\mathbf{1,0}\}}\mathbf{d}, \\ & \Gamma^{\{\mathbf{0}\}}(\mathbf{1} - \mathbf{b}) + \mathbf{\Gamma}^{\{\mathbf{1,0}\}}\mathbf{d} \quad], \end{aligned} \tag{3.15}$$

where $\mathbf{a} = [\alpha_{\mathbf{1}}, \alpha_{\mathbf{2}}, \ldots, \alpha_{\mathbf{M}}]^{\mathbf{T}}$ is an $M \times 1$ vector containing the beliefs of the φ_i propositions; $\mathbf{b} = [\beta_{\mathbf{1}}, \beta_{\mathbf{2}}, \ldots, \beta_{\mathbf{M}}]^{\mathbf{T}}$ is an $M \times 1$ vector containing the plausibilities of the φ_i propositions; $\mathbf{1}$ is the $M \times 1$ vector $[1, 1, \ldots, 1]^T$; $\mathbf{d} = [(\beta_{\mathbf{1}} - \alpha_{\mathbf{1}}), (\beta_{\mathbf{2}} - \alpha_{\mathbf{2}}), \ldots, (\beta_{\mathbf{M}} - \alpha_{\mathbf{M}})]^{\mathbf{T}}$ is an $M \times 1$ vector containing the uncertainty quantification $(\beta_i - \alpha_i)$ corresponding to each proposition φ_i; $\Gamma^{\{\mathbf{1}\}} = [\gamma_{\varphi_1}, \gamma_{\varphi_2}, \ldots, \gamma_{\varphi_M}]$ is a $1 \times M$ vector containing the CFE coefficients corresponding to the events in which φ_i is true; $\Gamma^{\{\mathbf{0}\}} = [\gamma_{\neg\varphi_1}, \gamma_{\neg\varphi_2}, \ldots, \gamma_{\neg\varphi_M}]$ is a $1 \times M$ vector containing the CFE coefficients corresponding to the events in which φ_i is false; and $\Gamma^{\{\mathbf{1,0}\}} = [\gamma_{\varphi_1,\neg\varphi_1}, \gamma_{\varphi_2,\neg\varphi_2}, \ldots, \gamma_{\varphi_M,\neg\varphi_M}]$ is a $1 \times M$ vector containing the CFE coefficients corresponding to the events in which φ_i is either true or false (but unknown). It is worth noting that the CFE coefficients in $\Gamma^{\{\mathbf{1}\}}$, $\Gamma^{\{\mathbf{0}\}}$ and $\Gamma^{\{\mathbf{1,0}\}}$ must be defined in such a way that the properties of the conjunction are maintained (see [42]).

Let us now define the likelihood function of the DS model that describes the belief of Λ being true as:

$$\begin{aligned}\mathcal{L}_{\Lambda}^{\{1\}}(m_i, \Gamma) &= \exp\left(m(\Lambda)\right) \\ &= \exp\left(\Gamma^{\{1\}}\mathbf{a} + \mathbf{\Gamma}^{\{1,0\}}\mathbf{d}\right),\end{aligned} \tag{3.16}$$

where α_{Λ} is the belief of Λ being true and $\mathbf{\Gamma} = \{\mathbf{\Gamma}^{\{1\}}, \mathbf{\Gamma}^{\{0\}}, \mathbf{\Gamma}^{\{1,0\}}\}$.

3.3.3.2 Likelihood of an ULP Formula Being False

Similarly, we can build the likelihood function of the DS model that describes the belief of Λ not being false as:

$$\begin{aligned}\mathcal{L}_{\Lambda}^{\{0\}}(m_i, \Gamma) &= \exp\left(m(\neg\Lambda_j)\right) \\ &= \exp\left(\Gamma^{\{0\}}(\mathbf{1} - \mathbf{b}) + \mathbf{\Gamma}^{\{1,0\}}\mathbf{d}\right),\end{aligned} \tag{3.17}$$

where the vectors $\mathbf{b}$, $\mathbf{d}$, $\mathbf{1}$ are defined as indicated in Section 3.3.3.1 above, and $\mathbf{\Gamma} = \{\mathbf{\Gamma}^{\{1\}}, \mathbf{\Gamma}^{\{0\}}, \mathbf{\Gamma}^{\{1,0\}}\}$.

3.3.3.3 Global Likelihood of an ULP Formula

By combining the concepts behind the definitions in (3.16) and (3.17), we can define the likelihood of the overall uncertain logic expression Λ as follows:

$$\begin{aligned}\mathcal{L}_{\Lambda}(m, \mathbf{a}, \mathbf{b}, \mathbf{\Gamma}) &= \exp\left(\quad m(\Lambda) + m(\neg\Lambda) \quad\right) \\ &= \exp\{(\Gamma^{\{1\}}\mathbf{a} + \mathbf{\Gamma}^{\{1,0\}}\mathbf{d}) + (\mathbf{\Gamma}^{\{0\}}(\mathbf{1} - \mathbf{b}) + \mathbf{\Gamma}^{\{1,0\}}\mathbf{d})\}.\end{aligned} \tag{3.18}$$

3.3.3.4 Likelihood of a Set of ULP Formulas

We can further define the likelihood function of the joint belief of all Λ_j, $j = 1, 2, ..., N$ (i.e., likelihood of the logic model $\Phi = \{\Lambda_1, \Lambda_2, \cdots, \Lambda_N\}$) as:

$$\begin{aligned}\mathcal{L}_{\Phi}(\{m_j\}, \{\Gamma_j\}) &= \prod_{j=1}^{N} \mathcal{L}_{\Lambda_j}.\\ &= \exp\{\sum_{j=1}^{N} \left(\Gamma_j^{\{1\}}\mathbf{a_j} + \mathbf{\Gamma_j^{\{1,0\}}d_j}\right), + \left(\Gamma_j^{\{0\}}(\mathbf{1} - \mathbf{b_j}) + \mathbf{\Gamma_j^{\{1,0\}}d_j}\right)\}.\end{aligned} \tag{3.19}$$

Note that, if we incorporate a normalization factor $\frac{1}{Z}$ and let $w_j = 1$ for all j in (3.13), this model becomes the cost function in (3.13), which is our generic formulation of DS-based graphical models. Given that this model represents a convex function, it supports the implementation of scalable solvers of (3.13). Furthermore, the CFE coefficients subsume the effect of the individual weights w_j.

3.4 Reasoning with DS-based graphical models

3.4.1 Pseudo-maximum Likelihood for ULP Expressions

The pseudo-likelihood functions introduced in Section 3.3.3 can be used for estimating the CFE coefficients of ULP logic formulas. Based on the pseudo-likelihood formulation in (3.19), the corresponding pseudo Maximum Likelihood Estimation (MLE) framework in this case is defined as:

$$\begin{aligned}&\arg\max_{\Gamma_j} \mathcal{L}_{\Phi}(\{m_j\}, \{\Gamma_j\})\\ &\quad = \arg\max_{\Gamma_j} \exp \sum_{j=1}^{N} \left(\Gamma_j^{\{1\}}\mathbf{a_j} + \mathbf{\Gamma_j^{\{1,0\}}d_j} + \mathbf{\Gamma_j^{\{0\}}}(\mathbf{1} - \mathbf{b_j}) + \mathbf{\Gamma_j^{\{1,0\}}d_j}\right).\end{aligned} \tag{3.20}$$

Similarly, given some CFE coefficients, MLE can be used for estimating unknown beliefs and plausibilities of propositions in a set of formulas Φ, that is:

$$\begin{aligned}&\arg\max_{\mathbf{a_j},\mathbf{b_j}} \mathcal{L}_\Phi(\{m_j\},\{\Gamma_j\})\\ &\qquad = \arg\max_{\mathbf{a_j},\mathbf{b_j}} \exp \sum_{j=1}^{N} \left(\Gamma_j^{\{\mathbf{1}\}}\mathbf{a_j} + \boldsymbol{\Gamma}_{\mathbf{j}}^{\{\mathbf{1,0}\}}\mathbf{d_j}\right) + \left(\Gamma_j^{\{\mathbf{0}\}}(\mathbf{1}-\mathbf{b_j}) + \boldsymbol{\Gamma}_{\mathbf{j}}^{\{\mathbf{1,0}\}}\mathbf{d_j}\right). \quad (3.21)\end{aligned}$$

In both cases, constraints can be added to the MLE problem to enforce satisfying particular properties of the solution. For example, [42] provides constraints on the CFE coefficients so that consistency with classical logic is satisfied. If an application does not demand such consistency, then these constraints may be relaxed and others more in tune with the application (or none, if that is the case) could be used instead.

Example: Estimating CFE coefficients. Consider the logic conjunction $\varphi_1 \wedge \varphi_2$, where the uncertainty of the propositions φ_1 and φ_2 is $[\alpha_1, \beta_1]$ and $[\alpha_2, \beta_2]$, respectively. A MLE model that allows us to estimate the CFE coefficients for this conjunction is given by:

$$\begin{aligned}&\arg\max_{\Gamma} \mathcal{L}_\Phi(\mathbf{a}=[\alpha_\mathbf{1},\alpha_\mathbf{2}], \mathbf{b}=[\beta_\mathbf{1},\beta_\mathbf{2}], \boldsymbol{\Gamma})\\
&= \arg\max_{\Gamma} \quad \Bigg(\begin{bmatrix}\gamma_1^{\{\mathbf{1}\}} & \gamma_2^{\{\mathbf{1}\}}\end{bmatrix} \times \begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix}\\
&\qquad\qquad + \begin{bmatrix}\gamma_1^{\{\mathbf{1,0}\}} & \gamma_2^{\{\mathbf{1,0}\}}\end{bmatrix} \times \begin{bmatrix}\beta_1-\alpha_1\\ \beta_2-\alpha_2\end{bmatrix} \Bigg)\\
&\qquad + \Bigg(\begin{bmatrix}\gamma_1^{\{\mathbf{0}\}} & \gamma_2^{\{\mathbf{0}\}}\end{bmatrix} \times \begin{bmatrix}1-\beta_1\\ 1-\beta_2\end{bmatrix}\\
&\qquad\qquad + \begin{bmatrix}\gamma_1^{\{\mathbf{1,0}\}} & \gamma_2^{\{\mathbf{1,0}\}}\end{bmatrix} \times \begin{bmatrix}\beta_1-\alpha_1\\ \beta_2-\alpha_2\end{bmatrix} \Bigg),\end{aligned}$$

where $\Gamma = \{\gamma_1^{\{\mathbf{1}\}}, \gamma_2^{\{\mathbf{1}\}}, \gamma_1^{\{\mathbf{0}\}}, \gamma_2^{\{\mathbf{0}\}}, \gamma_1^{\{\mathbf{1,0}\}}, \gamma_2^{\{\mathbf{1,0}\}}\}$. The following constraints are used for enforcing conjunction and DS properties:

$$\begin{aligned}
&\alpha_{\varphi_1 \wedge \varphi_2} \leq \alpha_1, \alpha_2; \\
&\beta_{\varphi_1 \wedge \varphi_2} \leq \beta_1, \beta_2; \\
&0 \leq \alpha_{\varphi_1 \wedge \varphi_2} \leq \beta_{\varphi_1 \wedge \varphi_2} \leq 1; \text{and} \\
&\sum_{i=1}^{2} \gamma_i^{\{\mathbf{1}\}} + \gamma_i^{\{\mathbf{0}\}} + \gamma_i^{\{\mathbf{1,0}\}} = 1.
\end{aligned} \tag{3.22}$$

Figure 3.1 shows the uncertainties of the formula $\varphi_1 \wedge \varphi_2$ that are obtained by maximizing the CFE coefficients in 4 different scenarios: (a) when the uncertainty of φ_2 is $[\alpha_2, \beta_2] = [0, 0]$, (b) when $[\alpha_2, \beta_2] = [1, 1]$, (c) when $[\alpha_2, \beta_2] = [0, 1]$, and (d) when $[\alpha_2, \beta_2] = [0.25, 0.80]$. The corresponding CFE coefficients obtained when the uncertainty of φ_1 is $[\alpha_1, \beta_1] = [1, 1]$ are shown in Figure 3.2. Some important observations:

- The MLE tunes the model for assigning the largest amount of evidence to the events $(\varphi_1 \wedge \varphi_2)$ and $\neg(\varphi_1 \wedge \varphi_2)$.
- The belief of the formula $(\varphi_1 \wedge \varphi_2)$ is always less than the belief of any of the propositions φ_1 and φ_2. This is consistent with the properties for the AND operation that we are enforcing through the constraints of the MLE model.
- Similarly, the plausibility of the formula $(\varphi_1 \wedge \varphi_2)$ is always less than the plausibility of any of the propositions φ_1 and φ_2.
- In the extreme cases, the following results are obtained:
  - When $[\alpha_1, \beta_1] = [0, 0]$ and $[\alpha_2, \beta_2] = [0, 0]$ the uncertainty of the formula $\varphi_1 \wedge \varphi_2$ is $[0, 0]$. This is consistent with the classical logic result where the AND operation is false if both of the input propositions are false.
  - When the uncertainty of any of the propositions φ_1 or φ_2 is $[0, 0]$, the uncertainty of the formula $\varphi_1 \wedge \varphi_2$ is $[0, 0]$. This is consistent with the

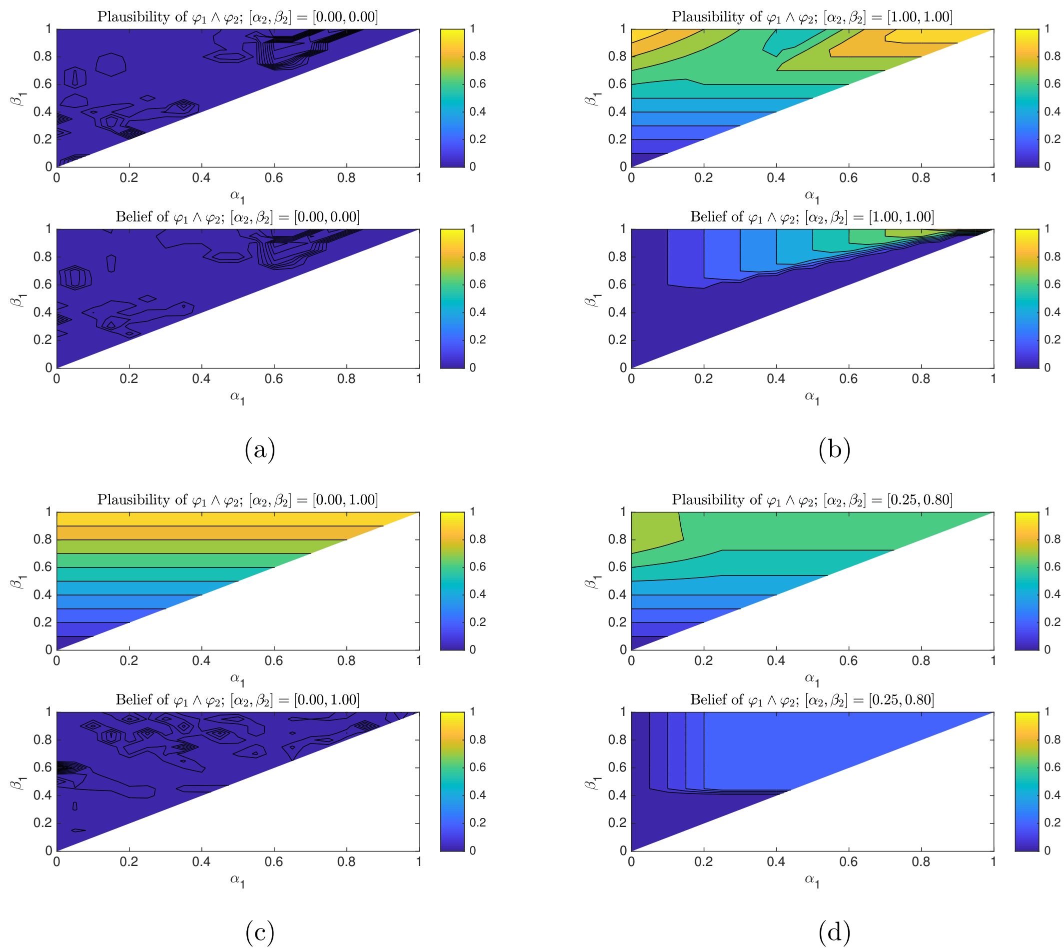


Figure 3.1: Uncertainty intervals (i.e., beliefs and plausibilities) obtained for the $\varphi_1 \wedge \varphi_2$ operation for different values of $[\alpha_1, \beta_2]$ (i.e., uncertainty of φ_1) when the uncertainty of φ_2 is: (a) $[0, 0]$, (b) $[1, 1]$, (c) $[0, 1]$, and $[0.25, 0.80]$.

classical logic result where the AND operation is false if any of the input propositions is false.

- When $[\alpha_1, \beta_1] = [\alpha_2, \beta_2] = [1, 1]$, the uncertainty of the formula $\varphi_1 \wedge \varphi_2$ is $[1, 1]$. This is consistent with the classical logic result where the AND operation is true if both of the input propositions are true.
- The MLE model is a convex optimization problem. Hence, its solution renders a global minimum.

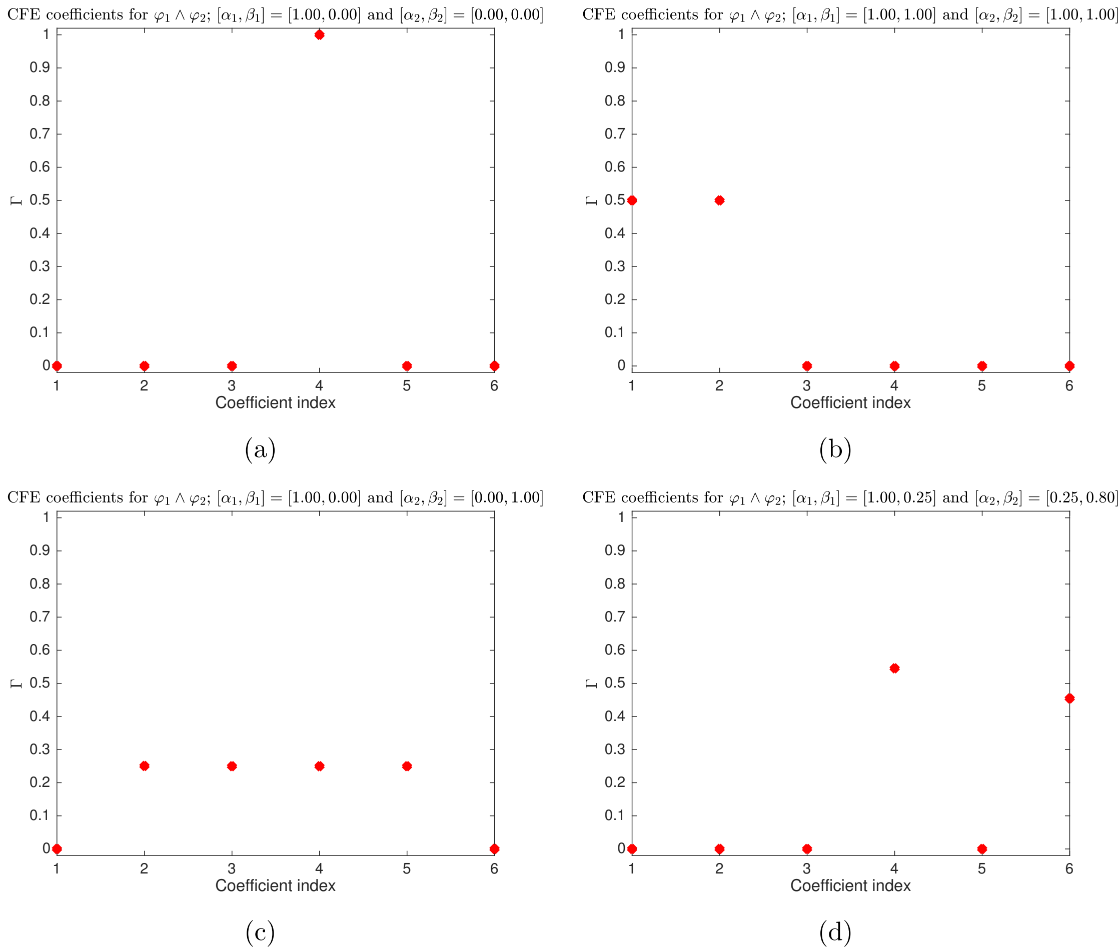


Figure 3.2: CFE coefficients obtained via pseudo MLE for the operation $\varphi_1 \wedge \varphi_2$ when the uncertainty of φ_1 is $[\alpha_1, \beta_1] = [1, 1]$ and the uncertainty of φ_2 is: (a) $[\alpha_2, \beta_2] = [0, 0]$, (b) $[\alpha_2, \beta_2] = [1, 1]$, (c) $[\alpha_2, \beta_2] = [0, 1]$, and (d) $[\alpha_2, \beta_2] = [0.25, 0.80]$.

  – When the uncertainty of one of the propositions is $[0, 1]$ (i.e., total ignorance), the belief of the formula $\varphi_1 \wedge \varphi_2$ is 0 and its plausibility is bounded by the plausibility of the second proposition.

- The MLE strategy easily extends to more general CFE operations. Its use for the fusion of dichotomous FoDs is straightforward.

3.4.2 Joint Estimation of CFE Coefficients and Unknown Uncertainties in ULP Models

An algorithm for joint estimation of CFE coefficients and unknown proposition uncertainties can be obtained through MM optimization [67]. In the following we describe an algorithm that is based on a particular class of MM optimization methods, namely, the Expectation-Maximization (EM) method. Although not directly aiming at estimating an expected value per se, we refer to this method as an EM algorithm for ULP.

The EM algorithm for ULP is defined by the following two-step iteration:

1. Solve
$$\arg\max_{\Gamma_j} \mathcal{L}_\Phi(\{m_j\}, \Gamma_j)$$
to find estimates of Γ_j.

2. Solve
$$\arg\max_{m_j} \mathcal{L}_\Phi(\{m_j\}, \Gamma_j)$$
to find estimates of uncertainties defined by m_j (and, in particular, defined by $\mathbf{a_j}$ and $\mathbf{b_j}$).

These iterations are repeated until convergence.

The use of the EM method for ULP has a number of advantages for reasoning under ULP. By relying on the definition of the pseudo-likelihood operator $\mathcal{L}$ introduced above, we ensure that the cost function is convex and its convergence can be proven. If nonlinear constraints are not introduced, the solution to the EM folder converges to a global minimum.

Example: Joint estimation of CFE coefficients and unknown uncertainties in Modus Tollens. Consider the following logic model:

$$\Lambda_1: \quad \varphi_1 \implies \varphi_2, \text{ with uncertainty } [\alpha_{F1}, \beta_{F1}]$$
$$\Lambda_2: \quad \varphi_1, \text{with uncertainty } [\alpha_{F2}, \beta_{F2}],$$

where the uncertainty of φ_1 is $[\alpha_1, \beta_1]$ and the uncertainty of φ_2 is $[\alpha_2, \beta_2]$. The pseudo-likelihood function $\mathcal{L}$ corresponding to this model is:

$$\begin{aligned}
\mathcal{L}(\mathbf{a} = [\alpha_{\mathbf{1}}, \alpha_{\mathbf{2}}], \mathbf{b} = [\beta_{\mathbf{1}}, \beta_{\mathbf{2}}], \mathbf{\Gamma}) &= \mathcal{L}_{\mathbf{\Lambda_1}}(\mathbf{a}, \mathbf{b}, \mathbf{\Gamma}) + \mathcal{L}_{\mathbf{\Lambda_2}}(\mathbf{a}, \mathbf{b}, \mathbf{\Gamma}) \\
&= \left(\begin{bmatrix} \gamma_{1,1}^{\{\mathbf{1}\}} & \gamma_{1,2}^{\{\mathbf{1}\}} \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} \gamma_{1,1}^{\{\mathbf{1,0}\}} & \gamma_{1,2}^{\{\mathbf{1,0}\}} \end{bmatrix} \times \begin{bmatrix} \beta_1 - \alpha_1 \\ \beta_2 - \alpha_2 \end{bmatrix} \right) \\
&\quad + \left(\begin{bmatrix} \gamma_{1,1}^{\{\mathbf{0}\}} & \gamma_{1,2}^{\{\mathbf{0}\}} \end{bmatrix} \times \begin{bmatrix} 1 - \beta_1 \\ 1 - \beta_2 \end{bmatrix} + \begin{bmatrix} \gamma_{1,1}^{\{\mathbf{1,0}\}} & \gamma_{1,2}^{\{\mathbf{1,0}\}} \end{bmatrix} \times \begin{bmatrix} \beta_1 - \alpha_1 \\ \beta_2 - \alpha_2 \end{bmatrix} \right) \\
&\quad + \left(\begin{bmatrix} \gamma_{2,1}^{\{\mathbf{1}\}} \end{bmatrix} \times [\alpha_1] + \begin{bmatrix} \gamma_{2,1}^{\{\mathbf{1,0}\}} \end{bmatrix} \times [\beta_1 - \alpha_1] \right) \\
&\quad + \left(\begin{bmatrix} \gamma_{2,1}^{\{\mathbf{0}\}} \end{bmatrix} \times [1 - \beta_1] + \begin{bmatrix} \gamma_{2,1}^{\{\mathbf{1,0}\}} \end{bmatrix} \times [\beta_1 - \alpha_1] \right),
\end{aligned}$$

where $\Gamma = \{\Gamma_{\Lambda_1}, \Gamma_{\Lambda_2}\}$, $\Gamma_{\Lambda_1} = \{\gamma_{1,1}^{\{\mathbf{1}\}}, \gamma_{1,2}^{\{\mathbf{1}\}}, \gamma_{1,1}^{\{\mathbf{0}\}}, \gamma_{1,2}^{\{\mathbf{0}\}}, \gamma_{1,1}^{\{\mathbf{1,0}\}}, \gamma_{1,2}^{\{\mathbf{1,0}\}}\}$, and $\Gamma_{\Lambda_2} = \{\gamma_{2,1}^{\{\mathbf{1}\}}, \gamma_{2,1}^{\{\mathbf{0}\}}, \gamma_{2,1}^{\{\mathbf{1,0}\}}\}$.

Now assume that only the uncertainty interval $[\alpha_1, \beta_1]$ is known. We can jointly estimate $[\alpha_{F1}, \beta_{F1}]$, the uncertainty of the implication rule $\varphi_1 \implies \varphi_2$, and the uncertainty $[\alpha_2, \beta_2]$ of proposition φ_2 using the EM algorithm for ULP. In this case, the maximization problems associated to the EM algorithm must be properly constrained to account for the expected behavior of uncertainty intervals (e.g., $0 \leq \alpha_i \leq \beta_i \leq 1, \forall i = 1, 2$), CFE coefficients (e.g., the sum of all coefficients Γ_{Λ_i} must be equal to 1, for $i = 1, 2$), and logic rules.

Figure 3.3 illustrates the uncertainty intervals delivered by the EM algorithm when $[\alpha_1, \beta_1] = [0, 1]$. In this scenario, we are interested on reasoning when we have complete ignorance regarding the truth value of φ_1. As expected, the figure shows that, in this case, we cannot infer anything else regarding the truth value of φ_2 (because the resulting uncertainty interval is $[0, 1]$). Similarly, we cannot infer anything about the truth value of the implication rule. The uncertainty interval $[\alpha_{F1}, \beta_{F1}]$ is symmetric around 0.5. We can also see in the figure how the algorithm converges to a global maximum. However, there is no guarantee that this maximum is unique.

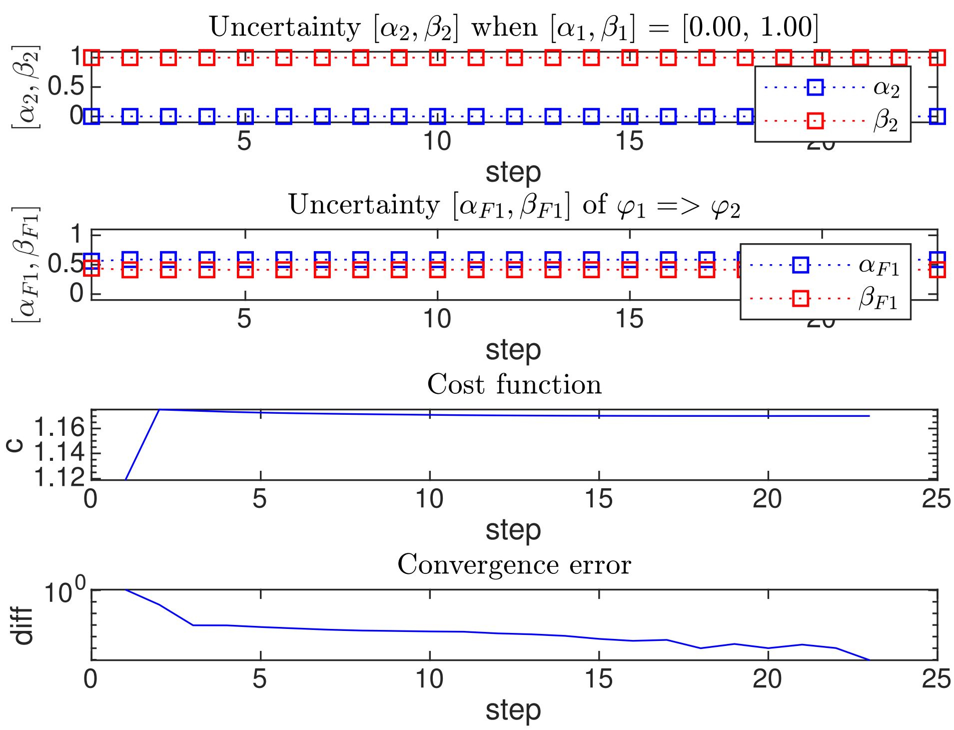


Figure 3.3: Uncertainty $[\alpha_2, \beta_2]$ of proposition φ_2 obtained when there is total ignorance regarding the truth value of the antecedent φ_1 in a logic rule $\varphi_1 \implies \varphi_2$. In this case, there is also total ignorance regarding the truth value of the consequent φ_2. The figure also shows the convergence of the EM algorithm in this case.

Figure 3.3 contains the uncertainty intervals delivered by the EM algorithm when $[\alpha_1, \beta_1] = [0, 0]$. In this case, we know that proposition φ_1 is false. In classical logic, we know that if the antecedent of an implication rule is false, then the consequent could be either true or false. This is replicated by the results shown in the figure, as the uncertainty interval for the consequent is $[0, 1]$.

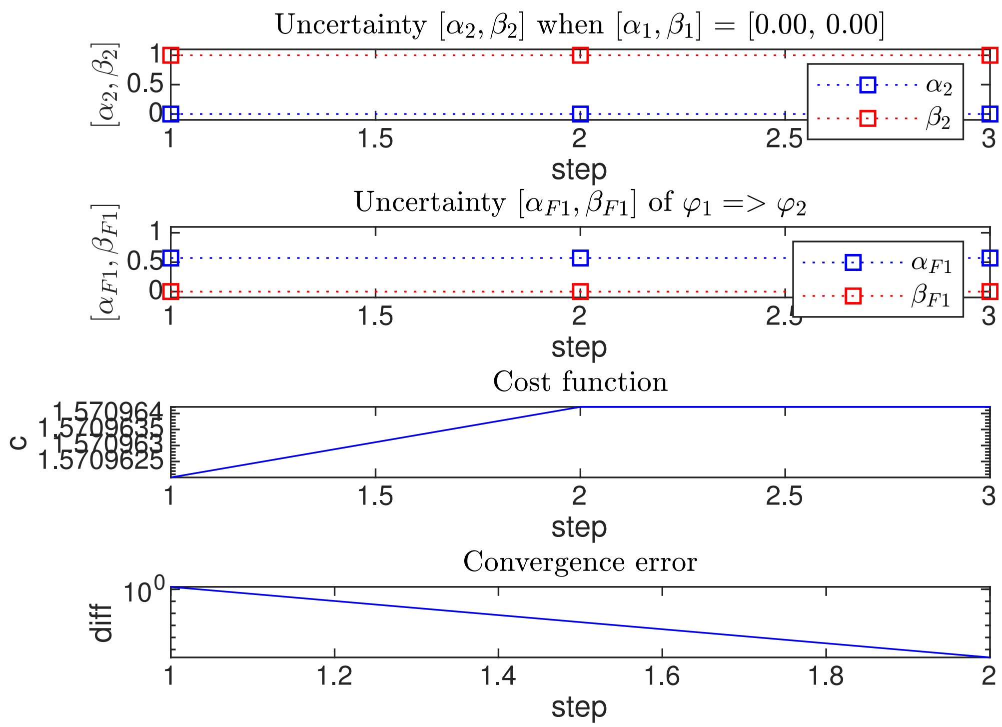


Figure 3.4: Uncertainty $[\alpha_2, \beta_2]$ of proposition φ_2 obtained when the uncertainty of φ_1 is $[0, 0]$ in a logic rule $\varphi_1 \implies \varphi_2$. In this case, and consistent with classical logic results, there is total ignorance regarding the truth value of the consequent φ_2.

A more general scenario is illustrated in Figure 3.5, which depicts the uncertainty of φ_2 when different uncertainties of φ_1 are evaluated. Although this scenario considers a particular set of constraints that do not aim at delivering consistency with classical logic, we can see that the uncertainty of φ_2 goes to $[1, 1]$ as the uncertainty of the antecedent φ_1 goes to $[1, 1]$. Indeed, only when the uncertainty of φ_1 gets close to [1, 1] the uncertainty of the consequent φ_2 decreases, and the support to φ_2 being true significantly increases.

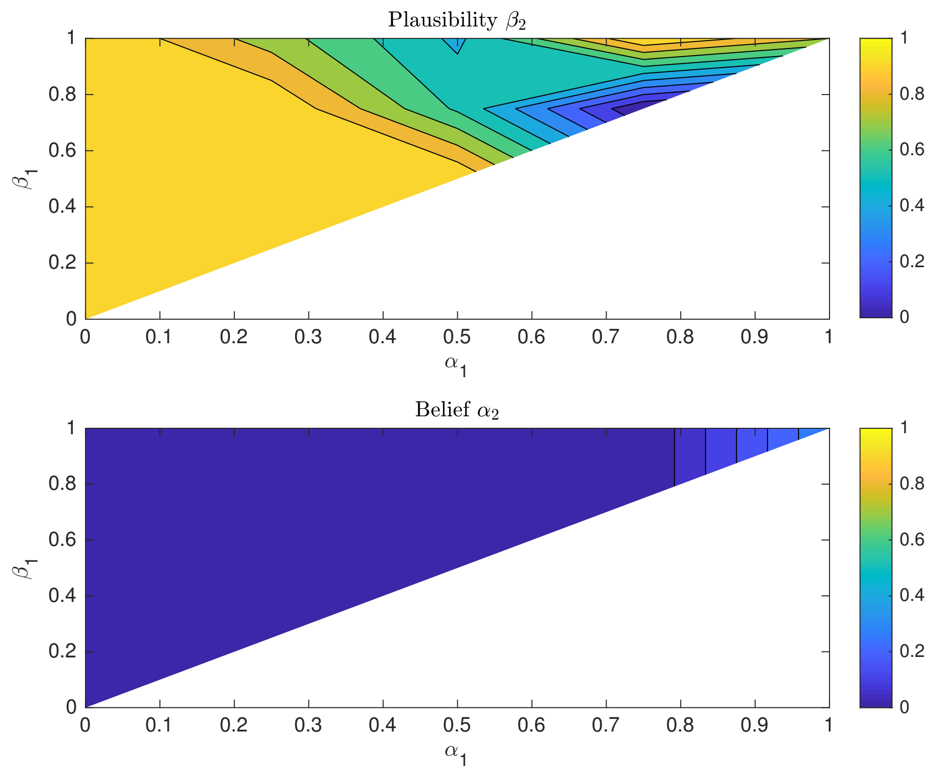


Figure 3.5: Uncertainty $[\alpha_2, \beta_2]$ of proposition φ_2 obtained with different uncertainties of the antecedent φ_1. In this case, we can see how the uncertainty of φ_2 decreases (and goes to $[1, 1]$) as the uncertainty interval of φ_1 goes to $[1, 1]$.

CHAPTER 4

Hard and Soft Data Fusion

4.1 General Filtering and Estimation Framework

4.1.1 The Filtering (and Prediction) Problem

Consider a generic (discrete-time) stochastic filtering problem in a dynamic state space form [1]:

$$\mathbf{x}_n = \mathbf{f}_{n-1}(\mathbf{x}_{n-1}, \mathbf{v}_{n-1}), \tag{4.1a}$$

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n, \mathbf{w}_n), \tag{4.1b}$$

where (4.1a) and (4.1b) are called state and measurement equations, respectively; the index $n \in \mathbb{N}$ is assigned to a continuous-time instant t_n; $\mathbf{x}_n \in \mathbb{R}^{N_\mathbf{x}}$ represents the state vector at time n, $\mathbf{z}_n \in \mathbb{R}^{N_\mathbf{z}}$ is the measurement vector at time n, $\mathbf{v}_n$ and $\mathbf{w}_n$ are the process and measurement noise, respectively, and $\mathbf{f} : \mathbb{R}^{N_\mathbf{x}} \mapsto \mathbb{R}^{N_\mathbf{x}}$ and $\mathbf{h} : \mathbb{R}^{N_\mathbf{x}} \mapsto \mathbb{R}^{N_\mathbf{z}}$ are possibly nonlinear functions. The initial state is assumed to have a known probability density function (pdf) $p(\mathbf{x}_0)$ and also to be independent of noise sequences. The state equation (4.1a) characterizes the transition probability $p(\mathbf{x}_n|\mathbf{x}_{n-1})$, and the measurement equation (4.1b) characterizes the probability $p(\mathbf{z}_n|\mathbf{x}_n)$.

The objective is to find estimates of $\mathbf{x}_n$ based on the sequence of all available measurements $\mathbf{Z}_n \triangleq \{\mathbf{z}_i, i = 1, 2, \cdots, n\}$. These estimates can be obtained through a posterior pdf $p(\mathbf{x}_n|\mathbf{Z}_n)$, which, in a Bayesian framework, may be obtained sequentially in two stages: prediction and update.

Defining $\mathbf{z}_0$ as the no measurement set, and assuming that the initial density of the state vector $p(\mathbf{x}_0|\mathbf{z}_0)$ and the pdf $p(\mathbf{x}_{n-1}|\mathbf{Z}_{n-1})$ are known, the prediction state at time n is given by [1]:

$$p(\mathbf{x}_n|\mathbf{Z}_{n-1}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})\ p(\mathbf{x}_{n-1}|\mathbf{Z}_{n-1})\ d\mathbf{x}_{n-1}. \tag{4.2}$$

The fact that (4.1a) describes a Markov process of order one is used to make $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{Z}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$ in the derivation of (4.2).

At time n the measurement $\mathbf{z}_n$ becomes available and the update can be obtained via the Bayes rule as:

$$p(\mathbf{x}_n|\mathbf{Z}_n) = \frac{p(\mathbf{z}_n|\mathbf{x}_n)\quad p(\mathbf{x}_n|\mathbf{Z}_{n-1})}{p(\mathbf{z}_n|\mathbf{Z}_{n-1})}, \tag{4.3}$$

with

$$p(\mathbf{z}_n|\mathbf{Z}_{n-1}) = \int p(\mathbf{z}_n|\mathbf{x}_n)\ p(\mathbf{x}_n|\mathbf{Z}_{n-1})\ d\mathbf{x}_n. \tag{4.4}$$

Note that $p(\mathbf{z}_n|\mathbf{x}_n)$ in (4.4) depends on the measurement model (4.1b).

Knowledge of the posterior pdf $p(\mathbf{x}_n|\mathbf{Z}_n)$ allows us to compute state estimates with respect to any criterion. For example, the minimum mean square error estimate $\hat{x}_{n|n}^{\mathrm{MMSE}} = E\{\mathbf{x}_n|\mathbf{Z}_n\}$, or the maximum a posteriori estimate $\hat{x}_{n|n}^{\mathrm{MAP}} = \arg\max_{\mathbf{x}_n} p(\mathbf{x}_n|\mathbf{Z}_n)$.

4.1.2 Incorporating Hard and Soft Information

When considering fusion of hard and soft data, it is useful to clearly differentiate hard from soft data (measurements), as well as hard state from "soft" state variables. We call *hard data* any measurement provided by a physical sensor, for example, video, radar, acoustic readings, etc. *Soft data* is any measurement that is obtained from a human source, for example, text from blogs, witness reports, etc. *Hard states* consist of traditional (discrete/continuous) state variables (e.g., position, velocity), while *soft states* consist of more qualitative variables of interest, such as mood and opinions.

We then define: $\mathbf{x}_n = \{\mathbf{x}_n^h, \mathbf{x}_n^s\}$ and $\mathbf{z}_n = \{\mathbf{z}_n^h, \mathbf{z}_n^s\}$, where $\mathbf{x}_n^h$ represents the vector of "hard" states at time n, $\mathbf{x}_n^s$ the vector of "soft" states, $\mathbf{z}_n^h$ the vector of hard (i.e., physical) measurements, and $\mathbf{z}_n^s$ is the vector of soft measurements or observations (in traditional systems, $\mathbf{z}_n = \mathbf{z}_n^h$ and $\mathbf{x}_n = \mathbf{x}_n^h$).

In order to combine soft and hard data models, we propose to represent the joint pdf $p(\mathbf{x}_n, \mathbf{Z}_n)$ as a normalized product of three (potentially decomposable) combining functions $\Psi_h(\cdot)$, $\Psi_{h,s}(\cdot)$, and $\Psi_s(\cdot)$ as follows:

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{Z}_n) &= p(\mathbf{x}_n^h, \mathbf{x}_n^s, \mathbf{Z}_n^h, \mathbf{Z}_n^s) \\ &= \tfrac{1}{K}\ \Psi_h(\mathbf{x}_n^h, \mathbf{Z}_n^h)\ \Psi_{h,s}(\mathbf{x}_n^h, \mathbf{x}_n^s, \mathbf{Z}_n^h, \mathbf{Z}_n^s)\ \Psi_s(\mathbf{x}_n^s, \mathbf{Z}_n^s), \end{aligned} \tag{4.5}$$

where $\Psi_h(\cdot)$ models the interactions between hard data and hard states, $\Psi_s(\cdot)$ models the interaction between soft data and soft states, $\Psi_{h,s}(\cdot)$ is a hybrid function that represents the interactions between some hard and soft variables, and K is a normalization factor. The posterior pdf $p(\mathbf{x}_n|\mathbf{Z}_n)$ could then be obtained using (4.5) as:

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{Z}_n) &= p(\mathbf{x}_n, \mathbf{Z}_n)/p(\mathbf{Z}_n) \\ &= \tfrac{1}{K'}\ \ \Psi_h(\mathbf{x}_n^h, \mathbf{Z}_n^h)\ \ \Psi_{h,s}(\mathbf{x}_n^h, \mathbf{x}_n^s, \mathbf{Z}_n^h, \mathbf{Z}_n^s)\ \ \Psi_s(\mathbf{x}_n^s, \mathbf{Z}_n^s), \end{aligned}$$

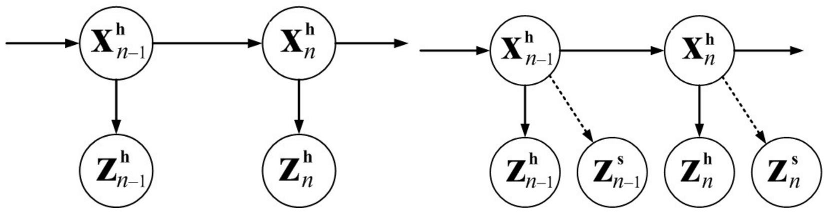
where $K' = K\ p(\mathbf{Z}_n)$.

The advantage of this partition of the joint probability function is that we can combine different models –the complexity of each of the individual models may be significantly lower than the complexity of a "global" model. This combination enables enhancing existing hard-data probabilistic models with soft data methods.

4.2 Incorporating Soft Sources

Let us consider a Bayesian filter, initially designed considering only hard data and hard states, now enhanced with "soft" observations $\mathbf{Z}_n^s$. If only hard data is available, the Bayesian processor (4.3) becomes:

$$p(\mathbf{x}_n^h|\mathbf{Z}_n^h) = \frac{p(\mathbf{z}_n^h|\mathbf{x}_n^h)\quad p(\mathbf{x}_n^h|\mathbf{Z}_{n-1}^h)}{p(\mathbf{z}_n^h|\mathbf{Z}_{n-1}^h)}. \tag{4.6}$$



(a) (b)

Figure 4.1: A graphical model of a generic Bayesian processor with only hard states and (a) hard-data observations; (b) hard and soft-data observations. The new edges enabled by the soft observations are shown as dashed lines.

This process can be described by a graphical model, as shown in Figure 4.1(a). When we incorporate a soft measurement vector, we are adding a node to this graph, rendering the model in Figure 4.1(b).

Let us consider how this changes when we add soft-data measurements $\mathbf{Z}_n^s$. We obtain:

$$p(\mathbf{x}_n^h|\mathbf{Z}_n^h, \mathbf{Z}_n^s) = \frac{p(\mathbf{z}_n^h, \mathbf{z}_n^s|\mathbf{x}_n^h) \quad p(\mathbf{x}_n^h|\mathbf{Z}_{n-1}^h, \mathbf{Z}_{n-1}^s)}{p(\mathbf{z}_n^h, \mathbf{z}_n^s|\mathbf{Z}_{n-1}^h, \mathbf{Z}_{n-1}^s)}. \tag{4.7}$$

Given that $\mathbf{z}_n^h$ is conditionally independent of $\mathbf{z}_n^s$ given $\mathbf{x}_n^h$ (i.e., $\mathbf{z}_n^h \perp \mathbf{z}_n^s|\mathbf{x}_n^h$), we can partition $p(\mathbf{z}_n^h, \mathbf{z}_n^s|\mathbf{x}_n^h)$ in (4.7) [68]. It can be shown that (4.7) becomes:

$$\begin{aligned} &p(\mathbf{x}_n^h|\mathbf{Z}_n^h, \mathbf{Z}_n^s) \\ &\quad = \frac{p(\mathbf{z}_n^h|\mathbf{x}_n^h)p(\mathbf{x}_n^h|\mathbf{Z}_{n-1}^h)}{p(\mathbf{z}_n^h|\mathbf{Z}_{n-1}^h)} \cdot \frac{p(\mathbf{z}_n^s|\mathbf{x}_n^h)p(\mathbf{Z}_{n-1}^s|\mathbf{x}_n^h, \mathbf{Z}_{n-1}^h)}{p(\mathbf{z}_n^s, \mathbf{Z}_{n-1}^s|\mathbf{z}_n^h, \mathbf{Z}_{n-1}^h)}. \end{aligned}$$

Defining: $\Psi_h(\cdot) = \big(\ p(\mathbf{z}_n^h|\mathbf{x}_n^h)\ p(\mathbf{x}_n^h|\mathbf{Z}_{n-1}^h)\ \big)/(p(\mathbf{z}_n^h|\mathbf{Z}_{n-1}^h))$,
$\Psi_{h,s}(\cdot) = p(\mathbf{z}_n^s|\mathbf{x}_n^h)p(\mathbf{Z}_{n-1}^s|\mathbf{x}_n^h, \mathbf{Z}_{n-1}^h)p(\mathbf{Z}_n^h)$, and $K = 1$,

$$p(\mathbf{x}_n^h, \mathbf{Z}_n^h, \mathbf{Z}_n^s) \quad = \quad \tfrac{1}{K}\Psi_h(\mathbf{x}_n^h, \mathbf{Z}_n^h) \ \ \Psi_{h,s}(\mathbf{x}_n^h, \mathbf{Z}_n^h, \mathbf{Z}_n^s), \tag{4.8}$$

which has the form of (4.5) (with $\Psi_s(\cdot) = 1$). Note that, in this case, $\Psi_h(\mathbf{x}_n^h, \mathbf{Z}_n^h)$ is what traditional Bayesian processors compute, i.e., it corresponds to the model in (4.6). For many applications, a number of solutions already exist to define this function. The new potential function $\Psi_{h,s}(\mathbf{x}_n^h, \mathbf{Z}_n^h, \mathbf{Z}_n^s)$ could be designed based on the application or the data that is being received and incorporated as an enhancement to the hard-data-only model. Through similar arguments, we can obtain the complete state posterior $p(\mathbf{x}_n|\mathbf{Z}_n^h, \mathbf{Z}_n^s)$. What is attractive about this decomposition is that one has freedom to independently design and tailor the combining functions according to different application scenarios, perhaps even using the flexibility of graphical models to define the combining functions as products of generalized exponential potential functions. An alternative for defining combining functions for soft data that is based on FOL representations and inference is described next.

4.3 Designing Combining Functions using First-Order Logic Models of Soft Data

In this section we introduce a method for modeling probability in the soft-and-hard environment based on DS uncertain logic inference. It is worth noting that this is not the only method for designing combining functions based on FOL. Other methods could be used (e.g., Hybrid Markov Logic Networks [34] in probabilistic reasoning) or pursued in subsequent research.

Recall that uncertain logic inference models render results that are expressed through BBAs. Let us call $m_j(\cdot), j = 1, 2, \ldots,$ the set of BBAs that model uncertain relations among hybrid hard and soft data/states, and $m_k(\cdot), k = 1, 2, \ldots,$ the set of BBAs that model relations among soft data and soft states. Then, we can define the corresponding combining functions as:

$$\begin{aligned} \Psi_{h,s}(\mathbf{x}_n^h, \mathbf{x}_n^s, \mathbf{Z}_n^h, \mathbf{Z}_n^s) &= \mathrm{BetP}(\oplus_j m_j(\mathbf{x}_n^h, \mathbf{x}_n^s, \mathbf{Z}_n^h, \mathbf{Z}_n^s)); \text{ and} \\ \Psi_s(\mathbf{x}_n^s, \mathbf{Z}_n^s) &= \mathrm{BetP}\left(\oplus_k m_k(\mathbf{x}_n^s, \mathbf{Z}_n^s)\right), \end{aligned} \tag{4.9}$$

where $\oplus$ is an appropriate fusion operator, and $\mathrm{BetP}(\cdot)$ is a transformation from the DS-domain to the probability domain defined as [49]: $\mathrm{BetP}(x) = \sum_{A \subseteq \Theta; x \in A} \frac{m(A)}{|A|}$, with $|A|$ denoting the cardinality of the set A. This transformation is called the pignistic probability function. Other transformations could also be used (e.g., plausibility transformation [52]). It is also possible to replace the transformation function with the belief or plausibility corresponding to the fused BBAs, and obtain lower and upper bounds of the probability distributions that result from the combining functions. As an example on how to incorporate these combining functions in the general framework, consider the model (4.5). Then, using (4.9) we obtain:

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{Z}_n) = \tfrac{1}{K} \quad & p_{\mathrm{B}}(\mathbf{x}_n^h, \mathbf{Z}_n^h) \\ & \cdot \operatorname{BetP}(\oplus_j m_j(\mathbf{x}_n^h, \mathbf{x}_n^s, \mathbf{Z}_n^h, \mathbf{Z}_n^s)) \cdot \operatorname{BetP}\left(\oplus_k m_k(\mathbf{x}_n^s, \mathbf{Z}_n^s)\right), \end{aligned}$$

where the hard data is processed by a conventional Bayesian processor that renders $p_{\mathrm{B}} = \Psi_h(\mathbf{x}_n^h, \mathbf{Z}_n^h)$, j is an index of independent hybrid mass functions, and k is an index of the independent "soft" mass functions.

CHAPTER 5

Application Examples

5.1 Human-Robot Interaction

In this section we show an application on human-robot interaction. Consider a human H giving an implicit instruction to robot R with uncertainty boundaries $[\alpha_1, \beta_1]$. This interval reflects the degree to which R believes that H's statement is true. Note that this uncertainty may change depending on the parsing process, as well as on the actual instruction provided by H. For example, in a real-life scenario, H may use words (or indicate rules to R) that entail uncertainty, such as *usually*, *typically*, or *generally*.

In this example, the instruction provided by H is: "Commander Z really needs a medkit". The robot then runs an inference process in which R needs to find if it needs to get the medkit for Z or not. If the conclusion is very precise (i.e., low uncertainty), then R could simply execute the required action, which could be either get the medkit for Z or not. However, if the conclusion is highly ambiguous, then the robot R could respond "Should I get it for him?" and solve the ambiguity problem.

An inference process to solve R's problem could be as follows. Suppose that the following rules were given (in natural language) to R:

1. "if x needs y, then x has a goal to have y", with uncertainty $[\alpha_2, \beta_2]$;

2. "Commander Z is likely of higher rank than robot R", with uncertainty $[\alpha_3, \beta_3]$;

3. "*usually*, if x is of higher rank than y and x has goal g, then y should have the goal for x to have goal g, with uncertainty $[\alpha_4, \beta_4]$; and

4. "if the robot has a goal for x to have a goal to have y, then the robot *should have* the goal to get y for x", with uncertainty $[\alpha_5, \beta_5]$.

Note that all these expressions entail some uncertainty, which may be due uncertain information already known to the robot, or by imprecise words that were used to describe the instructions, such as "likely", "usually", and "should have".

These instructions can be expressed in first-order logic as is shown in rows 1 to 5 of Table 5.1. These rows represent the premises of our inference process. Based on these premises, an inference process could continue as shown in rows 6 to 9 of Table 5.1. For simplicity, we assume that the required inference rules carry over from classical logic to the uncertainty case, an assumption that is true in the case of CFE-based uncertain logic.

Table 5.1: Inference process for the human-robot interaction problem of the case study (M = medkit).

	Logic Formula	Premises	Uncertainty
1	$\texttt{Needs}(\mathrm{Z}, \mathrm{M})$	Δ	$[\alpha_1, \beta_1]$
2	$\forall x\ \forall y:\ \texttt{Needs}(x, y)$		
	$\implies \texttt{Goal}(x, \texttt{Have}(x, y))$	Δ	$[\alpha_2, \beta_2]$
3	$\texttt{Rank}(\mathrm{Z}) > \texttt{Rank}(\mathrm{R})$	Δ	$[\alpha_3, \beta_3]$
4	$\forall x\ \forall y\ \forall g:\ \texttt{Rank}(x) > \texttt{Rank}(y)\big)$		
	$\wedge \texttt{Goal}(x, g) \implies \texttt{Goal}(y, \texttt{Goal}(x, g)$	Δ	$[\alpha_4, \beta_4]$
5	$\forall x\ \forall y:\ \texttt{Goal}\big(\mathrm{R}, \texttt{Goal}(x, \texttt{Have}(x, y))\big)$		
	$\implies \texttt{GetFor}(\mathrm{R}, y, x)$	Δ	$[\alpha_5, \beta_5]$
6	$\texttt{Goal}(\mathrm{Z}, \texttt{Have}(\mathrm{Z}, \mathrm{M}))$	1, 2, MP	$[\alpha_6, \beta_6]$
7	$\texttt{Rank}(\mathrm{Z}) > \texttt{Rank}(\mathrm{R}) \wedge \texttt{Goal}(\mathrm{Z}, \texttt{Have}(\mathrm{Z}, \mathrm{M}))$	3, 6, AI	$[\alpha_7, \beta_7]$
8	$\texttt{Goal}\big(\mathrm{R}, \texttt{Goal}(\mathrm{Z}, \texttt{Have}(\mathrm{Z}, \mathrm{M}))\big)$	4, 7, MP	$[\alpha_8, \beta_8]$
9	$\texttt{GetFor}(\mathrm{R}, \mathrm{M}, \mathrm{Z})$	5, 8, MP	$[\alpha_9, \beta_9]$

In order to better understand how the uncertainty propagates in this example, we analyze four cases: 1) Perfect scenario (i.e., no uncertainty); 2) Probabilistic scenario (i.e., $\alpha_i = \beta_i, i = 1, 2, \ldots, 5$); 3) Probabilistic scenario with insufficient evidence; and 4) General scenario.

5.1.1 Perfect Scenario (i.e., No Uncertainty)

Figure 5.1 illustrates a scenario where all the premises and rules are taken as truth. This is, the uncertainty intervals $[\alpha_i, \beta_i], i = 1, 2, \ldots, 5$, for the premises A1 to A5 is $[1, 1]$. In this case, the uncertainty of every step in our reasoning process is defined by the uncertainty interval $[1, 1]$, which is consistent with classical logic results. Note that the result is the same for both of the models analyzed, namely, the CFE-based inference model (top), and the DCR-based inference model (center). The figure also shows, at the bottom, the output of our ambiguity measure λ, which, as expected, remains at 1 throughout all the inference process.

5.1.2 Probabilistic Scenario

Figure 5.2 illustrates a scenario where all the premises and rules are probabilistic. That is, the uncertainty intervals $[\alpha_i, \beta_i], i = 1, 2, \ldots, 5$, for the premises A1 to A5 are characterized by $\alpha_i = \beta_i$. In this case, the CFE-based model (top) maintains the probabilistic behavior throughout the inference process. The DCR-based model (center), on the contrary, departs from the probabilistic model and, in this case, the uncertainty increases as the inference process progresses. This is also seen in the ambiguity measure (bottom), as it is always decreasing for the DCR-based inference (becoming more ambiguous).

5.1.3 Probabilistic Scenario with Insufficient Evidence

Figure 5.3 illustrates a scenario where all the premises and rules are probabilistic, but in which the support for the premises makes the inference process stop after a certain number of steps. Note that, as expected, CFE-based inference (top) stops rendering results when the evidence is not enough to provide a conclusion. DCR-

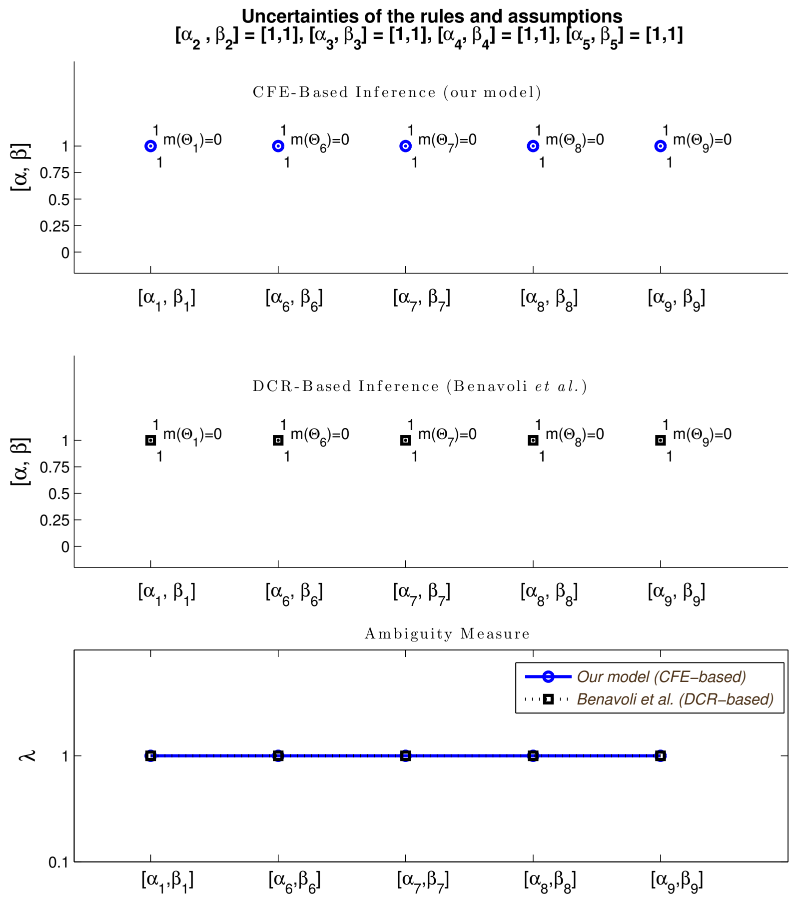


Figure 5.1: DS-based uncertain logic inference on a perfect (i.e., no uncertainty) scenario. In this case, there is no uncertainty on any of the inferred premises.

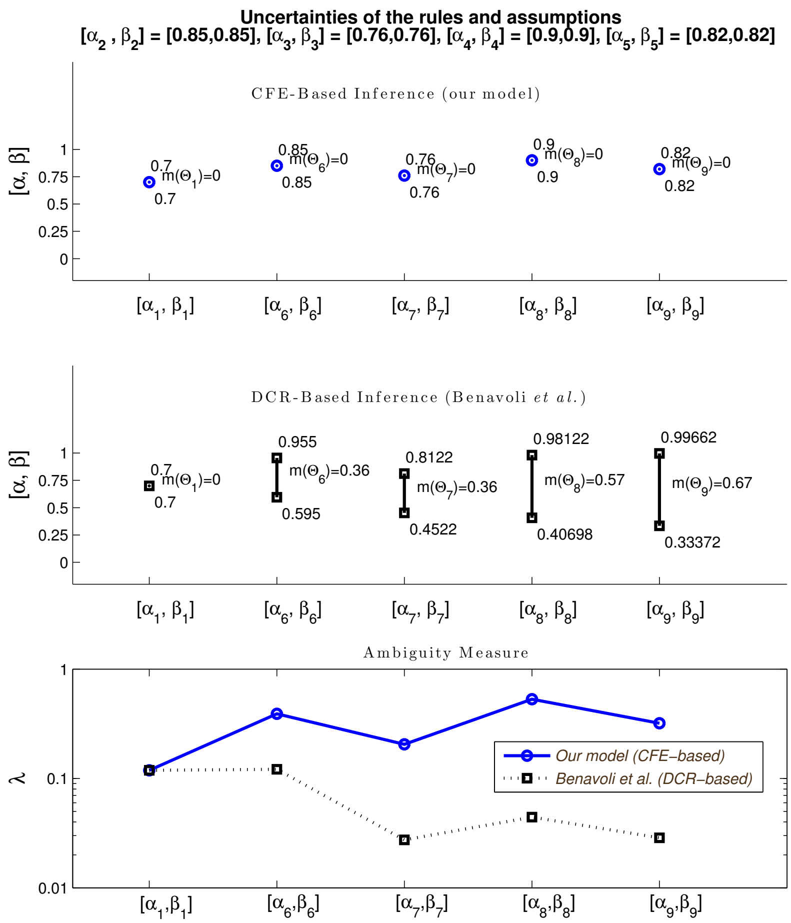


Figure 5.2: DS-based uncertain logic inference on a probabilistic scenario. In this case, CFE-based inference maintains the probabilistic behavior throughout the inference process, while DCR-based inference cannot maintain it. The uncertainty of the DCR-based inference increases as the inference process progresses.

based inference (center), on the contrary, keeps delivering results without indicating the risk of making a decision based on the output results. Thus, the ambiguity measure could be incorporated into decision-making processes.

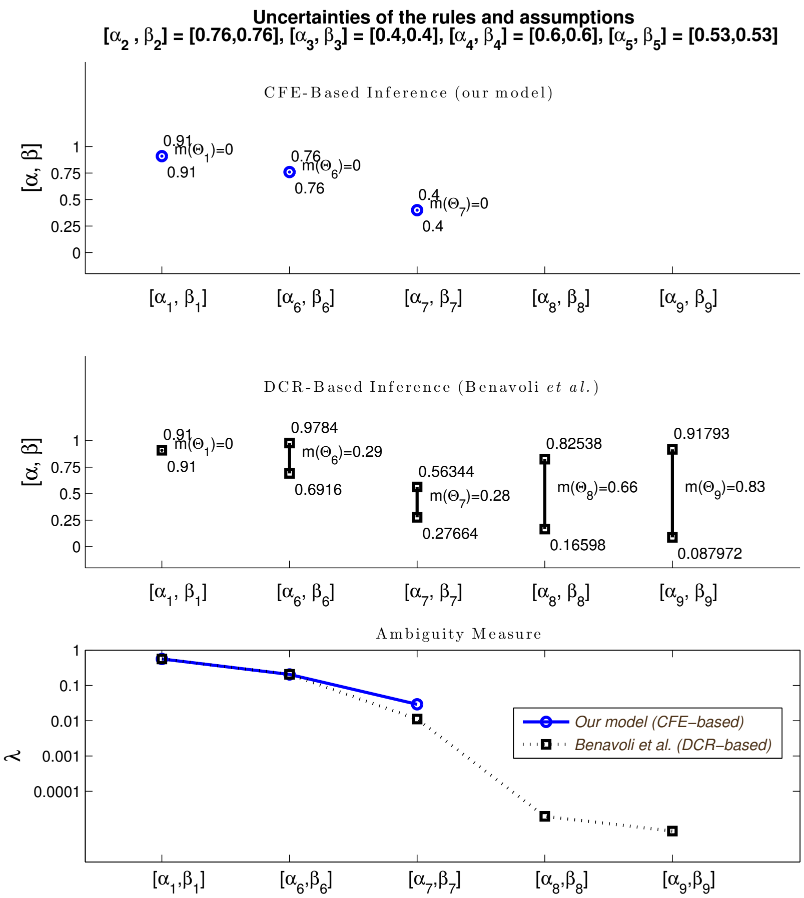


Figure 5.3: DS-based uncertain logic inference on a probabilistic scenario with insufficient evidence. In this case, CFE-based inference stops providing results when there is not enough evidence to support them. DCR-based inference, on the contrary, keeps providing results without any indication about the risk of making a decision based on these results.

5.1.4 General-case Scenario

Figure 5.4 illustrates a general-case scenario where there are no restrictions regarding the uncertainty intervals $[\alpha_i, \beta_i], i = 1, 2, \ldots, 5$, for the premises A1 to A5. In this case, the CFE-based model (top) renders more "certain" results than the DCR-based model (center), as evidenced from the ambiguity measure (bottom).

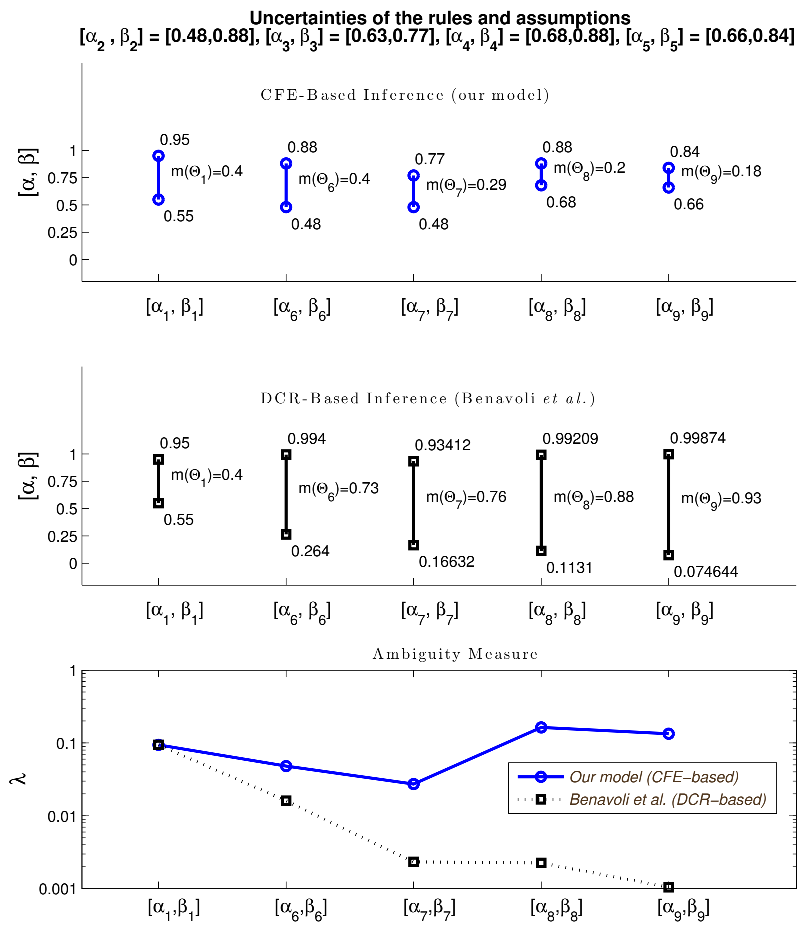


Figure 5.4: DS-based inference on a general-case scenario. In this case, although both CFE and DCR-based inference provide uncertain results, the ambiguity measure on the resulting BBAs is higher for the CFE-based inference, as indicated by λ.

5.2 Tracking with Human Observers

Now let us consider an example in which we augment the terrain-aided tracking example in [1] with soft states and soft measurements. The scenario used for the example is shown in Figure 5.5. This figure represents a road map with four roads (AJ, BJ, CJ, and DJ) meeting at J. The road segments represented by solid lines allow entry to or exit from the roads. The road segments shown by broken lines (TU and BJ) indicate that targets on the road cannot get off or those off the road cannot get on the road.

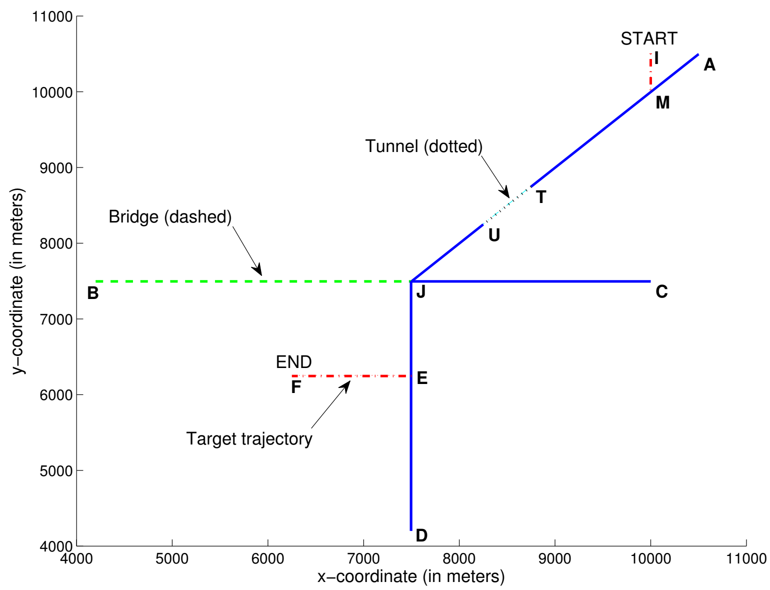


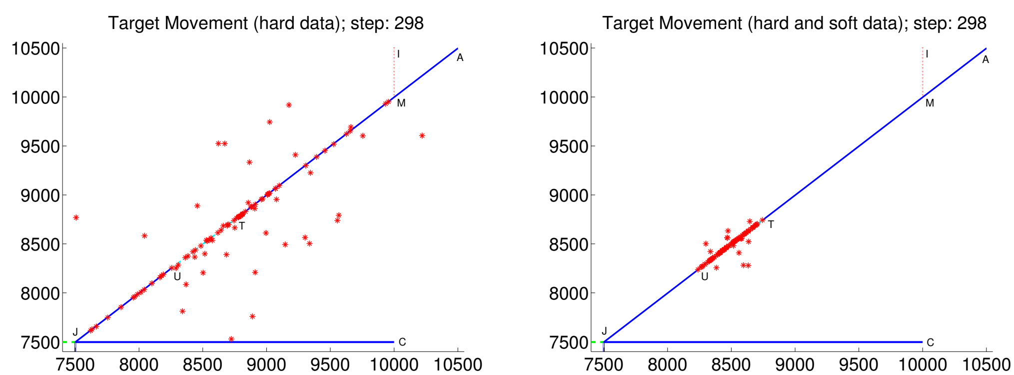
Figure 5.5: An example of a road network with a target trajectory, as introduced in [1].

A (hard-data) solution for tracking in this type of scenario is the Variable Structure – Multiple Model Particle Filter (VS-MMPF) [1], with a radar system located at the origin. This filter outputs a joint probability distribution $p_{\text{VS-MMPF}}(\mathbf{x}_n^h, \mathbf{Z}_n^h)$. When the target enters the tunnel, the radar is unable to receive measurements, and the accuracy of this filter is compromised (see Figure 5.6(a)). This system can be enhanced with a human observer located inside the tunnel. In this case, we could either redesign the VS-MMPF with the new "human sensor", or we could use the method introduced in Chapter 4 to easily enhance the existing radar system.

For the latter case, suppose we know a probability mass function $p(\mathbf{x}^h, \mathbf{z}^s)$ for the accuracy of the "human sensor", which is in Table 5.2. Then, we can combine the information in this table with the original VS-MMPF by subsituting $\Psi_h(\mathbf{x}_n^h, \mathbf{Z}_n^h) = p_{\text{VS-MMPF}}(\mathbf{x}_n^h, \mathbf{Z}_n^h)$ and $\Psi_{h,s}(\mathbf{x}_n^h, \mathbf{Z}_n^h, , \mathbf{Z}_n^s) = p(\mathbf{x}^h, \mathbf{z}^s)$ into (4.8). We then obtain a model for the joint probability $p(\mathbf{x}^h, \mathbf{z}^h, \mathbf{z}^s)$, which generates an enhanced tracking system (see Figures 5.6(b) and 5.7). Although this performance enhancement is expected due to the addition of a new sensor, the method used to obtain such an enhancement is simple and makes use of existing components.

Table 5.2: $p(\mathbf{x}^h, \mathbf{z}^s)$ for a human observer seeing a target.

	Target is in UT	Target is not in UT
Human sees the target	0.45	0.05
Human does not see it	0.05	0.45



(a) (b)

Figure 5.6: 100 predicted particles in step 298 (target in tunnel) using: (a) hard measurements, and (b) both hard and soft measurements.

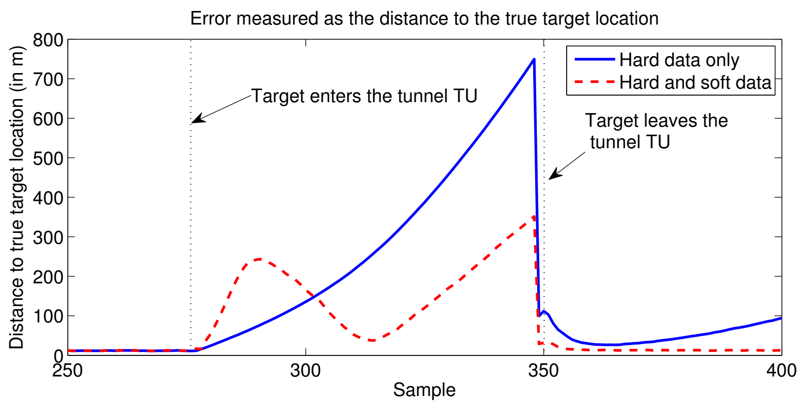


Figure 5.7: Distance from true target to estimated location (500 trials).

5.3 Joint Tracking and Intent Detection

5.3.1 Expert Knowledge for Intent Detection

Based on the same scenario used in 5.2, consider the case where an expert provides soft information to estimate an additional (hard) state x_d, the intended destination. Our state vector then becomes $\mathbf{x}_n^{h*} = \{\mathbf{x}_n^h; x_d\}$, where $\mathbf{x}_n^h$ is the state vector of the example in 5.2. The expert provides the following rules: (R_1): "If the target is moving North on AJ then the destination is A", and (R_2): "If the target is moving South on AJ then the destination is J". Furthermore, the expert indicates that each of the rules is true, with uncertainty being between $70\% - 90\%$. A possible model for R_1 and R_2 in FOL, as well as the corresponding uncertainty parameters is in Table 5.3, where clauses 1 and 2 reproduce the information given by R_1, and clauses 3 and 4 the information given by R_2. Clauses 1 and 3 are incorporated to define a simple (order 1) direction estimator. If the target is going North then $r_1 = 1$. If target is going South then $r_2 = 1$. The methods described in Section 4.3-B allows one to do inference based on the propositions and uncertainty parameters of Table 5.3, which renders a BBA $m(x_d, \mathbf{Z}_n^h)$ that models the relations among hard data and intended destination.

A probabilistic model for this scenario requires that we redefine the combining function $\Psi_h(\mathbf{x}_n^h, \mathbf{Z}_n^h)$ used in 5.2, to account for the new state and its corre-

Table 5.3: FOL Model and uncertainty parameters for the estimation of the state x_d in 5.3.1.

i	FOL Clause	Uncertainty
1	$(x_{2\ \{n\}} > x_{2\ \{n-1\}}) \implies (r_1 = \mathbf{1})$	$[1.0, 1.0]$
2	$((r_1 = \mathbf{1}) \wedge (x \in \text{AJ})) \implies (d = \text{A})$	$[0.7, 0.9]$
3	$(x_{2\ \{n\}} < x_{2\ \{n-1\}}) \implies (r_2 = \mathbf{1})$	$[1.0, 1.0]$
4	$((r_2 = \mathbf{1}) \wedge (x \in \text{AJ})) \implies (d = \text{J})$	$[0.7, 0.9]$

sponding inference method. In this case, the combining function can be defined as $\Psi_h(\mathbf{x}_n^{h*}, \mathbf{Z}_n^h) = \Psi_1(\mathbf{x}_n^h, \mathbf{Z}_n^h) \cdot \Psi_2(\mathbf{x}_n^{h*}, \mathbf{Z}_n^h)$, with $\Psi_1(\mathbf{x}_n^h, \mathbf{Z}_n^h) = p_{\text{VS-MMPF}}(\mathbf{x}_n^h, \mathbf{Z}_n^h)$ and $\Psi_2(\mathbf{x}_n^h, \mathbf{Z}_n^h) = \text{BetP}(m(x_d, \mathbf{Z}_n^h))$.

Using this model on a simulation scenario where the target is equally likely to take the routes IMA and IMJ, renders an estimator with low error probability when the target is in AJ, as is shown in Figure 5.8.

Figure 5.8: Probability of error (MSE) in intention detection when the DS-based model is used (500 trials). The mean error is a function of the uncertainty intervals in Table 5.3.

5.3.2 Expert Knowledge and FOL Input Data in a Tracking and Intent-detection Application

Consider the joint tracking and destination detection scenario introduced in [69], and enhanced into a larger grid, as shown in Figure 5.9. An object moves from an initial state $\mathbf{x}_{t=0}^h = 1$ to a final state $\mathbf{x}_{t=T}^h = \mathbf{x}_T^h = k, k = 1, 2, \ldots, 25$. The value of $\mathbf{x}_t^h, t = 1, 2, \ldots,$ is obtained from a tracking system. The problem is to estimate the intended destination $\mathbf{x}_T^h$. The method introduced in [69] estimates this final state based on Reciprocal Chains (RC) [70]. Assuming that transitions from state $\mathbf{x}_t^h$ to $\mathbf{x}_{t+1}^h$ are Markovian, the intended destination can be found as described in [71]. This is summarized as follows. Let us define:

$$\alpha_i^k(t) = P(\mathbf{x}_t^h = i, \mathbf{Z}_t^h | \mathbf{x}_T^h = k). \tag{5.1}$$

Also, based on (known) stationary Markov transition probabilities $B_{i,j}^k(t) = P(\mathbf{x}_{t+1}^h = j | \mathbf{x}_t^h = i, \mathbf{x}_T^h = k)$, it is possible to define the recursion [69, 71]:

$$\alpha_i^k(t+1) = P(\mathbf{z}_{t+1}^h = i | \mathbf{x}_{t+1}^h) \sum_{j=1}^{N} B_{j,i}^k(t) \alpha_j^k(t). \tag{5.2}$$

The values of $B_{j,i}^k(t)$ can be obtained from A, the transition probability matrix that models the problem, as:

$$B_{j,i}^k(t) = \frac{A_{i,j}(A^{T-t+1})_{j,k}}{(A^{T-t})_{i,k}}. \tag{5.3}$$

Then, the intended destination state can be estimated as:

$$\hat{\mathbf{x}}_T^h = \underset{k}{\operatorname{argmax}} \sum_{i=1}^{N} \alpha_i^k(t) P(\mathbf{x}_T^h = k), \tag{5.4}$$

where N is the number of states.

Now, assume that this process needs to be enhanced by incorporating (uncertain/imprecise) expert advice that indicates the following: (E1) if the object is moving East, then the destination is not $\mathbf{x}_T^h = 21$; (E2) if the object is moving West, then the destination is not $\mathbf{x}_T^h = 5$; (E3) if the object is moving South, then the destination is not $\mathbf{x}_T^h = 5$; (E4) if the object is moving North, then the destination is not $\mathbf{x}_T^h = 21$; (E5) if $(d_x^{(t)} > 2)$, then the destination is $\mathbf{x}_T^h = 5$; (E6) if $(d_y^{(t)} > 2)$, then the destination is $\mathbf{x}_T^h = 21$; (E7) if the object is moving South-East, then the destination is $\mathbf{x}_T^h = 13$; and (E8) if $(d_x^{(t)} > 1)$ and $(d_y^{(t)} > 1)$, then the destination is $\mathbf{x}_T^h = 13$. This information can be converted into FOL sentences, as is shown in Table 5.4, which also contains additional FOL sentences that can be obtained from measurements and knowledge bases. Based on the premises in Table 5.4, it is possible

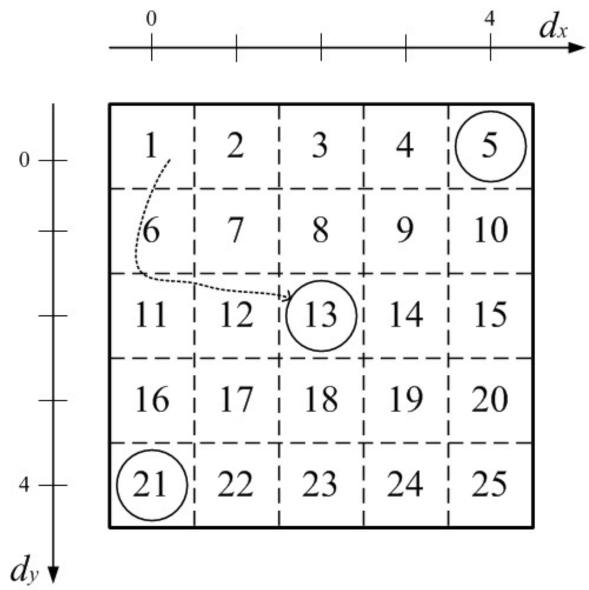


Figure 5.9: Destination detection scenario in 5.3.2. Each block in the grid represents a possible value $\mathbf{x}_t^h = i, i = 1, 2, \ldots, 25$. At time $t = 0$ an object starts moving from $\mathbf{x}_{t=0}^h = 1$ to one of three possible destinations, which are defined by circles. The dotted trajectory shows an example of a possible path followed by an object to arrive at $\mathbf{x}_T^h = 13$. d_x and d_y represent distances along the horizontal and vertical axes, respectively.

to go through an inference process, as shown in Table 5.5. At the end of this inference process, we obtain the following BBAs:

$$
\begin{aligned}
\text{I11}: \; & m_{\text{I11}}(\mathbf{x}_T^h = 5) = \alpha_{\text{I11}}; \\
& m_{\text{I11}}(\mathbf{x}_T^h \neq 5) = 1 - \beta_{\text{I11}}; \\
& m_{\text{I11}}(\{\mathbf{x}_T^h = 5, \mathbf{x}_T^h \neq 5\}) = \beta_{\text{I11}} - \alpha_{\text{I11}}; \\
\text{I12}: \; & m_{\text{I12}}(\mathbf{x}_T^h = 21) = \alpha_{\text{I12}}; \\
& m_{\text{I12}}(\mathbf{x}_T^h \neq 21) = 1 - \beta_{\text{I12}}; \\
& m_{\text{I12}}(\{\mathbf{x}_T^h = 21, \mathbf{x}_T^h \neq 21\}) = \beta_{\text{I12}} - \alpha_{\text{I12}}; \text{ and} \\
\text{I14}: \; & m_{\text{I14}}(\mathbf{x}_T^h = 13) = \alpha_{\text{I13}}; \\
& m_{\text{I14}}(\mathbf{x}_T^h \neq 13) = 1 - \beta_{\text{I14}}; \\
& m_{\text{I14}}(\{\mathbf{x}_T^h = 13, \mathbf{x}_T^h \neq 13\}) = \beta_{\text{I14}} - \alpha_{\text{I14}}. \qquad (5.5)
\end{aligned}
$$

We can then apply CFE fusion (as indicated in Section 4.3). Defining the CFE coefficients as $\gamma_i(A) = m_i(A)$, we obtain a fused BBA $m_{\text{I11,I12,I14}} = m_{\text{I12}}(\cdot) \oplus m_{\text{I12}}(\cdot) \oplus m_{\text{I14}}(\cdot)$. We transform this fused BBA into a combining function as:

$$\Psi_1(\mathbf{x}_T^h, \mathbf{x}_t^h, \mathbf{Z}_t^h) = \text{BetP}\left(m_{\text{I11,I12,I14}}(\mathbf{x}_T^h, \mathbf{x}_t^h, \mathbf{Z}_t^h)\right)/K, \tag{5.6}$$

where K is a normalization constant. Let $\Psi_2(\mathbf{x}_T^h, \mathbf{x}_t^h, \mathbf{Z}_t^h) = \alpha_i^k(t)P(\mathbf{x}_T^h = k)$. Then we obtain the enhanced estimator:

$$\begin{aligned}\hat{\mathbf{x}}_T^{h*} &= \operatorname*{argmax}_k \Psi_1(\mathbf{x}_T^h, \mathbf{x}_t^h, \mathbf{Z}_t^h) \cdot \Psi_2(\mathbf{x}_T^h, \mathbf{x}_t^h, \mathbf{Z}_t^h) \\ &= \operatorname*{argmax}_k \text{BetP}\left(m_{\text{I11,I12,I14}}(\mathbf{x}_T^h, \mathbf{x}_t^h, \mathbf{Z}_t^h)\right)\alpha_i^k(t)P(\mathbf{x}_T^h = k).\end{aligned}$$

Table 5.6 compares the performance of both estimators, $\hat{\mathbf{x}}_T^h$ (Reciprocal Chains, as in [69]) and $\hat{\mathbf{x}}_T^{h*}$ (enhanced by the methods introduced in this manuscript). In particular, this table contains the error obtained on each of the first three iterations for a scenario characterized by a final state $\mathbf{x}_T^h \in \{5, 13, 21\}$ with uniformly distributed prior, transition probability matrices for each state defined based on the parameters in Table 5.7, and zero-mean Gaussian noise with variance 0.5. In this case, the uncertainty in expressions E1–E7 is $[1, 1]$, and the uncertainty in E8 is $[0.7, 1]$. It can be seen that the performance of the original estimator is improved when the soft data is incorporated following the approach described in this manuscript. Indeed, unless the knowledge expert is wrong or deceptive, the estimation of intended destination can achieve faster convergence than the basic RC-based estimator. The convergence rate is a function of the uncertainty intervals in the soft data. Note that, when needed, deception and erroneous soft data can be pre-processed using credibly and/or reliability estimations such as the ones described in [6, 4]. Furthermore, a thorough characterization of the sensitivity of the estimators as a function of the uncertainty intervals would also aid in tuning the performance of the DS-enhanced estimator. These conditions (i.e., deception, erroneous soft data, and sensitivity study) are not considered in this study, and are matter of future research.

Table 5.4: FOL expressions that model the information provided by an expert, as well as information obtained from complementary knowledge base and measurements, for the enhanced destination detection scenario in 5.3.2.

	FOL Clause	**Uncertainty**
	Expert Knowledge	
E1	$E \implies \neg(\mathbf{x}_T^h = 21)$	$[\alpha_{\mathrm{R1}}, \beta_{\mathrm{R1}}]$
E2	$W \implies \neg(\mathbf{x}_T^h = 5)$	$[\alpha_{\mathrm{R2}}, \beta_{\mathrm{R2}}]$
E3	$S \implies \neg(\mathbf{x}_T^h = 5)$	$[\alpha_{\mathrm{R3}}, \beta_{\mathrm{R3}}]$
E4	$N \implies \neg(\mathbf{x}_T^h = 21)$	$[\alpha_{\mathrm{R4}}, \beta_{\mathrm{R4}}]$
E5	$(d_x^{(t)} > 2) \implies (\mathbf{x}_T^h = 5)$	$[\alpha_{\mathrm{R5}}, \beta_{\mathrm{R5}}]$
E6	$(d_y^{(t)} > 2) \implies (\mathbf{x}_T^h = 21)$	$[\alpha_{\mathrm{R6}}, \beta_{\mathrm{R6}}]$
E7	$(S \wedge E) \implies (\mathbf{x}_T^h = 13)$	$[\alpha_{\mathrm{R7}}, \beta_{\mathrm{R7}}]$
E8	$((d_x^{(t)} > 1) \wedge (d_2^{(t)} > 1))$ $\implies (\mathbf{x}_T^h = 13)$	$[\alpha_{\mathrm{R8}}, \beta_{\mathrm{R8}}]$
	Knowledge Base	
K1	$(d_x^{(t)} > d_x^{(t-1)}) \implies E$	$[1, 1]$
K2	$(d_x^{(t)} < d_x^{(t-1)}) \implies W$	$[1, 1]$
K3	$(d_y^{(t)} > d_y^{(t-1)}) \implies S$	$[1, 1]$
K4	$(d_y^{(t)} < d_y^{(t-1)}) \implies N$	$[1, 1]$
	Measurement data	
D1	$(d_x^{(t)} > d_x^{(t-1)})$	$[1, 1]$ if $(d_x^{(t)} > d_x^{(t-1)})$; $[0, 0]$ o/w.
D2	$(d_x^{(t)} < d_x^{(t-1)})$	$[1, 1]$ if $(d_x^{(t)} < d_x^{(t-1)})$; $[0, 0]$ o/w.
D3	$(d_y^{(t)} > d_y^{(t-1)})$	$[1, 1]$ if $(d_y^{(t)} > d_y^{(t-1)})$; $[0, 0]$ o/w.
D4	$(d_y^{(t)} < d_y^{(t-1)})$	$[1, 1]$ if $(d_y^{(t)} < d_y^{(t-1)})$; $[0, 0]$ o/w.
D5	$(d_x^{(t)} > 2)$	$[1, 1]$ if $(d_x^{(t)} > 2)$; $[0, 0]$ o/w.
D6	$(d_y^{(t)} > 2)$	$[1, 1]$ if $(d_y^{(t)} > 2$; $[0, 0]$ o/w.
D7	$(d_x^{(t)} > 1) \wedge (d_y^{(t)} > 1)$	$[1, 1]$ if $(d_x^{(t)}, d_y^{(t)} > 1)$; $[0, 0]$ o/w.

Table 5.5: Uncertain logic inference for processing soft information in 5.3.2. MP: Modus Ponens, as defined in (2.33) and (2.34). OR: Uncertain logic OR, as defined in (2.16).

FOL Clause		Uncertainty	Uncertain Rule Used
I1	E	$[\alpha_{\text{I1}}, \beta_{\text{I1}}]$	MP KB1, D1
I2	W	$[\alpha_{\text{I2}}, \beta_{\text{I2}}]$	MP KB2, D2
I3	S	$[\alpha_{\text{I3}}, \beta_{\text{I3}}]$	MP KB3, D3
I4	N	$[\alpha_{\text{I4}}, \beta_{\text{I4}}]$	MP KB4, D4
I5	$\neg(\mathbf{x}_T^h = 21)$	$[\alpha_{\text{I5}}, \beta_{\text{I5}}]$	MP E1, I1
I6	$\neg(\mathbf{x}_T^h = 5)$	$[\alpha_{\text{I6}}, \beta_{\text{I6}}]$	MP E2, I2
I7	$\neg(\mathbf{x}_T^h = 5)$	$[\alpha_{\text{I7}}, \beta_{\text{I7}}]$	MP E3, I3
I8	$\neg(\mathbf{x}_T^h = 21)$	$[\alpha_{\text{I8}}, \beta_{\text{I8}}]$	MP E4, I4
I9	$(\mathbf{x}_T^h = 5)$	$[\alpha_{\text{I9}}, \beta_{\text{I9}}]$	MP E5, D5
I10	$(\mathbf{x}_T^h = 21)$	$[\alpha_{\text{I10}}, \beta_{\text{I10}}]$	MP KB1, D6
I11	$(\mathbf{x}_T^h = 5)$	$[\alpha_{\text{I11}}, \beta_{\text{I11}}]$	OR ¬I6, ¬I7, I9
I12	$(\mathbf{x}_T^h = 21)$	$[\alpha_{\text{I12}}, \beta_{\text{I12}}]$	OR ¬I5, ¬I8, I10
I13	$(\mathbf{x}_T^h = 13)$	$[\alpha_{\text{I13}}, \beta_{\text{I13}}]$	MP E7, K1, K3
I14	$(\mathbf{x}_T^h = 13)$	$[\alpha_{\text{I14}}, \beta_{\text{I14}}]$	OR I13, D7

Table 5.6: Error on the intended destination estimated by the methods of 5.3.2. By combining an existing Bayesian estimator with an DS-based logic inference method as described in this paper, it is possible to achieve maximum accuracy (minimum error) faster than using the Bayesian estimator alone.

Method	Time step		
	$t = 1$	$t = 2$	$t = 3$
Reciprocal Chains [69]	0.690	0.688	**0.346**
Reciprocal Chains and DS-based combining function	0.690	0.688	**0.010**

Table 5.7: Parameters that define the Markov transition probabilities for the problem in 5.3.2.

From state:	**To state:**	**with probability:**
$\mathbf{x}_T^h = 5$		
1	2, 7	0.7, 0.3
2	3, 8	0.8, 0.2
3	4, 9	0.9, 0.1
4	5	1
7	3, 8	0.8, 0.2
8	4, 9	0.9, 0.1
9	5	1
5	5	1
All other states	1,2,3...25	0.04
$\mathbf{x}_T^h = 13$		
1	2, 6, 7	0.2, 0.1, 0.7
2	7,8	0.8, 0.2
6	7, 12	0.7, 0.3
7	8, 12, 13	0.1, 0.2, 0.7
8	12, 13	0.2, 0.8
12	8, 13	0.1, 0.9
13	13	1
All other states	1,2,3...25	0.04
$\mathbf{x}_T^h = 21$		
1	6 ,7	0.7, 0.3
6	11, 12	0.8, 0.2
7	11, 12	0.8, 0.2
11	16, 17	0.9, 0.1
12	16, 17	0.9, 0.1
16	21, 22	0.9, 0.1
17	21, 22	0.9, 0.1
21	21	1
22	21	1

CHAPTER 6

Conclusions

6.1 Summary

There is a growing necessity for the development of new methods in fusion, estimation, and tracking that can incorporate and suitably combine heterogeneous forms of both hard (e.g., physics-based sensor data) and soft (e.g., text from witness statements, blogs, newspapers) data. Although hard data fusion is a well understood problem, reasoning with soft data imposes new challenges to the data fusion problem. By its nature, soft data in the form of text is more qualitative in nature, inherently possessing uncertainty and imperfections. The work in this dissertation addresses hard and soft data fusion needs as follows:

1. We have introduced Uncertain Logic Processing (ULP), a DS theoretic approach for first order logic operations. ULP provides support for handling variables and quantifiers, in addition to fundamental logic operations (i.e., $\neg, \wedge, \vee, \implies$). ULP enables systematic generation of mass assignments for data fusion applications. Furthermore, by using appropriate fusion operators, higher-level applications are possible within this framework, such as inference and resolution based on uncertain data models. ULP is consistent with classical logic, rendering the classical logic results when the scenario represents "perfect" (i.e., without uncertainty) data/models. The consistency with classical logic gives

the confidence to apply our proposed models as an extension of classical logic in reasoning.

2. We have developed a filtering and tracking framework for incorporating both hard and soft data, demonstrating how the probability posterior can be expressed as a product of combining functions over subsets of the state and measurement variables. This flexible approach engenders the possibility of employing different approaches to model the interaction of soft and hard states and variables. Our results for the DS models illustrate the potential for incorporating natural language in the form of ULP within tracking systems (e.g., particle filters). This flexible approach has the potential to allow for the incorporation of more sophisticated forms of uncertainty modeling (e.g., more powerful DS fusion models, random sets).

3. We have introduced a new method for reducing computational complexity and increasing robustness against conflicting evidence in ULP reasoning systems. The new reasoning method is based on a convex optimization formulation of the satisfiability problem associated with the ULP model.

6.2 Future Work

As mentioned above, this dissertation describes new methods for dealing with hard and soft data fusion in scenarios characterized by incomplete, inconsistent, and/or imperfect data. These new methods provide the foundation of a robust data fusion theory that can be further developed and integrated into high-level applications. Potential avenues for future research include, among others:

- **Exploring adaptability of ULP to changing data and applications.** The applications of ULP illustrated in this dissertation are focused on a a set of particular applications, namely, human-robot interaction, detection, and tracking. Translating the results that we have obtained in these problems to other types of applications may prove challenging. For example, the adaptation of reasoning models have proven challenging in machine learning, where researchers and engineers typically rely on alternatives such as transfer learning or domain adaptation to facilitate the adaptation of deep neural networks to other domains.

A future research area then deals with answering the question: How can we properly and efficiently configure our ULP framework for adapting to other applications and data? A first alternative to explore in this area is the relaxation of classical logic requirements in ULP (recall that the first implementation of ULP focuses on ensuring consistency with classical logic). In particular, we could explore the configuration of ULP with paraconsistent logic properties. By allowing paraconsistent logic formulations the ULP framework will be better prepared for handling data inconsistencies and changes. Once the use of paraconsistent logic configurations is analyzed and thoroughly understood, we could move to the application of other techniques, which could borrow concepts from the traditional transfer learning theory.

- **Approaching causal inference through ULP models.** One of the main obstacles that researchers have met when trying to implement systems that exhibit human-level intelligence is the understanding of cause-effect connections [72]. Surpassing this obstacle means that machines would be able to purposely distort the data with "acts of imagination" for answering "What if?" questions.

  The goal would be then to explore opportunities for addressing causal inference. To this end, we could exploit the formulation of ULP models as probabilistic graphical models. For causal inference, the use of graphical models could be very powerful as they allow encoding of data relations and assumptions as graphs, which is one of the tools used by researchers for analyzing cause-effect relations. Once we validate the use of graphical models as proper tools for causal inference, we could move to using it for counterfactual analysis, as well as for enhancing adaptability and reducing model biases.

- **Incorporating self-diagnosing capabilities.** Dealing with large amounts of data brings higher chances of being faced with inconsistent data samples. These samples could occur due to data sampling limitations, incomplete or imperfect data/reasoning models, or erroneous data fed to the reasoning system by adversaries. Humans abilities for introspection allow them to pinpoint data inconsistencies and to question the quality of both data and reasoning model. To the best of our knowledge, the incorporation of self-diagnosing capabilities in modern automated reasoning systems have not been explored yet. We could explore opportunities in this area by analyzing ways in which philosophic approaches (represented in a vast literature on the topic) can be translated for integration of self-diagnosing capabilities into automated reasoning systems.

The work on these areas could provide more robust automated reasoning systems for multiple applications, such as detection and tracking, situational awareness, crowdsourcing, human-robot interaction, medical diagnosis, among others (see Section 1.2 for a description of the specific needs in these applications).

Bibliography

[1] B. Ristic, B. R. S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Radar Library. Artech House, 2004.

[2] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[3] D. L. Hall and J. M. Jordan. *Human-centered Information Fusion*. Artech House, 2010.

[4] Rafael C. Núñez, Thanuka Wickramarathne, Kamal Premaratne, Manohar N. Murthi, Sandra Kübler, Matthias Scheutz, and Marco Pravia. Credibility assessment and inference for fusion of hard and soft information. In *Proc. 2nd International Conference on Cross-Cultural Decision Making: Focus 2012 (also in Advances in Design for Cross-Cultural Activities, 1, pp. 96-105, Eds: D. D. Schmorrow, D. M. Nicholson, CRC Press, 2013)*, July 2012.

[5] Rafael C. Núñez, Buddhika Samarakoon, Kamal Premaratne, and Manohar N. Murthi. Hard and soft data fusion for joint tracking and classification/intent-detection. In *16th International Conference on Information Fusion*, Istanbul, Turkey, July 2013.

[6] Thanuka L. Wickramarathne, Kamal Premaratne, and Manohar N. Murthi. Consensus-based credibility estimation of soft evidence for robust data fusion. In *Proc. 2nd International Conference on Belief Functions*, May 2012.

[7] B. Khaleghi and F. Karray. Human-centered multi-target tracking: A random set theoretic approach. In *IEEE 13th International Conference on Information Reuse and Integration (IRI)*, pages 193 –200, August 2012.

[8] Mica R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37:32–64(33), March 1995.

[9] Jie Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power. Using Social Media to Enhance Emergency Situation Awareness. *IEEE Intelligent Systems*, 27(6):52–59, November 2012.

[10] Jeff Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, New York, NY, USA, 1 edition, 2008.

[11] Poonam Gupta and Vishal Gupta. Article: A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4):1–8, September 2012.

[12] Matthias Scheutz, Gordon Briggs, Rehj Cantrell, Evan Krause, Thomas Williams, and Richard Veale. Novel Mechanisms for Natural Human-Robot Interactions in the DIARC Architecture. In *Proceedings of AAAI Workshop on Intelligent Robotic Systems*, 2013.

[13] Darin Archer. GraceMed – An Automated Medical Diagnosis System, May 2009.

[14] Joseph Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46(3):311–350, December 1990.

[15] Joseph Y. Halpern. A logical approach to reasoning about uncertainty: a tutorial. In X. Arrazola, K. Korta, and F.J. Pelletier, editors, *Discourse, Interaction and Communication*, Philosophical Studies Series. Springer, 1998.

[16] Didier Dubois, Petr Hájek, and Henri Prade. Knowledge-driven versus data-driven logics. *Journal of Logic, Language and Information*, 9(1):65–89, January 2000.

[17] Luca Boldrin and Alessandro Saffiotti. A modal logic for fusing partial belief of multiple reasoners. *Journal of Logic and Computation*, 9(1):81–103, 1999.

[18] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[19] Lluis Godo, Petr Hájek, and Francesc Esteva. A fuzzy modal logic for belief functions. *Fundam. Inform.*, 57(2-4):127–146, 2003.

[20] Nils J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–88, February 1986.

[21] Didier Dubois and Henri Prade. Possibilistic logic: a retrospective and prospective view. *Fuzzy Sets and Systems*, 144(1):3–23, 2004.

[22] Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, February 2006.

[23] Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A Short Introduction to Probabilistic Soft Logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*, December 2012.

[24] Grzegorz Malinowski. *Many-Valued Logics*. Oxford Logic Guides. Clarendon Press, 1993.

[25] Lotfi Asker Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[26] Didier Dubois and Henri Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty (traduction revue et augmente de "Thorie des Possibilits")*. Plenum Press, New York, 1988.

[27] Ronald Fagin and Joseph Y. Halpern. Uncertainty, belief, and probability. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'89, pages 1161–1167, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[28] Gregory M. Provan. A logic-based analysis of Dempster-Shafer theory. *Int. J. Approx. Reasoning*, 4(5-6):451–495, 1990.

[29] Alessandro Saffiotti. A belief-function logic. In *Proceedings of the tenth national conference on Artificial intelligence*, AAAI'92, pages 642–647. AAAI Press, 1992.

[30] A. Saffiotti and E. Umkehrer. Inference-driven construction of valuation systems from first-order clauses. *IEEE Transactions on Systems, Man and Cybernetics*, 24(11):1611–1624, 1994.

[31] B. Ristic and P. Smets. Target identification using belief functions and implication rules. *IEEE transactions on Aerospace and Electronic Systems*, 41, July 2005.

[32] A. Benavoli, L. Chisci, A. Farina, and B. Ristic. Modelling uncertain implication rules in evidence theory. In *11th International Conference on Information Fusion*, 2008.

[33] A. N. Bishop and B. Ristic. Fusion of natural language propositions: Bayesian random set framework. In *Proc. 14th International Conference on Information Fusion (FUSION)*, July 2011.

[34] J. Wang and P. Domingos. Hybrid Markov Logic Networks. In *Proc. 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, pages 1106–1111. AAAI Press, 2008.

[35] E. Doucette. *The Application of a Particle Filter to Urban Ground Target Localization, Tracking, and Intercept*. Ph.D. thesis, Auburn University, 2012.

[36] D. L. Hall, M. McNeese, J. Llinas, and T. Mullen. A framework for dynamic hard/soft fusion. In *11th International Conference on Information Fusion*, July 2008.

[37] K. Premaratne, M. N. Murthi, J. Zhang, M. Scheutz, and P. H. Bauer. A Dempster-Shafer theoretic conditional approach to evidence updating for fusion of hard and soft data. In *12th International Conference on Information Fusion, FUSION '09*, pages 2122 –2129, July 2009.

[38] Thanuka L. Wickramarathne, Kamal Premaratne, Manohar N. Murthi, and Matthias Scheutz. A dempster-shafer theoretic evidence updating strategy for non-identical frames of discernment. In *Proc. Workshop on the Theory of Belief Functions (WTBF10)*, 2010.

[39] Thanuka L. Wickramarathne, Kamal Premaratne, Manohar N. Murthi, Matthias Scheutz, Sandra Kubler, and Marco Pravia. Belief theoretic methods for soft and hard data fusion. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2388–2391, May 2011.

[40] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[41] Rafael C. Núñez, Matthias Scheutz, Kamal Premaratne, and Manohar N. Murthi. Modeling Uncertainty in First-Order Logic: A Dempster-Shafer Theoretic Approach. In *8th International Symposium on Imprecise Probability: Theories and Applications*, Compiègne, France, July 2013.

[42] Rafael C. Núñez, Manohar N. Murthi, Kamal Premaratne, Matthias Scheutz, and Otávio Bueno. Uncertain Logic Processing: logic-based inference and reasoning using Dempster–Shafer models. *International Journal of Approximate Reasoning*, 95:1 – 21, 2018.

[43] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[44] Ronald R. Yager. On the Dempster-Shafer Framework and New Combination Rules. *Information Sciences*, 41(2):93–137, March 1987.

[45] T. Inagaki. Interdependence between safety-control policy and multiple-sensor schemes via Dempster-Shafer theory. *IEEE Transactions on Reliability*, 40(2):182–188, June 1991.

[46] Ronald Fagin and Joseph Y. Halpern. A new approach to updating beliefs. In *Uncertainty in Artificial Intelligence*, pages 347–374. Elsevier Science Publishers, 1991.

[47] Anne-Laure Jousselme, Liu Chunsheng, D. Grenier, and E. Boss. Measuring ambiguity in the evidence theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 36(5):890–903, September 2006.

[48] Philippe Smets. Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38(2):133 – 147, 2005.

[49] P. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, 66(2):191 – 234, 1994.

[50] Milan Daniel. Probabilistic Transformations of Belief Functions. In Lluís Godo, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 3571 of *Lecture Notes in Computer Science*, pages 539–551. Springer Berlin Heidelberg, 2005.

[51] Pietro Baroni and Paolo Vicig. Transformations from Imprecise to Precise Probabilities. In Thomas Dyhre Nielsen and Nevin Lianwen Zhang, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 2711 of *Lecture Notes in Computer Science*, pages 37–49. Springer Berlin Heidelberg, 2003.

[52] Barry R. Cobb and Prakash P. Shenoy. A Comparison of Methods for Transforming Belief Function Models to Probability Models. In ThomasDyhre Nielsen and NevinLianwen Zhang, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 2711 of *Lecture Notes in Computer Science*, pages 255–266. Springer Berlin Heidelberg, 2003.

[53] Edward A. Bender. *Mathematical Methods in Artificial Intelligence*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1996.

[54] Didier Dubois and Henri Prade. A set-theoretic view of belief functions. In Roland Yager and Liping Liu, editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 375–410. Springer Berlin / Heidelberg, 2008.

[55] Guo-Jun Wang and Yee Leung. Integrated semantics and logic metric spaces. *Fuzzy Sets Systems*, 136(1):71–91, May 2003.

[56] Rafael C. Núñez, Ranga Dabarera, Matthias Scheutz, Otavio Bueno, Kamal Premaratne, and Manohar N. Murthi. DS-based uncertain implication rules for inference and fusion applications. In *16th International Conference on Information Fusion*, Istanbul, Turkey, July 2013.

[57] Lintao Zhang and Sharad Malik. The quest for efficient boolean satisfiability solvers. In Ed Brinksma and KimGuldstrand Larsen, editors, *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pages 17–36. Springer Berlin Heidelberg, 2002.

[58] Pedro Domingos and Daniel Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan and Claypool Publishers, 1st edition, 2009.

[59] S. Kok, P. Singla, M. Richardson, P. Domingos, M. Sumner, H. Poon, D. Lowd, J. Wang, and A. Nath. *The Alchemy System for Statistical Relational AI: User Manual*. Department of Computer Science and Engineering, University of Washington, January 2010.

[60] Charu C. Aggarwal. *Managing and Mining Uncertain Data*. Springer Publishing Company, Incorporated, 2009.

[61] Michael I. Jordan. Graphical models. *Statistical Science, (Special Issue on Bayesian Statistics)*, 19:140–155, 2004.

[62] Daniel Lowd and Pedro Domingos. Efficient weight learning for Markov logic networks. *Knowledge Discovery in Databases: PKDD 2007*, pages 200–211, 2007.

[63] Stanley Kok and Pedro Domingos. Learning the Structure of Markov Logic Networks. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 441–448, New York, NY, USA, 2005. ACM.

[64] Pedro Domingos and William Austin Webb. A Tractable First-Order Probabilistic Logic. In Jörg Hoffmann and Bart Selman, editors, *AAAI*. AAAI Press, 2012.

[65] H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690, November 2011.

[66] Pedro Domingos, Daniel Lowd, Stanley Kok, Aniruddh Nath, Hoifung Poon, Matthew Richardson, and Parag Singla. Unifying Logical and Statistical AI. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '16, pages 1–11, New York, NY, USA, 2016. ACM.

[67] Kenneth Lange. *MM Optimization Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[68] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[69] Mustafa Fanaswala, Vikram Krishnamurthy, and Langford White. Destination-aware target tracking via syntactic signal processing. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3692–3695. IEEE, May 2011.

[70] Benton Jamison. Reciprocal processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 30:65–86, 1974.

[71] Langford B White and Francesco Carravetta. Optimal Smoothing for Finite State Hidden Reciprocal Processes. *IEEE Transactions on Automatic Control*, 56(9):2156–2161, September 2011.

[72] Judea Pearl. The seven pillars of causal reasoning with reflections on machine learning. Technical Report R-481, Department of Computer Science, University of California, 2018.

APPENDIX A

DS model for CFE-based AND operations using the Logic-Consistent Parameter Selection Strategy

Based on the definition of the CFE fusion operator:

$$m(x) = \gamma_1(x) + \gamma_1(\Theta)m_1(x) + \gamma_2(x) + \gamma_2(\Theta)m_2(x). \tag{A.1}$$

Substituting the CFE coefficients for the AND operation, as indicated by Definition 12, in (A.1):

$$m(x) = 2\gamma(x) + 2\gamma(\Theta)(\alpha_1 + \alpha_2).$$

- When $\delta_1 + \delta_2 \neq 0$:

$$\begin{aligned} m(x) &= \frac{\underline{\alpha}(\beta_1 + \beta_2) - \underline{\beta}(\alpha_1 + \alpha_2)}{\delta_1 + \delta_2} + \frac{\underline{\delta}(\alpha_1 + \alpha_2)}{\delta_1 + \delta_2} \\ &= \tfrac{1}{\delta_1+\delta_2}(\underline{\alpha}\beta_1 + \underline{\alpha}\beta_2 - \alpha_1\underline{\beta} - \alpha_2\underline{\beta} + \underline{\delta}(\alpha_1 + \alpha_2)). \end{aligned}$$

Since $\underline{\delta} = \underline{\beta} - \underline{\alpha}$:

$$\begin{aligned} m(x) &= \tfrac{1}{\delta_1+\delta_2}(\underline{\alpha}\beta_1 + \underline{\alpha}\beta_2 - \alpha_1\underline{\beta} - \alpha_2\underline{\beta} \\ &\qquad + \alpha_1\underline{\beta} + \alpha_2\underline{\beta} - \alpha_1\underline{\alpha} - \alpha_2\underline{\alpha}) \\ &= \tfrac{1}{\delta_1+\delta_2}(\underline{\alpha}\beta_1 + \underline{\alpha}\beta_2 - \alpha_1\underline{\alpha} - \alpha_2\underline{\alpha}) \\ &= \tfrac{1}{\delta_1+\delta_2}(\underline{\alpha}(\beta_2 - \alpha_2 + \beta_1 - \alpha_1)). \end{aligned} \tag{A.2}$$

Substituting $\delta_1 = \beta_1 - \alpha_1$ and $\delta_2 = \beta_2 - \alpha_2$ in (A.2): $m(x) = \underline{\alpha}$.

- When $\delta_1 + \delta_2 = 0$, and making $\gamma(\Theta) = 0$:

$$m(x) = 2\gamma(x) = \underline{\alpha}.$$

The mass $m(\overline{x})$ is given by:

$$m(\overline{x}) = \gamma_1(\overline{x}) + \gamma_1(\Theta)m_1(\overline{x}) + \gamma_2(\overline{x}) + \gamma_2(\Theta)m_2(\overline{x}). \tag{A.3}$$

Substituting the CFE coefficients as indicated by Definition 12 in (C.1):

$$m(\overline{x}) = 2\gamma(\overline{x}) + 2\gamma(\Theta)(2 - \beta_1 - \beta_2).$$

- When $\delta_1 + \delta_2 \neq 0$:

$$\begin{aligned} m(\overline{x}) = & \frac{\delta_1 + \delta_2 - \underline{\beta}(2 - \alpha_1 - \alpha_2) + \underline{\alpha}(2 - \beta_1 - \beta_2)}{\delta_1 + \delta_2} \\ & + \frac{\underline{\delta}(2 - \beta_1 - \beta_2)}{\delta_1 + \delta_2} \\ = & \tfrac{1}{\delta_1+\delta_2}(\delta_1 + \delta_2 - \underline{\beta}(2 - \alpha_1 - \alpha_2) \\ & + \underline{\alpha}(2 - \beta_1 - \beta_2) + \underline{\delta}(2 - \beta_1 - \beta_2)). \end{aligned}$$

  Since $\underline{\delta} = \underline{\beta} - \underline{\alpha}$:

$$\begin{aligned} m(\overline{x}) &= \tfrac{1}{\delta_1+\delta_2}(\delta_1 + \delta_2 - \underline{\beta}(2 - \alpha_1 - \alpha_2) \\ &\quad + \underline{\alpha}(2 - \beta_1 - \beta_2) + (\underline{\beta} - \underline{\alpha})(2 - \beta_1 - \beta_2)) \\ &= \tfrac{1}{\delta_1+\delta_2}(\delta_1 + \delta_2 - \underline{\beta}(2 - \alpha_1 - \alpha_2 - 2 + \beta_1 + \beta_2)) \\ &= \tfrac{1}{\delta_1+\delta_2}(\delta_1 + \delta_2 - \underline{\beta}(\beta_1 - \alpha_1 + \beta_2 - \alpha_2)). \end{aligned}$$

  Substituting $\delta_1 = \beta_1 - \alpha_1$ and $\delta_2 = \beta_2 - \alpha_2$ in (C.1):

$$m(x) = 1 - \underline{\beta}.$$

- When $\delta_1 + \delta_2 = 0$, and making $\gamma(\Theta) = 0$:

$$m(\overline{x}) = 2\gamma(\overline{x}) = 1 - \underline{\alpha} = 1 - \underline{\beta}.$$

Finally, $m(\Theta) = 1 - m(x) - m(\overline{x}) = \underline{\beta} - \underline{\alpha}$.

APPENDIX B

Properties of Classically Consistent CFE-based Uncertain Logic Operations

Consider logic expressions of the form $\varphi(x_i)$, with $1 \leq i \leq N$. Then, the following properties are satisfied:

1. *Idempotency*: This property is defined by: $\varphi_i(x) \wedge \varphi_i(x) = \varphi_i(x) \vee \varphi_i(x) = \varphi_i(x)$. In this case:

$$\begin{aligned}
m_\wedge(x) &= \underline{\alpha} = \min(\alpha_i, \alpha_i) = \alpha_i \\
&= \max(\alpha_i, \alpha_i) = \overline{\alpha} = m_\vee(x); \\
m_\wedge(\overline{x}) &= 1 - \underline{\beta} = 1 - \min(\beta_i, \beta_i) = 1 - \beta_i \\
&= 1 - \max(\beta_i, \beta_i) = 1 - \overline{\beta} = m_\vee(\overline{x}); \\
m_\wedge(\Theta) &= \underline{\beta} - \underline{\alpha} = \beta_i - \alpha_i \\
&= \overline{\beta} - \overline{\alpha} = m_\vee(\Theta).
\end{aligned}$$

2. *Commutativity*: This property refers to satisfying: $\varphi_1(x) \wedge \varphi_2(x) = \varphi_2(x) \wedge \varphi_1(x)$,
and $\varphi_1(x) \vee \varphi_2(x) = \varphi_2(x) \vee \varphi_1(x)$. Let us call $m_{\varphi_i \wedge \varphi_j}(\cdot)$ the BBA resulting

from $\varphi_i(x) \wedge \varphi_j(x)$, $i = \{1, 2\}$. Then, for the AND operation:

$$\begin{aligned}
m_{\varphi_1 \wedge \varphi_2}(x) &= \min(\alpha_1, \alpha_2) \\
&= \min(\alpha_2, \alpha_1) = m_{\varphi_2 \wedge \varphi_1}(x) \\
m_{\varphi_1 \wedge \varphi_2}(\overline{x}) &= 1 - \min(\beta_1, \beta_2) \\
&= 1 - \min(\beta_2, \beta_1) = m_{\varphi_2 \wedge \varphi_1}(\overline{x}) \\
m_{\varphi_1 \wedge \varphi_2}(\Theta) &= \min(\beta_1, \beta_2) - \min(\alpha_1, \alpha_2) \\
&= \min(\beta_2, \beta_1) - \min(\alpha_2, \alpha_1) \\
&= m_{\varphi_2 \wedge \varphi_1}(\Theta).
\end{aligned}$$

A proof for commutativity for the logical OR operation is obtained by following a similar procedure.

3. *Associativity*: The associative property is defined by: $\varphi_1(x) \wedge [\, \varphi_2(x) \wedge \varphi_3(x)\,] = [\varphi_1(x) \wedge \varphi_2(x)\,] \wedge \varphi_3(x)$, and $\varphi_1(x) \vee [\, \varphi_2(x) \vee \varphi_3(x)\,] = [\, \varphi_1(x) \vee \varphi_2(x)\,] \vee \varphi_3(x)$. Let us call $\varphi_4(\cdot)$ the model generated by $\varphi_2(x) \wedge \varphi_3(x)$, and $\varphi_5(\cdot)$ the model generated by $\varphi_1(x) \wedge \varphi_2(x)$. Also, let us call $m_{\varphi_i \wedge \varphi_j}(\cdot)$ the BBA resulting from $\varphi_i(x) \wedge \varphi_j(x)$, $i = \{1, \ldots, 5\}$. Our goal (for the AND operation) is to show that the model for $\varphi_1(\cdot) \wedge \varphi_4(\cdot)$ is equivalent to the model for $\varphi_5(\cdot) \wedge \varphi_3(\cdot)$:

$$\begin{aligned}
m_{\varphi_1 \wedge \varphi_4}(x) &= \min(\alpha_1, \min(\alpha_2, \alpha_3)) \\
&= \min(\min(\alpha_1, \alpha_2), \alpha_3) = m_{\varphi_5 \wedge \varphi_3}(x) \\
m_{\varphi_1 \wedge \varphi_4}(\overline{x}) &= 1 - \min(\beta_1, \min(\beta_2, \beta_3)) \\
&= 1 - \min(\min(\beta_1, \beta_2), \beta_3) \\
&= m_{\varphi_5 \wedge \varphi_2}(\overline{x}) \\
m_{\varphi_1 \wedge \varphi_4}(\Theta) &= \min(\beta_1, \min(\beta_2, \beta_3)) \\
&\quad - \min(\alpha_1, \min(\alpha_2, \alpha_3)) \\
&= \min(\min(\beta_1, \beta_2), \beta_3) \\
&\quad - \min(\min(\alpha_1, \alpha_2), \alpha_3) = m_{\varphi_5 \wedge \varphi_3}(\Theta).
\end{aligned}$$

A proof for associativity for the logical OR operation is obtained by following a similar procedure.

4. *Distributivity*: Distributive operations satisfy: $\varphi_1(x_i) \wedge [\, \varphi_2(x_j) \vee \varphi_3(x_k)\,] = [\, \varphi_1(x_i) \wedge \varphi_2(x_j)\,] \vee [\, \varphi_1(x_i) \wedge \varphi_3(x_j)\,]$, and $\varphi_1(x_i) \vee [\, \varphi_2(x_j) \wedge \varphi_3(x_k)\,] = [\, \varphi_1(x_i) \vee \varphi_2(x_j)\,] \wedge [\, \varphi_1(x_i) \vee \varphi_3(x_j)\,]$. Let us call $\varphi_4(\cdot)$ the model generated by $\varphi_1(x) \wedge [\varphi_2(x) \vee \varphi_3(x)]$, and $\varphi_5(\cdot)$ the model generated by $[\varphi_1(x) \wedge \varphi_2(x)] \vee [\varphi_1(x) \wedge \varphi_3(x)]$. Our goal is to show that the model for $\varphi_4(\cdot)$ is equivalent to the model for $\varphi_5(\cdot)$. In general, these two models are:

$$\begin{aligned} m_{\varphi_4}(x) &= \min(\alpha_1, \max(\alpha_2, \alpha_3)); \\ m_{\varphi_4}(\overline{x}) &= 1 - \min(\beta_1, \max(\beta_2, \beta_3)); \\ m_{\varphi_4}(\Theta) &= \min(\beta_1, \max(\beta_2, \beta_3)); \\ &\quad - \min(\alpha_1, \max(\alpha_2, \alpha_3)); \text{ and} \end{aligned}$$

$$\begin{aligned} m_{\varphi_5}(x) &= \max(\min(\alpha_1, \alpha_2), \min(\alpha_1, \alpha_3)); \\ m_{\varphi_5}(\overline{x}) &= 1 - \max(\min(\beta_1, \beta_2), \min(\beta_1, \beta_3)); \\ m_{\varphi_5}(\Theta) &= \max(\min(\beta_1, \beta_2), \min(\beta_1, \beta_3)) \\ &\quad - \max(\min(\alpha_1, \alpha_2), \min(\alpha_1, \alpha_3)). \end{aligned}$$

Now, consider the focal set x. We have three cases (other possible cases are equivalent to these three after applying the commutativity rule): (a) $\alpha_1 \leq \alpha_2 \leq \alpha_3$; (b) $\alpha_2 \leq \alpha_1 \leq \alpha_3$; and (c) $\alpha_2 \leq \alpha_3 \leq \alpha_1$. The mass associated to the focal set x is:

(a) $m_{\varphi_4}(x) = \alpha_1 = m_{\varphi_5}(x)$;

(b) $m_{\varphi_4}(x) = \alpha_1 = m_{\varphi_5}(x)$; and

(c) $m_{\varphi_4}(x) = \alpha_3 = m_{\varphi_5}(x)$;

i.e., $m_{\varphi_4}(x) = m_{\varphi_5}(x)$ in all the cases. For the focal set $\overline{x}$ we also have three basic cases: (a) $\beta_1 \leq \beta_2 \leq \beta_3$; (b) $\beta_2 \leq \beta_1 \leq \beta_3$; and (c) $\beta_2 \leq \beta_3 \leq \beta_1$; which render:

(a) $m_{\varphi_4}(\overline{x}) = 1 - \beta_1 = m_{\varphi_5}(\overline{x})$;

(b) $m_{\varphi_4}(\overline{x}) = 1 - \beta_1 = m_{\varphi_5}(\overline{x})$; and

(c) $m_{\varphi_4}(\overline{x}) = 1 - \beta_3 = m_{\varphi_5}(\overline{x})$;

Based on the cases above, it can be shown that also $m_{\varphi_4}(\Theta) = m_{\varphi_5}(\Theta)$, proving distributivity for the logical AND operation. A proof for distributivity for the logical OR operation is obtained by following a similar procedure.

APPENDIX C

Probabilistic Models in Uncertain Logic

Table C.1 illustrates the probabilistic case of CFE-based uncertainty logic. In this case, uncertainty parameters are defined as $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$. Let us denote $\varphi_1(x)$ and $\varphi_2(x)$ as $\varphi(x)[\alpha_1]$ and $\varphi(x)[\alpha_2]$, respectively. For the probabilistic case, we then get

$$\begin{aligned} &\varphi(x)[\alpha_1] \wedge \varphi(x)[\alpha_2] = \varphi(x)[\underline{\alpha}]; \\ &\varphi(x)[\alpha_1] \vee \varphi(x)[\alpha_2] = \varphi(x)[\overline{\alpha}]. \end{aligned} \tag{C.1}$$

Table C.1: CFE-Based Logical AND/OR Operations: Probabilistic Scenario

Parameters	**Logical AND**	**Logical OR**
$[\alpha_1, \beta_1] = [\alpha_1, \alpha_1]$ $[\alpha_2, \beta_2] = [\alpha_2, \alpha_2]$	$m(x) = \gamma_1(x) + \gamma_2(x) + \gamma_1(\Theta)\alpha_1 + \gamma_2(\Theta)\alpha_2$ $m(\overline{x}) = \gamma_1(\overline{x}) + \gamma_2(\overline{x})$ $+\gamma_1(\Theta)(1-\alpha_1) + \gamma_2(\Theta)(1-\alpha_2)$ $m(\Theta) = 0$	$m(x) = \gamma_1(\overline{x}) + \gamma_2(\overline{x}) + \gamma_1(\Theta)\alpha_1 + \gamma_2(\Theta)\alpha_2$ $m(\overline{x}) = \gamma_1(x) + \gamma_2(x)$ $+\gamma_1(\Theta)(1-\alpha_1) + \gamma_2(\Theta)(1-\alpha_2)$ $m(\Theta) = 0$
LCR Strategy	$m(x) = \underline{\alpha}$ $m(\overline{x}) = 1 - \underline{\alpha}$ $m(\Theta) = 0$	$m(x) = \overline{\alpha}$ $m(\overline{x}) = 1 - \overline{\alpha}$ $m(\Theta) = 0$