Introduction to uncertainty formalisms

Anthony Hunter[1] and Simon Parsons[2]

[1] Department of Computer Science, University College London,
Gower Street, London, WC1E 6BT
[2] Department of Electronic Engineering, Queen Mary and Westfield College,
University of London, London, E1 4NS, United Kingdom.

Abstract. The heterogeneity of uncertainty in the real-world has driven the development of a wide variety of formal approaches to representing and reasoning with uncertainty in knowledge. There is now a shift to analysing the application of many of these formalisms. Here, we briefly consider some of the issues in the application of uncertainty formalisms.

1 Introduction

Most of the information we have about the real world is uncertain. We are often uncertain about the information we need to run our lives; what the weather will be like tomorrow, what time the train we need to catch home will actually depart from the station, and how many students will make it to the next tutorial we organise. The ubiquitous nature of uncertainty coupled with the increasing computerisation of all areas of life means that uncertainty in information is a significant problem in computing. Increasing numbers of applications—such as diagnostic systems, data mining systems, scientific databases, information retrieval systems, knowledge-based systems, and natural language interfaces—require an ability to represent, manage, and reason with uncertain information. In order to handle uncertain information, we first need sufficiently expressive means to represent it. Then we need to incorporate a reasoning component into our computing systems in order to derive answers from the uncertain information. In addition, we assume that given the diverse kinds of uncertainty in information, we need to develop a range of techniques.

Adopting *ad hoc* approaches to computer-based handling of uncertain information can easily become counter-productive. Using computers in an organization is in some respects like delegating. Tasks are delegated when the delegator has confidence in what the delegatee will do. So for example, there is confidence in delegating payroll activities to computers because of the well-understood principles of arithmetic and accountancy. When handling uncertain information, the behaviour of even small datasets can be difficult to predict. Therefore, if we are to delegate handling of uncertain information, we should only do so within the context of well-understood principles.

Numerous formalisms for dealing with uncertainty have been studied over the years. These include those essentially numerical methods based on probability theory, fuzzy set theory, and possibility theory through to those largely symbolic methods such as default logics, paraconsistent logics and argumentation. We provide a review of key formalisms in Chapter 2, and other reviews include [3, 4, 7]. Whilst many questions remain in developing these kinds of uncertainty formalism, there is now a significant shift to developing applications using these formalisms. This shift to applications is raising many new questions of viability. In the past, interest often used to be focussed on questions such as:

- Does this formalism have a clear semantics?
- What are the properties of this formalism?
- How does this formalism relate to others?

Nowadays questions which relate more to the use rather than the development of the various formalisms are becoming important such as:

- Are we now ready to use these uncertainty formalisms in "real-world" problems?
- What kinds of applications have been successful?
- What are the shortcomings in the current formalisms?
- What developments will be required for uncertainty management techniques to be more widely used?
- In what ways can uncertainty formalisms be applied?
- What are the issues that need to be addressed for wider uptake?
- What limitations can we identify that currently prevent their application?

In this chapter, we discuss some of the background to these questions.

2 Application areas

The nature of uncertainty in information is complex. Many factors affect the types of uncertainty [2, 8], sources of uncertainty [1, 5], and the degrees of uncertainty. Nevertheless, there are many strategies that users adopt for aggregating such factors in order to minimize the negative ramifications of operating under uncertainty. Indeed, via learning, humans, and similarly organizations, can be become highly adept at using uncertain information.

The ubiquitous usage of uncertain information by people contrasts sharply with the low level of computer-based handling of uncertain information. Whilst many theoretical models have been proposed for the management of uncertainty, present generation information systems have very limited capabilities in this respect.

The situation seems set to change as the expanding role of computing means that handling uncertain information will become increasingly significant. Indeed as uncertainty pervades any real-world scenario, uncertainty handling must be incorporated into any computing system that attempts to provide a substantive model of the real-world. Even now, diverse types of information system such as database systems, information retrieval systems, expert systems, and groupware systems are currently being developed to incorporate uncertainty formalisms.

Application feature	Chapter											
	3	4	5	6	7	8	9	10	11	12	13	14
Embedded knowledgebase			x				x	x	x	x		
Intelligent interface		x				x						
Decision-support	x			x	x						x	x
Information retrieval						x						
Classification				x					x			x
Sensor interpretation								x	x			
Real-time control			x				x	x				

Table 1. Application features addressed in each chapter

In this book, we look at a number of case studies in which uncertainty formalisms are used in applications. These case studies can be grouped in a number of ways according to application features. In the following we discuss some of the key application features that are considered in one or more case studies. In Table 1, we identify which application features are addressed in each chapter.

Embedded KBS. Knowledge-based systems are frequently used within a larger software system. Here, high-level knowledge, such as rules, can be used to execute management or control activites in the wider software system.

Intelligent interface. Accessing information in computers can be difficult, so there is a need for computer interfaces to interact with users in a way that can help the user locate the information required, communicating in ways that are more appropriate for users.

Decision-support systems. The term "decision-support system" covers a wide range of tools designed to help end-users make decisions. This assistance can include provision of relevant information, reasoning with information to make arguments for possible decisions, and identifying qualifications, ramifications, or risk associated with possible decisions.

Information retrieval. The aim of information retrieval is to provide a user with information that best meets the user's request. This function involves various forms of uncertainty including determining, in some sense, the contents, of each document or article in the system, and determining the user's actual needs from the request.

Classification. Determining the classification for an instance given a number of attributes or observations can be a difficult problem in many spheres. Classification schemes can be complicated and interpreting them can involve resolving ambiguity, and handling the information about attributes or observations can require dealing with features such as incompleteness, inconsistency and imprecision.

Sensor interpretation. Devices are being designed with increasing numbers of devices incorporated. This is creating increasing pressure for automated means for interpreting the potentially enormous volume of data. The range of

Problem area	Chapter											
	3	4	5	6	7	8	9	10	11	12	13	14
Biomedicine	x				x						x	
Biology				x								x
Telecommunications			x				x			x		
Manufacturing									x			
Mobile robotics								x				

Table 2. Problem areas addressed in each chapter

devices incorporating sensors is very diverse and ranges from engine management systems through production line quality control systems and security systems, to mobile robots.

Real-time control. Software is increasingly being used to provide real-time control of systems ranging from avionics to chemical plants, and robots to medical monitors. A key compromise here is between correct/optimal inference in each scenario the system is used and the cost of engineering the system—so for example the approach of fuzzy control systems is popular for an application when it is relatively cheap to develop, and acceptable to use, a system that provides slightly sub-optimal reasoning.

The cases studies in this book show how an application feature can be potentially addressed using an uncertainty formalism. In Table 2, we then categorize the case-studies according to problem area. These areas are manufacturing, mobile robotics, telecommunications, biomedicine, and biology.

3 Technology questions

In this book we focus on applications where there is a need for the representation of information in a readily accessible and transparent form for the engineer or end-user. This need is prevalent in a wide range of systems including relational databases, decision-support systems, information retrieval systems, information filtering systems, and requirements engineering tools. For example, decision-support tools can be used by human decision-makers to enhance their performance. However, if the decision-support tool is opaque—as when the tool does not explain or justify its reasoning—then the confidence that the user has in the tool's output is decreased. Moreover, if the tool is opaque, then the user cannot qualify or adapt the output in the context of the wider sphere of information and experience that the user has access to.

The wide range of potential applications leads us to believe that we need a range of formal systems for handling uncertainty, and that each of these will incorporate a high-level language for representing uncertain information. In Table 3, we list the chapters in this book that use each of the main classes of uncertainty formalism.

Application feature	Chapter																		
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Dempster-Shafer theory			x			x			x										x
Probability theory	x									x	x				x				
Fuzzy set theory		x		x			x	x		x						x			
Default systems												x	x	x	x				
Modal logics																	x	x	
Graphical models	x									x						x			x
Argument systems					x						x								x

Table 3. Approaches used in each chapter

Whilst the case studies discussed in this book and elsewhere demonstrate the utility of a range of uncertainty formalisms, many application problems remain unaddressed. Current uncertainty formalisms provide foundations, but further technology questions exist in bridging the gap with applications. We do not aim to provide a comprehensive coverage of current technology questions, rather we have included some papers that are of particular relevance to applications, which are in the following technology areas.

Knowledge representation. Clearly very general formalisms such as classical logic can be used for representing any kind of knowledge. Unfortunately, for some applications this may involve much effort in determining the appropriate predicates, functions and constants, and in writing the appropriate formulae. Consider an application for reasoning with temporal information. What is the model of time? Does it use intervals or time points? Is the model linear or does it branch in the future? Therefore, to facilitate the development of knowledge-bases, there is a need for formalisms that contain the appropriate language and inference techniques for each application. This means more sophisticated and more specialized formalisms are being developed that are useful for particular classes of application. In the case of temporal information, there are specialized temporal formalisms that incorporate constructs for different kinds of temporal reasoning thereby simplifying the task of a knowledge-base developer.

Automated reasoning. Given a knowledge-base, automated reasoning is required to generate inferences or answer queries. In most uncertainty formalisms, this is expensive—often involving reasoning that is intractable or undecidable in the worst case. Therefore there is a need to develop efficient algorithms for sublanguages and for approximation techniques. Since there are many possible sublanguages and approximation techniques, these need to be developed with respect to classes of application.

Machine learning This is about inducing, or refining, knowledge in some formalism from other relevant information. Given the difficulty and expense of developing knowledge-bases—particularly with uncertainty formalisms—

machine learning offers algorithms and techniques to partially automate the process of knowledge engineering. Machine learning is a large topic in artificial intelligence and some key problems reside in the intersection with uncertainty formalisms.

These topics incorporate many open questions. In the following chapters, there is some discussion of all of them.

4 Conclusions

The roles of computing systems are clearly very diverse, and as a result addressing the problem of handling uncertainty in such systems is a broad subject. In addition, the problem of handling uncertainty in any kind of system is a complex and difficult task, requiring a range of formalisms, and there are many choices to be made in adopting an uncertainty formalism. As a result, the application of uncertainty handling techniques across the whole range of computing systems is a tremendously complex problem to deal with. In order to minimize problems arising from using uncertain information in an application, there is pressure to adopt a rich and powerful model, but the development and computational costs of such a solution may be too great. So there is always a balance to be struck. Through considering case studies such as those contained in this book, it is hoped that sufficient experience will be compiled to make it possible to strike this balance.

Since the aim of this book is to look at applications of uncertainty formalisms, it would seem appropriate to offer some form of matrix relating particular uncertainty formalisms to particular classes of application problem. However, uncertainty is difficult to formalize—witness the problems of just trying to categorize the different kinds of ignorance in [8][1]. Therefore we at least need to better grasp the notions within uncertainty formalisms before we have the conceptual apparatus to relate formalisms to applications, and it will be some time before it is possible to provide a definitive mapping between formalisms and applications. However, we believe that this book is a useful contribution towards identifying such a mapping by comparing and contrasting a range of formalisms in some diverse case studies.

References

1. P. P. Bonissone. Reasoning, Plausible. In S. Shapiro, editor, *Encyclopaedia of Artificial Intelligence*, pages 854–863. John Wiley & Sons, London, UK, 1987.
2. P. P. Bonissone and R. M. Tong. Editorial: reasoning with uncertainty in expert systems. *International Journal of Man-Machine Studies*, 22:241–250, 1985.
3. A. Hunter. *Uncertainty in Information Systems*. McGraw-Hill, London, UK, 1996.
4. P. Krause and D. Clark. *Representing Uncertain Knowledge: An Artificial Intelligence Approach*. Intellect, Oxford, UK, 1993.

[1] For a more extensive discussion of this problem see Chapter 2 of [6].

5. S. Kwan, F. Olken, and D. Rotem. Uncertain, incomplete and inconsistent data in scientific and statistical databases. In A. Motro and Ph. Smets, editors, *Uncertainty in Information Systems*. Kluwer, 1996.
6. S. Parsons. *Qualitative approaches to reasoning under uncertainty*. MIT Press, Cambridge, MA, 1998.
7. Ph. Smets and A. Motro, editors. *Uncertainty Management in Information Systems*. Kluwer, Boston, MA, 1997.
8. M. Smithson. *Ignorance and Uncertainty: Emerging Paradigms*. Springer Verlag, New York, NY, 1989.