Current approaches to handling imperfect information in data and knowledge bases

Simon Parsons

Advanced Computation Laboratory, Imperial Cancer Research Fund,
P.O. Box 123, Lincoln's Inn Fields,
London WC2A 3PX, United Kingdom.

Department of Electronic Engineering,
Queen Mary and Westfield College,
Mile End Road, London E1 4NS, United Kingdom.

Abstract

This paper surveys methods for representing and reasoning with imperfect information. It opens with an attempt to classify the different types of imperfection that may pervade data, and a discussion of the sources of such imperfections. The classification is then used as a framework for considering work that explicitly concerns the representation of imperfect information, and related work on how imperfect information may be used as a basis for reasoning. The work that is surveyed is drawn from both the field of databases and the field of artificial intelligence. Both of these areas have long been concerned with the problems caused by imperfect information, and this paper stresses the relationships between the approaches developed in each.

1 Introduction

Imperfect information is ubiquitous—almost all the information that we have about the real world is not certain, complete and precise. Thus to insist on studying just certain information, as has been the case in most work in databases, is to concentrate upon a small part of the whole problem. As Smithson [148] points out, much of Western philosophy has been taken up with building attractive models that are based upon idealisations that are never approached in reality. This he claims has led to the study of uncertainty being marginalised, even to the extent that it is defined only in terms of negative connotations—that which is not certain, those things that may not be known exactly—rather than being accepted as the natural state of all information. The fact that the study of uncertainty has not, in general, attained intellectual respectibility in turn has meant that the norm is to attempt to model the real world using some idealisation by *engineering out* the inherent uncertainty [24].

This means that one ends up with an elegant model, but one which can never give completely correct answers because it does not attempt to model precisely what is going on [137]. Instead, one should take the uncertainty into account, trading the loss of elegance and simplicity for more accurate modelling.

Motro [97] summarises this argument with the terse statement that:

> Uncertainty permeates our understanding of the real world. The purpose of information systems is to model the real world. Hence information systems must be able to deal with uncertainty.

As a result, Motro is interested in how imperfect information may be represented in a database. Turtle and Croft [154] in their discussion of uncertainty in information retrieval systems also argue that the issue of imperfect information cannot be ignored. However, their interest is slightly different. In information retrieval systems it is not so much the representation of imperfections in stored information that is of concern, as taking imperfections into account when deciding what information to retrieve. In other words reasoning with imperfect information is also important, and such reasoning will also be necessary in an expert database system [3, 80], as well as any deductive database.

Thus, to build useful information systems[1], it is necessary to learn how to represent and reason with imperfect information. In order to do this, it is necessary to learn a little about what it is and where it comes from. As a result this paper begins with a discussion of classifications of imperfect information, and some analysis of the sources of the imperfection. Both of these topics are covered in Section 2. This is followed by a detailed survey of some of the means of representing and reasoning with imperfect information that have been proposed. Many of these proposals have come from people directly interested in databases, but there is also a good deal of relevant work that has been carried out in the field of artificial intelligence. This work is especially relevant given the need [54] to integrate reasoning with data management to create the next generation of information systems [3]. As a result, this paper attempts a broad survey of work carried out in both fields, with work from the database field in Section 3 and that from artificial intelligence in Section 4. Note that the paper is mainly concerned with imperfection in data rather than imperfection in the properties of data, such as would be introduced by vague integrity constraints, or in queries made by users who are not completely sure of what query they want to make.

Now, I had hoped to be able to draw a sharp boundary between work on storing imperfect data in databases and work done on reasoning with imperfect data in artificial intelligence. However, the distinction is a lot less clear cut, with authors in both fields writing about both topics, and with some fields, such as logic programming, having close connections with both deductive databases and artificial intelligence. Despite this, I believe that it is worth making some distinction between the two camps in order to demonstrate what each may

[1] We will use the term *information system* as a suitable gloss for a system which is either a database in the sense of the database world, or a knowledge base in the sense of the artificial intelligence world.

learn from the other. As a result I have maintained the division by following the claims of the authors, so that work which the author claims is concerned with databases may be found in Section 3, while work which the author claims is more to do with the study of uncertainty in artificial intelligence will be found in Section 4. Within each section the approaches are divided up along the lines of the classification introduced in Section 2.

Before beginning, it is making a brief aside on the use I have made of the word "uncertainty". Throughout the literature this term is overloaded, being commonly used both as a generic term for imperfection in data, as in the quotation from Motro, and as a term for a particular form of imperfect knowledge of whether or not a statement is true. There is no easy way to resolve this overloading. I have attempted to use "imperfection" and "imperfect" in the generic sense, and "uncertainty" and "uncertain" in the specific, but this has not always been possible (as in the case of the quotation) and when possible has involved reinterpreting what others have said. I hope that in the main I have avoided confusion and trust that any errors I have introduced will be forgiven.

2 Imperfect information in general

Over the years there have been many attempts to produce a categorical classification of the different types of uncertainty, and to elucidate the relationships between them. Several of these manage to seem entirely self-consistent and intuitively plausible whilst managing to be mutually inconsistent suggesting that there is no one best classification. However, it is worth considering a couple in order to get a feel for the subject.

2.1 General classifications of imperfect information

One of the earlier classifications is that of Bonnissone and Tong [16] who advocated the point of view that there are three types of imperfection that might be found in an information system. These are *uncertainty*, *incompleteness* and *imprecision*. Incompleteness arises from the absence of a value, imprecision from the existence of a value which cannot be measured with suitable precision, and uncertainty from the fact that an agent has constructed a subjective opinion about the truth of a fact which it does not know for certain.

Incompleteness can be existential, as when a particular instance of the value of an attribute is unknown. Thus the statement that: "The author of 'Naked Lunch' is an important figure in twentieth century literature" is an example of existential incompleteness since the author's identity is left unknown. Alternatively incompleteness can be universal, when all instances of particular attributes are unknown as in the statement: "Sylvia Plath wrote some very moving poems" which fails to say what the names of the poems are.

Imprecision can be interval valued, as in the case that the author's age at the time of writing is given as "between 25 and 30", or fuzzy valued, as in the case that the author is said to be "quite young". It is also possible to have a discrete form of imprecision, such as that arising from disjunctive information as in the statement that "the author is either 26, 27 or 28", and there is a form of imprecision that arises from negation. If all that we know is that "Simon is not married to Ann", very little may be said about his marital status since he could easily be married to someone else.

Both imprecision and incompleteness as defined here are objective to some degree. They stem from limitations in the way that quantities are measured, and reflect the absolute nature of the available information. In contrast uncertainty is seen by Tong and Bonnissone as being inherently subjective. Uncertainty is an estimate of the truth of some fact by some individual. The estimate may be made by estimating the probability that some proposition is true, by making some statement about one's belief that the proposition is true, or by using some form of epistemic possibility or necessity that gives, for instance, the degree to which the author's age may be, or must be, 26, given that he is young.

Bosc and Prade [15], echoing an earlier classification made by Dubois and Prade [40], are in broad agreement with this classifying imperfections as being due to *uncertainty*, *imprecision*, *vagueness* and *inconsistency*. However, the definitions of some of these terms are given slightly different emphasis. Thus uncertainty arises from a lack of information about the state of the world. This lack of information makes it impossible to determine if certain statements about the world are true or false—all that can be done is to estimate the tendency of the statement to be true or false either by using some numerical measure of the degree to which one may be sure, or may speculate, that the statement is true.

Imprecision is considered as arising from the granularity of the language used to make the imprecise statements. Thus the statement that "Simon is 26 years old" is precise only if we are not interested in Simon's exact age in terms of years and months. In general, imprecise information is represented as non-singleton subsets of values from a given domain, and in extreme cases may encompass every possible value of the domain, as in the case in which Simon's age is completely unknown.

Bosc and Prade also make the important point that uncertainty and imprecision may arise together in the same piece of information. Thus a statement about Simon's age may be imprecise, such as "between 26 and 28", but this information may not be certain since its certainty will depend upon the knowledge of the person making the estimate. However, there is a heuristic connection between precision and certainty since very imprecise statements such as "Simon's age is less than 50" have a greater chance of being correct than precise ones.

Vagueness is a new category, and in Bosc and Prade's terms is essentially fuzzy valued imprecision as classified by Bonnissone and Tong. Thus a vague statement contains some vague predicate such as "young". This information may be used in a number of ways. For instance, given the statement that "Simon is quite young", we can use this information to say something about the certainty of the uncertain statement that "Simon is 28 years old". Alternatively, the statement could be used to establish the range of possible values for Simon's age, based on a fuzzy set interpretation of the term "young". Furthermore, if we know that the statement "Simon is 28" is true, then the statement "Simon is young" will be true to some degree.

Inconsistency is also a new categorisation, and describes the situation in which there are two or more conflicting values for a variable, for instance "Simon is 26" and "Simon is over 30". In such a case there is no possible way of merging the two pieces of information to obtain a consensus value since there is no value that is consistent with both, and it is possible that the inconsistency has arisen from a number of sources of information. In such a case the only solution is to retract the information from the least reliable sources, given of course that it is possible to determine which these are.

Combining these two slightly different means of classifying imperfections in information, we can split the area into five seperate parts. We have uncertainty which arises from the lack of information about the real world, and which may be due to subjective error on the part of some observer. We have imprecision, which arises from a lack of granularity and may be disjunctive, existential or universal, but which is distinguished from vagueness. We have incompleteness, which is simply a lack of relevant information, and we have inconsistency, which arises from having too much information from too many sources. In addition to these we also have the term "ignorance", which is used in a number of different ways in the literature, but which we will use to describe a lack of knowledge, particularly a lack of knowledge about the relative certainty of a number of statements.

2.2 Types and sources of imperfect information in databases

Having established some kind of classification, we can use it to attempt to make some overall sense of the kinds of uncertainty that people have studied with respect to databases. The types of uncertainty that may be found specifically in relational databases are the concern of Motro [97], who builds upon his earlier work [98]. Within relations the value of individual data items may be unknown, the applicability of a tuple to the objects contained in it may be vague, and relations themselves may be uncertain. Thus the salary of an employee may be unknown, the assignment of employees to projects may be undefined, though "employee" and "project" are themselves well defined, and the relationship between employee and department might be uncertain because it might not be known which of a possible range of departments some employees belong to. Queries may also introduce uncertainty, since the relevance of particular queries to the objects in the database might not be certain. Thus in terms of our classification we may have incomplete, vague and imprecise data, though the imprecision may well be related to some degree of uncertainty. Motro also identifes the sources of these imperfections. They may result from unreliable sources, such as faulty sensors, input errors, or the inappropriate choice of representation.

Further discussion of the problems in compiling information comes from Kwan et al. [76] who suggest a number of sources of uncertainty and incompleteness in databases used in scientific applications, taking a database of the human genome as an example. They suggest that some data is recorded statistically and so is inherently uncertain, whilst other data is deliberately made uncertain for reasons of security. Other data may not be measured accurately, due perhaps to some quantum mechanical effect, and so will include some irreducible uncertainty—a point of view which concords with mine in suggesting that "perfect information" is an unattainable ideal.

In all of these cases there is a one-to-one mapping between *real* values and *measured* values. However, Kwan et al. also claim that it is very likely that some values will be missed because of imperfect measuring equipment. The different types of uncertainty vary from application to application. Thus incomplete data, due to an inability to record every desired value, will characterise biomolecular databases, and these also have problems with measurement noise, while evasive and untruthful responses from people worried about higher taxes are apparently a big problem in introducing uncertainty into economic data.

Uncertainty in information retrieval is considered by Turtle and Croft [154], who identify three areas in which they claim uncertainty may be found. Firstly there is the problem of the representation of a document. Given the text of a document, humans cannot agree on the set of concepts that should be used to represent it and automated procedures for creating descriptions will introduce considerable uncertainty. Furthermore, difficulties arise when attempting to represent the degree to which a document addresses the concepts in its description. The second problem is the representation of the kind of information that a user needs to retrieve. Modelling this need is especially difficult since it typically changes during the search process. Thirdly, it is necessary to match needs to concepts, and the process of doing so would be likely to be approximate even if content and needs were precisely represented since the needs would not necessarily map cleanly onto the concepts, for example since they may be expressed in different languages. In our terminology, the problems do not seem to be so much of uncertainty as of vagueness and imprecision. The concepts by which a document is indexed, and the description of the document that the user wants to retrieve, are limited by the precision of the terms in which the descriptions are couched and the vagueness of the concepts. However, it is quite likely that some form of uncertainty will be introduced in that the imprecision and vagueness will be subject to some degree of subjective opinion.

3 Imperfect information in databases

Having attempted a form of classification, we will try to use some elements of it in order to carve up the work that has been done on modelling imperfection in a database setting into manageable chunks. Thus we will consider, in turn, storing incomplete information, which has strong links with work on non-monotonic reasoning [55], imprecise information, which has been handled by various applications of fuzzy sets [168] and fuzzy logic [165, 166], and uncertain information, which has been dealt with using probability [48], possibility [40, 167] and Dempster-Shafer theory [141]. It should be noted, however, that the different forms of imperfection cannot be so cleanly seperated as this description suggests. For instance, the fact that data is imprecise will lead to queries that are made upon it returning answers that are uncertain. In contrast, if data were precisely known and the queries were imperfect, the answer generated by the query would not be uncertain, but would only imprecisely match the query.

3.1 Incomplete information

Much of the work on handling imperfect information in databases has centered on the question of how to handle information that is imperfect due to its incompleteness, and most of this has been concerned with how to handle missing attribute values. As a result quite a number of schemes have been proposed, most of which centre around the *null* value, a placeholder for the missing value which was initially introduced by Codd [23]. While the use of a null value seems a very sensible way of handling the problem of representing incompleteness, it introduces a new problem— interpreting what the null value is representing.

A number of possible interpretations have been suggested, including "unknown" [23] when the value is assumed to exist but is not known, and "nonexistent" interpretations [81]. In addition there is the "no information" interpretation [169] under which a null value is merely a placeholder for a value that is either unknown or non-existent and interpretations that take null as being a value that is "undefined", "inapplicable" or "non-existent". For each of these interpretations there is a relational algebra which takes the interpretation into account in an appropriate way when answering queries. The use of null values has also been considered in the context of object oriented databases [171] with the conclusion that missing values of object attributes can be handled in the way that null values are handled in relational databases, although the added complication introduced by inheritance must also be taken into account.

Related to this work on missing attribute values are the efforts of Lipski [84] who concentrated upon the problem of incompletely specified attribute values, and contrasted the mismatch between the common occurrence of incomplete information and the fact that at the time no commercial databases provided support for incomplete information. Lipski was concerned with the problem of answering queries such as "List all blue objects" where not all objects in the database have their colours listed. Thus all that can be said is that there are some objects that are known to be blue, and some that are known not to be blue. These two sets of objects bound the possible answers to the query in a way that suggests similarities with modal logic [67] and rough sets [111, 112]. Whilst the procedure of establishing the bounds is simple enough for elementary properties, such as colour, it becomes much harder for queries involving combinations of elementary properties. Lipski provides the basis for a system capable of answering these more complex queries, giving a simple query language and algorithms for establishing the bounding values.

In a similar vein, Imieliński and Lipski [69] discuss extensions to Codd's model of null values in which unknown values are replaced by variables. The use of variables makes it possible to distinguish the situations in which values are unknown but known to be equal and those in which they are unknown and known to be unequal. The authors show that while Codd's system supports projection and selection it is unable to support a combined "project and join" operations. They also show that their system can support projection, selection, union and join but cannot, unlike Codd's, support combined "project and select". Finally they introduce a new scheme, in which the admissable values of the variables are made explicit, and discuss how this can overcome the "project and select" problem which their original system suffers from.

Further problems with incomplete attribute values are created by deductive databases and these are considered by Williams and Kong [161]. Not least among the problems is the interpretation of what incompletely specified deductive rules mean, and thus what they might be used to infer. The suggestion made by Williams and Kong is that an incomplete set of rules be considered to represent all the complete sets that may be established by replacing every missing value by every possible legal value. Allowable inferences are then those that are sanctioned by the set of sets, and they distinguish between those that may be made from every set, and are thus definite, and those that may be made from at least one set and are thus possible. Later work extended this concept to cover time as well [160], and defined an extended SQL for deductive databases which allows the manipulation of both facts and rules [158].

Another approach to handling deduction in the context of incompleteness is suggested by Demolombe and Fariñas del Cerro [31] who advocate the extension of relational algebra with Skolem constants. They devise and fix a number of problems with a naive approach to achieving this, proving that their system is sound and complete for a reasonably unrestricted class of formulae.

Clearly, the kind of querying it is possible to achieve with deductive databases is more complex and powerful than that possible with ordinary relational databases, and the computational complexity of the kind of reasoning one can perform with deductive databases which handle can incompleteness is the central concern of [68]. In this paper Imieliński provides a characterisation of incomplete deductive databases and finds that in general inference in such databases is undecidable[2]. However, he does identify some classes of database, defined in terms of the characteristics of the rules that they contain, for which the complexity results are more promising, especially given the complexity of simple queries on such databases [1].

Despite their widespread use, explicit null values are not the only means of handling incomplete information since it is also possible to try to "fill in" the incompleteness in some way. This is especially possible when questions which elicit yes/no answers are posed. In this case there are two basic ways of making an assumption about the unknown value and using this to reply to a query. The first is to implicitly acknowledge that some information may not be present. This is known as the *open-world assumption*, and in some circumstances will mean that a query will not be answered because the database does not conatin the relevant information.

A more useful alternative is discussed by Reiter [131], who was the first to formalise the *closed-world assumption*—the idea that when the database cannot answer a particular query it assumes that the answer is 'no'. As Reiter shows, the closed-world assumption is not infallible, but it does not produce

[2] This result is not very surprising since deductive databases are an implementation of a subset of first order logic which does not improve on the semi-decidable nature of logical inference.

inconsistent answers for databases that are expressed as Horn clauses, and so is a reasonable assumption for many cases. Clearly the closed world assumption is closely related to 'negation as failure' [21], the procedure that a Prolog system uses in order to decide that a goal is not true if the system fails to prove it, and schemes for non-monotonic reasoning developed in artificial intelligence which are discussed below. The use of open and closed world assumptions also provides a way of interpreting null values [170] in which the assumptions may either be global and fixed, or local and dependent upon where the null value is encountered.

Finally, it should be noted that there are other forms of incompleteness than missing attribute values. As Zicari [170] points out, the use of a relational database scheme implies that data has a clearly defined structure, that this structure fits into a relational schema, and that it may be represented by values of attributes in the schema. This set of assumptions may fail either because the structure of the data is incompletely specified, or because the data does not fit into the relations, as well as because attribute values are not known, and the former two problems are yet to be addressed.

3.2 Imprecise information

Most of the work on the modelling of imprecise information within databases has involved the use of fuzzy sets [168] and fuzzy logic [165, 166]. Fuzzy set theory is a generalisation of normal set theory in which it is recognised that the kinds of classes of objects one encounters in the real world do not always have precisely defined criteria of membership. Thus it is clear that the class of living things should include people, dogs and trees and should not include roads and quasars, but whether viruses should be included is more controversial.

One way to resolve this problem is to attach a *degree of membership* to every object which indicates the degree to which it is a member of a given set. Thus dogs would be members of the set of living things with degree 1, quasars would have a degree of membership 0, while viruses would have a degree of membership somewhere in between. Degrees of membership may be combined, so that it is possible to compute the degree to which objects are members of logical combinations of sets. Degrees of membership may be applied to predicates in first order logic to form a fuzzy logic, and relations can be fuzzified in a similar way.

There are four distinct ways in which fuzzy notions may be applied to handling uncertainty in databases. The first is to associate a fuzzy degree of membership with each tuple of a relation. This is the approach taken, for instance, by Baldwin and Zhou [10]. As Dubois and Prade [37] point out, such a degree may be interpreted in a number of ways. It may be taken to be a degree of association between the elements of a tuple, that is the degree to which they all belong together in the tuple, as a measure of confidence about the information that is stored in the tuple, that is as a measure of the certainty of the information, and as an estimate of the degree to which the tuple is a typical example of the relation to which it belongs. Baldwin and Zhou opt to take the degree of membership attached to a tuple to be the degree to which it satisfies the relation it belongs to, and construct a fuzzy valued algebra for manipulating collections of relations to create new relations in response to queries. This approach has been automated in the programming language FRIL (see Section 4.2).

A very similar approach is that of Zvelli [172] who handles the problem of representing uncertain information by generalising the relational database model to include fuzzy relations, introduced by means of a fuzzy first order predicate logic. In doing this he essentially fuzzifies the whole of the relational model, giving fuzzy relations in which every member of a tuple has a degree of membership, fuzzy structures in which object names are mapped to fuzzy objects, fuzzy assignment of variables to fuzzy sets, and fuzzy formulae with weights on clauses and the use of fuzzy connectives. All of these are features one would expect from a fuzzy logic. Zvelli also introduces operations which transform relations into semantically related relations (similar to fuzzy similarity) and fuzzy satisfaction where a fuzzy formula satsifies a fuzzy structure to some degree. This work, however, stops short of defining a fuzzy query language.

The second basic approach is to use a fuzzy similarity relation to measure the extent to which the elements of an attribute domain may be interchanged. This is the approach taken by Buckles and Petry [17, 19] while Prade and Testemale [124] describe a related idea using fuzzified rough sets which also encompass the modelling of incomplete information. Under Buckles and Petry's approach, a relational database is augmented with an explicit record of the degree of similarity between certain attribute values in a particular domain. These values are created by the database designer, but have to conform to certain rules which ensure that the relation is reflexive, symmetric, and has a form of transitivity. Since an equivalence relation would be reflexive and symetric, but have a more limited form of transitivity, it is clear that a similarity relation is a generalisation of an equivalence relation. The similarity measures allow the usual relational operations to be extended to give fuzzy solutions, so that the result of a select operation, for instance, will not only give the records that exactly match the query, but also those that match it to some degree.

The advantage of using similarity relations is that they enable a form of fuzzy pattern matching to be used when answering queries. Thus, to use Buckles and Petry's baseball example, when searching for a replacement leftfielder, a manager would want to consider possible right or centerfielders since outfielders are almost completely interchangeable. Similarly when choosing a replacement shortstop, it is quite likely that a second baseman could do a good job, but a catcher would never be a possible replacement for a pitcher. All of these possibilitites could be offered by the system by giving the appropriate similarity measures.

Shenoi and Melton [142] extend the work of Buckles and Petry by replacing similarity relations with proximity relations. These relations are a generalisation of similarity relations, which result from deciding to partition a scalar domain with reflexive and symmetric relations. Thus the transitivity of Buckles and Petry's similarity relation is dropped, since, it is argued, it is a very restrictive constraint. The result is that all the "nice" properties of Buckles and Petry's model are retained, so that Shenoi and Melton's model is at least as good as its predecessor and, if one rejects transitivity, is better. However, it can equally be argued that the transitivity of similarity relations is not too restrictive for certain properties so that its applicability is dependent on the semantics of the attributes to which it is applied—a point that seems to have escaped both Buckles and Petry and Shenoi and Melton.

Another extension is covered in [18] in which the basic framework is altered to allow fuzzy numbers to be used to specify the value of attributes. This necessitates the extension of the fuzzy relational algebra that is used to answer database queries to allow the combination of fuzzy numbers, but this work is justified by the additional functionality, such as the ability to establish the average of a particular domain, that is provided.

A further extension to the similarity-based approach is provided by George et al. [54] who consider applying it to an object-oriented data model. In order to do this they first fuzzify the notion of a class heirachy so that a given class need only be a subclass of another to some degree. This in itself is a useful notion given the difficulty with which some class/subclass classifications are made, and allows the definition of a fuzzy inheritance mechanism so that the degree to which some object is a member of its superclasses can be determined. On top of this George at al. build the similarity mechanism, making it possible to provide answers that partially match queries as in the relational similarity systems, but which also take inheritance into account. Thus, when returning the set of all young research staff from a university database, the system would both take into account the fuzziness of the term "young" using similarity matching, but would also return appropriately aged members of the class of graduate students since the latter is a fuzzy subclass of the former.

The third fuzzy approach is to make use of fuzzy inference mechanisms. Thus Leung et al. [80] use fuzzy production rules coupled to a standard relational database to create a deductive "expert database system" which can handle imprecise information. The expert system uses fuzzy production rules to answer queries, calling the database system for information when required. However, the database itself does not have any fuzzy information in it—instead fuzzy queries are satisified by making several exact queries, the translation between fuzzy and exact queries being carried out by the expert system. A similar procedure underlies Guardalben and Lucarella's information retrieval system [58]. Fuzzy inference is used to determine which documents are most relevant to a request for information, and a query is then formulated for a document base.

The final approach which is strogly related to the use of fuzzy sets is to use possibility distributions, which are based upon fuzzy restrictions, to represent uncertain information. Such possibilistic approaches are covered in the next section. Before we pass on to look at uncertain information, however, it is worth considering two other, unfuzzy, approaches to dealing with imprecise information.

The first is provided by Morrissey [95, 96], who is concerned with the representation of disjunctive information about a single valued attribute, for instance knowing that Bill's phone number is one of 909281, 904131, or 909591. This is handled by allowing attributes to have sets of possible values, and queries are answered by supplying "possible" and "known" values in a way reminiscent of

Lipski [84]. The most interesting thing about this approach is that Morrissey then proceeds to quantify the uncertainty that this imprecision introduces by using techniques from information theory, effectively hybridising Lipski's work with more numerical methods. He provides two measures for the uncertainty of an answer to a query—one assesses how much more information would be needed to be certain that the attribute in question exactly fulfils the query, and the other estimates the entropy involved in deciding that the attribute satisfies a query. Both measures make the strong assumption that *a priori* all values of the attributes are equally likely.

The second approach is suggested by Gunter and Libkin [59] who discuss the same problem from a different angle. They are interested not in the kind of queries investgated by Morrissey for which the necessary information is present in the database, but rather in queries that can only be answered by reasoning with the disjunctive information that is present.

3.3 Uncertain information

Uncertain information is typically handled by attaching a number, which represents a subjective measure of the certainty of the uncertain element according to some observer, to that element. The way in which the number is manipulated depends upon the theory that underlies the number. We will consider approaches based upon possibility and probability theories, the latter including those approaches that make use of Dempster-Shafer evidence theory.

3.3.1 Possibilistic databases

Possibility theory [167] is built upon the idea of a fuzzy restriction. Consider a variable that is constrained to take its value from some fuzzy set of values. Any value within that set is a possible value for the variable. However, since the values have different degrees of membership in the set, they are possible to different degrees. A value that has a degree of membership of 1 will be completely possible as a value, while a value that has a degree of membership of 0.1 will be much less possible. In fact, the degree of membership of the set of a particular value is taken to be the possibility of the variable taking that value. The use of a possibilistic approach has been developed over a number of years by Dubois, Prade and Testemale and some of their work is discussed below. For another view of this, and the relation between it and other methods, see [15].

The initial method suggested by Prade and Testemale [123] is the most obvious one—to attach a possibility degree to every value of every attribute, so that one may represent uncertainty about the age of Simon's car by giving a possibility distribution over the set of possible ages. They also allow the inclusion of the null hypothesis in the set of values over which the possibility distribution is defined, so that it is possible to attach a value to the hypothesis that "Simon does not have a car" as well as the hypothesis that "Simon's car is 5 years old". Thus the approach explicitly considers the modelling of incomplete information, and by sleight of hand transforms a possibly incomplete database into a complete one. Now, given the possibility measures of the degree to which certain values are held, the query language given by Prade and Testemale allows the degree to which a certain tuple must and may satisfy a condition to be determined. This approach is reminiscent of that of Lipski [84], but quantifies the degree of satisfaction.

This work is extended in [125] to cover multivalued attributes. Thus the values over which the possibility distributions are defined may be fuzzy, so that, for instance, we could store information about the degree to which "Simon is old" is known to be possible. This represents a useful generalisation of the original approach since it allows the method to be applied to "linguistic" values, making it possible to argue that information represented in a form of natural language may be represented.

The paper also discusses ways to represent constraints on the value of an attribute and between the values of two attributes in such a way that the constraint is not strictly enforced, but merely serves to alter the possibility of certain values being adopted, and a query language is developed that can manipulate the possibility values. This generalised method is then applied to the problem of retrieving documents using vaguely specified information [126]. The approach is related to fuzzy pattern matching [43].

Prade [122] discusses the ways in which the use of possibility theory to model uncertainty in databases is related to the approach taken by Lipski [84]. He shows how Lipski's approach may be captured by using possibility degrees of 0 and 1, and then generalises this to use intermediate degrees. Results are given for establishing the upper and lower measures in the cases that the attributes on which the query is made are independent, dependent, or have some fuzzy relation that relates them.

3.3.2 Probabilistic databases

The simplest possible method for using probabilities to quantify the uncertainty in a database is that of attaching a probability to every member of a relation, and to use these values to provide the probability that a particualr value is the correct answer to aprticualr query. This is exactly the course proposed by Wong [162] who shows how the relational model may be modified to take account of probabilistic information and to provide answers to queries.

Cavallo and Pittarelli [20] take a similar course giveing projection and join operations for relational databases systems augmented with probability measures. They also discuss in some depth the use of information theoretic measures of information, in particular Shannon entropy, and show how the information loss introduced by projection and join operations may be calculated. Throughout the paper Cavallo and Pittarelli stress that their system is completely general, and may be used to combine probabilistic and fuzzy approaches, but unfortunately they give no inkling of how this may be done. In later papers the approach is extended to cover the case in which only the bounds on the probability of an event are known [115], and to extend the relational algebra to provide decision support [114].

Another similar approach, but one applied to object oriented models, is proposed by Kornatzky and Shimony [74]. As one might expect, the uncertain values of attributes are modelled by allowing them to have ranges of values, including the null value, across which a probability distribution is defined. However, this is not all that Kornatzky and Shimony offer. They are interested in uncertain inheritance as well, and so want to be able to find the probability that a given object is a member of class C_i given that it is a member of C_j. They provide a mechanism for doing this, provided that the class heirachy does not allow multiple inheritance, along with a system for answering conjunctive queries about the probabilities of particular object attributes having particular values.

Other relevant work has been carried out by Ng and Subrahmanian who have considered the use of probabilities in deductive databases and logic programs [101, 104]. To do this they allow logical clauses to be annotated with a probability interval, and provide rules for establishing the bounds on combinations of clauses, that do not make the kind of restrictive assumptions about the independence of clauses required by previous efforts [46]. The propagation rules are backed up by a full proof procedure, fixpoint theory, and a formal model theory which is shown to be probabilistic.

This work is extended in [103] to cover the use of nonmonotonic negation, which makes it possible to capture some kind of default reasoning, and in [102] to cover objective probabilities. Now, it might not seem that objective probabilities raise any problems not covered by a scheme that can handle subjective probabilities, but this is not the case (due to a technical hitch with Herbrand universes). Despite this problem Ng and Subrahmanian provide a means of answering queries and ensuring the consistency of the database when objective probabilities are used.

Barbara et al. [11] develop a rather different probabilistic model. Given a particular relational tuple, it is possible to specify a probability distribution for the values of a given attribute. Thus if it is precisely known that employee John Smith is in the Toy Department, but there is uncertain knowledge about his sales for the year, this information is modelled by providing a probability distribution across the possible values. The paper gives a definition for suitable Project, Select and Join operations which largely respect the probabilistic semantics given to the relations—where they differ it is because of ignorance about the distribution, and this seems acceptable. What is particularly interesting about this approach is that Barbara et al. are concerned with how to handle ignorance about what probabilities to attach to particular facts, and re-invent a part of the Dempster-Shafer theory [141] in order to do so.

Writing a few years later, and in apparent ignorance of Barbara et al.'s work, Lee [79] deals with the same problem and proposes the same solution[3]. He, however, is aware of Dempster-Shafer theory, and uses it to define a general relational algebra that allows the belief and plausibilities of complex queries to be established from those of their constituents. The modelling of ignorance is also considered by Schocken and Hummel [140] who also use the Dempster-Shafer

[3] "Hegel remarks somewhere that all the great events and characters of world history occur, so to speak, twice." (Karl Marx, The Eighteenth Brumaire of Louis Bonaparte)

theory. Whilst their concern is to pool expert opinions about the relevance of documents in a retrieval system, many of their concerns are equally applicable to other database systems.

Another type of probabilistic model is due to Van Rijsbergen [132] who introduces the idea of using a logic for information retrieval, that is a logic that matches keywords in a query to keywords in a document to decide which document in a large collection is most relevant to a particular user. To do this he introduces a form of implication "$\rightarrow$", which is undefined but distinct from material implication, such that:

$$\{set\ of\ keywords\} \rightarrow query$$

This implication is then associated with the conditional probability of the query given the keywords, $p(query \mid \{set\ of\ keywords\})$, and a possible world semantics introduced in order to allow the estimation of $p(x \mid y) = p(y \rightarrow x)$. This semantics is dependent upon the notion that the certainty of $y \rightarrow x$ is dependent upon the amount of additional information that is required to establish the truth of $y \rightarrow x$. Van Rijsbergen's idea is extended by Nie [105] who brings in ideas from modal logic, associates a possible world with the list of keywords, and then uses a generalised accessibility relation to estimate the degree of correspondence between the keywords.

Finally, we should briefly mention the work of Kießling and colleagues [71] who have introduced uncertainty into inheritance relations in a similar way to that suggested by George et al. [54], but using a probabilistic measure of the degree to which it is certain that a property is inherited.

3.4 Summary

To summarise, there are four basic ways in which we can handle imperfect information and all the methods we have discussed involve variations on one or more of them. We can: (1) use a number or symbol to indicate the degree to which a given attribute is known to satisfy a relation, (2) use a number or symbol to indicate the strength of the relation between attributes, (3) use a number or symbol to indicate the strength of inheritance, or (4) derive the appropriate number or symbol to result from a query. Different methods simply provide different mechanisms for doing some or all of these things, and give different meanings to the numbers or symbols that are provided. For instance, we can use null values to indicate that a given attribute (telephone number) within a relation (employee) has a value that is unknown, or we can use probability of 0.8 to indicate that a particular instantiation of an attribute (a particular number) in a given relation is known to be very likely to be correct. Often a choice of one way of modelling forces the choice of others. Thus the use of fuzzy sets to model the degree to which attributes satisfy relations will impose constraints upon the way in which various database operations, such as Project and Join, are carried out, and similar constraints are imposed by the use of the various types of null values to complete relations.

4 Imperfect information in artificial intelligence

As is the case within the database community, there is a split in the artificial intelligence community between those who deal symbolically with the problem of incomplete information, and those who deal numerically with the problem of imprecise and uncertain information. This split is largely historical, and stems from the schism between mainstream practioners of artificial intelligence who scorned quantitative approaches and those who championed numerical methods of handling uncertainty. As a result of the split, work on incomplete information and that on uncertain information has largely taken place with no regard for the other, and it is only comparatively recently that work has begun on relating the two. As a result much of the work on combining the approaches is rather preliminary, and we thus present the methods seperately.

Firstly we consider the major techniques for handling incomplete information, all of which are nonmonotonic logics. That is, in general, they are methods based upon classical logic in which the ususal inference mechanism is augmented with some method for making assumptions about missing pieces of information. Whereas classical logic has the property that adding formulae can never make old conclusions invalid, adding a formula in a nonmonotonic logic can violate an assumption and so cause a conclusion to be withdrawn (and hence the name). Thus, to some extent, non-monotonic logics are extensions of classical logics. However, there are also non-monotonic logics that are weaker than classical logic in that many of the theorems of classical logic are not true of them.

Having covered the main symbolic methods, we then take a look at the state of the art in numerical methods where there is work both on combining logical methods with numerical quantifiers and on the use of pure numerical techniques. The former has much to recommend it to those interested in deductive reasoning, but the mismatch between logical inference and the methods used to propagate the numbers can cause problems. As a result the community has concentrated largely upon purely numerical techniques. Much recent work has centred upon using network models to represent the dependencies between relevant facts, and the use of such models has been crucial in establishing the handling of uncertainty as a major subfield of artificial intelligence. However, the use of network models has some serious problems, and these and their proposed solutions are also discussed.

Finally, we turn to a closely related set of approaches that have been developed over the last few years. These are methods based upon systems of argumentation, and they provide a very general framework for reasoning that enable the representation of both numerical and symbolic information. They thus represent a form of integration between the approaches to incomplete and uncertain data, and thus seem particularly appropriate for use in deductive databases in which both types of imperfect information must be handled.

4.1 Incomplete information

From the point of view of the artificial intelligence community, the problem of handling incomplete information is bound up with the problem of representing information about typicality and prototypical objects. For instance, to take the usual example, we know that, generally speaking, birds fly. Thus, when confronted by a particular bird Tweety, one naturally assumes that Tweety can fly, although we do not know this for sure, and despite the fact that there are many types of bird (penguins, dead birds, and birds whose feet have been set in concrete are the most commonly quoted ones) that cannot fly. This assumption may be used to draw new conclusions, about the wisdom of letting Tweety out of her cage when the window is open for example, and as the basis for making decisions based on those conclusions.

However, it is also possible that the assumption is invalid, as in the case that Tweety's wings have been clipped precisely because of her predeliction for open windows, and we have no problem in revising conclusions and decisions in the light of this new information. Now, it turns out that this kind of reasoning, involving making assumptions and then revising beliefs in the light of new information, cannot be captured in classical logic because the monotonicity of classical logic prevents conclusions being withdrawn. As a result, the artificial intelligence community has invested a lot of time and effort in producing nonmonotonic logics.

In this section we present a brief description of three of the original nonmonotonic formalisms and some of their descendents. There are many other varieties of nonmonotonic logic. Some are covered in the collection edited by Ginsberg [55], others may be found in the books by Brewka [14] and Besnard [13] or the recent special issue of the *Journal of Applied Non-Classical Logics* [22].

One of the first nonmonotonic systems to be proposed was Reiter's default logic. This augments standard first order logic with a set of default rules which explicitly state what assumptions may be made. Thus, if we want to make it possible to infer that a given bird can fly, unless we have information to the contrary, we can write the default rule:

$$\frac{bird(x) : \mathsf{M} fly(x)}{fly(x)}$$

where the M is an operator read as 'it is consistent that', and "$bird(x)$" and "$fly(x)$" represent the statements "x is a bird" and "x flies", respectively. This default rule allows the tentative conclusion that an individual can fly to be drawn if it is known that the individual is a bird, and may be paraphrased "if it is known that x is a bird, and it is consistent that x flies, then assume that x flies". The statement of rules outside the logic that say how assumptions can be made is very appealing, and default logic has, perhaps as a result, proved very enduring.

It has, however, had some technical problems. The major one is that there is no constructive procedure for building all the consequences of a set of defaults, and this has frustrated attempts to build an efficient implementation. The other main problem is the interpretation of the notion of consistency that is implicit in the idea of a default. Given that first order logic is only semi-decidable, it is not necessarily possible to decide, for instance, that Tweety cannot fly, and so it is not necessarily clear whether it is possible to apply a default about Tweety's ability to fly.

Another early scheme for nonmonotonic reasoning is circumscription [89, 88]—an attempt to formalise closed world reasoning in a way that defeats some of the problems of simple techniques such as the closed world assumption [131] and negation as failure [21] where the failure to derive particular facts allows their negation to be assumed. Unlike other techniques, circumscription applies a set of rules of conjecture that are based on syntactic manipulation rather than appeals to undecidable provability or consistency.

Predicate circumscription [89] was the first variant to be introduced, and consequently is the most studied of the differing varieties of circumscription. Predicate circumscription allows explicit assumptions of completeness to be made as required, providing a way of applying a closed world assumption to a particular predicate at a given moment. A schema for a set of first order sentences is generated, and then instantiated replacing predicate variables with particular predicates, the choice determining the composition of the extension of the circumscribed predicate. This has the effect of asserting that the only positive instances of that predicate that exist are those known to exist at that time within the circumscribed domain. Any instances that are not known to be positive are assumed to be negative.

Predicate circumscription has been amended and refined to handle various technical difficulties that have been discovered in the years since the formalism was introduced. These include formula circumscription [88] which permits arbitrary predicate expressions to be circumscribed, and which forms the basis of a simple means of implementing commonsense reasoning which is commonly known as abnormality theory. This allows the "birds fly" example to be encoded as:

$$bird(x) \wedge \neg abnormal(x) \rightarrow flies(x)$$

and if penguins are distinguished as abnormal birds formula circumscription does not sanction the inference that penguins can fly. There is also pointwise circumscription [82] in which the circumscription, instead of being carried out everywhere simultaneously, is performed by minimising one point at a time, and domain circumscription [47] which provides a formalisation of the so called domain closure assumption; the assumption that the only individuals that a system must deal with are those explicitly named.

While default logic attempts to formalise particular assumptions, and circumscription the basis for assuming that something is not true, a third approach was proposed that attempted to formalise nonmonotonic reasoning using the notion of what is known. Following the initial attempts by McDermott and Doyle [91, 90], Moore proposed his autoepistemic logic [94] which used ideas from modal logic to authorise conclusions that are either necessarily true, or not necessarily untrue.

Moore claims that autoepistemic reasoning is the kind of reasoning intuitively employed in many situations, giving the example of how he determines whether or not he has an elder brother. He argues that he does not know that he has no elder brother because he has been explicitly told that no such person exists, or by carefully sifting all the available evidence, but simply because if he had an elder brother he would know about him. In its original form autoepistemic logic is a purely propositional modal logic, with no means of incorporating quantifiers or individual variables. Konolige [73, 72] extends Moore's approach to a first order system initially [73] to a a system that does not allow "quantifying in" to the scope of a modality and later [72] to a full first order system.

4.2 Imprecise information

The pioneering work in combining fuzzy sets and logic to allow the representation of and inference with vague information was performed by Zadeh [165, 166]. He described a system which generalised both two-valued and multi-valued logic by allowing all predicates, and the relations between predicates, to be descibed by fuzzy sets. Thus within fuzzy logic it is possible to provide a mathematical description of the statement: "If Hans has a new red Porsche then it is likely that his wife is young" which Zadeh claims is able to take account of the natural fuzziness of the terms "new", "likely" and "young". Representation is handled by defining fuzzy sets for these fuzzy terms, and inference by applying methods for infering the fuzzy term that is implied by what is known. A number of methods are proposed, including one that involves solving a non-linear program, but the most widely used is a generalisation of the classical inference pattern of modus ponens. This allows the inference of one fuzzy predicate from another and a fuzzy implication that relates them.

The idea behind Zadeh's proposal is extremely appealing, and many people have been moved to build upon his work. There are many applications of fuzzy logic (see for example those collected in [86]), especially in the domain of control where a vast number of sucesses have been reported. There has also been a lot of theoretical work, ranging from philosophical assaults on the basis of the theory [45, 61] and their rebuttals [42, 51], to detailed elaborations of the nature of the connectives it uses [146].

Building on the work on fuzzy logic, there have been several approaches to providing some form of fuzzy logic programming environment. Indeed, one could consider FRIL as such although as Baldwin and Zhou [10] point out when introducing it, the system is based upon the mathematics of relations rather than the predicate calculus. As discussed above, in FRIL one specifies a database of facts and rules using a fuzzy relational algebra, and then queries the database. In order to answer the query FRIL combines relations and the fuzzy degrees of membership of the members of the relations to compute which facts fit the query and to what degree. Thus FRIL is inherently fuzzy, but can also deal with point, interval and fuzzy probabilities [9].

A subsequent development [8], which has now been combined with the fuzzy relational inference mechanism described above, is support logic programming. In this system each clause is quantified with a *support pair*, that is a pair of numbers which represent the possible and necessary degree to which the statement is supported. Roughly speaking the possible support is the largest degree of support that might be provided for a clause and the necessary support is the smallest degree of support that is known to be provided. These degrees of support are related to possibility measures (and their dual necessity measures) and the measures introduced in Shafer's [141] theory of evidence.

Another approach to providing a fuzzy logic programming is FProlog [87], a fuzzy prolog interpreter. This builds the association of a degree of membership to clauses into the proof mechanism, so that each time a fuzzy fact is used to define a subgoal the degree of membership of the goal within the set of true facts is adjusted according to the rules of fuzzy logic. This means that the degree of membership is in fact a degree of truth. In the FProlog system backtracking may be triggered by partial failure when a truth value falls below a certain threshold, and the *not* operator is extended so that if the query X succeeds with truth value v, then $not(X)$ succeeds with degree $1 - v$.

4.3 Logical approaches to uncertain information

One obvious approach to handling deductive reasoning with uncertain information is to take a logic and attach some measure of validity to every piece of information with which one wants to reason. Just as in Section 3.3 this measure can be a probability, possibility, Dempster-Shafer belief, or indeed any other kind of measure that one desires to use, and each measure will dictate different rules for combining the measures when reasoning. A variation on this theme is the use of augmented logic programming systems, and these are clearly very relevant from the point of view of deductive databases since there is a close connection between logic programs and certain classes of deductive database. Thus we will cover both the use of augmented logics and logic programming languages in this section.

4.3.1 Possibilistic logic

While fuzzy logic may be used to represent non-Boolean properties which may be satisfied to some degree, possibility theory is a natural tool for representing the uncertainty in Boolean properties that is created by incomplete knowledge [42]. To handle this uncertainty in a logical framework, possibility theory has been extended by Dubois and Prade [41] to create a numerically quantified logic called possibilistic logic. In this formalism either a possibility measure or a necessity measure[4] is attached to every formula of classical first order logic. Extensions of classical inference patterns such as modus ponens and the resolution principle that include the propagation of the possibility measures are provided, and the approach provides a means of performing theorem proving under uncertainty [36].

Possibilistic logic has several advantages over probabilistic logic. In particular the measure attached to a formula is not necessarily reduced when the formula is combined with others, and the possibility and necessity bounds on the measure may be obtained with more precision than the probabilistic ones. Further extensions to the system may be found in [35, 39], including a resolution

[4] Necessity measures are the dual of posibility measures and quantify the degree to which propositions are known to be true.

principle that allows possibility weighted clauses to be combined with necessity weighted ones, and a scheme for including fuzzy predicates. Furthermore the completeness of resolution in possibilistic logic has been proven, and a scheme for representing default information in possibilistic logic has been proposed [12].

There has also been progress in providing a logic programming language in which uncertainty is quantified with possibility values. Poslog [34] is an automated theorem prover based on resolution refutation for first order clauses weighted with lower bounds on their possibility or necessity degree. The system is complete in that it has been proved that it finds optimal refutations—those refutations with maximal possibility or necessity degrees.

Extending this Dubois et al. [32] lay the groundwork for a logic programming environment based on Prolog in which clauses may be quantified with possibility measures. This allows the logical conclusions of a set of quantified clauses to be obtained along with their associated degrees of possibility. In a similar vein the same authors [33] discuss a possibilistic truth maintenance system which maintains the degree of possibility of every piece of information that it deals with, allowing reasoning that combines handling uncertainty and maintaining consistency. Indeed, the degrees of possibility are used in order to resolve conflicts between inconsistent pieces of information.

4.3.2 Probabilistic logics

The classic paper on reasoning combining logic and probability is due to Nilsson [107], though unbeknownst to him he was restating for an artificial intelligence audience work that was originally carried out by Smith [147] and de Finetti [48]. The paper considers the consequence of combining the probabilities assigned to P and $P \supset Q$ when the two are combined using modus ponens. In general the probabilities on a set of sentences do not completely determine the underlying joint distribution so that it is only possible to determine the bounds on derived sentences. Thus:

$$p(P) + p(P \supset Q) - 1 \leq p(Q) \leq p(P \supset Q)$$

The advantage of the approach is that it makes no assumptions about the joint distribution over all the sentences, the disadvantage is that only the bounds on probabilities may be computed. A second problem is that it is possible to argue that $p(P \supset Q)$ is not a good representation of the certainty of the rule *"if P then Q"*. In a more recent paper Nilsson [106] suggests that associating the conditional value $p(Q \mid P)$ gives more natural results which include:

$$p(Q \mid P)p(P) \leq p(Q) \leq 1$$

McLeish [92] extends Nilsson's method to cover cases in which $p(P)+p(P \supset Q) < 1$ since these give invalid results for reasonable probability values in Nilsson's scheme. This extension allows the representation of default information (assumptions that may be contradicted at a later stage). A further extension of Nilsson's method replaces the probabilities of P, $P \supset Q$ and Q with the relevant belief functions [141] allowing McLeish to explore notions of belief function entailment, a mechanism which differs from probabilistic entailment. It is worth noting that in doing so she uses both the open and closed world assumptions. She has also [93] considered how probabilistic logic may be used to handle default information by allowing inconsistent information and suggesting how probabilistic entailment may be carried out in such a situation.

Another attempt to combine logic and belief functions is Saffiotti's belief function logic [133, 134]. In this system every sentence of a first order logic language is quantified with a pair of numbers $[a, b]$ which respectively represent the degree to which the sentence is known to be true, and may be true. The quantification makes the logic a generalisation of standard first order logic that boasts a well defined semantics and notion of deduction in which the sentences that may be derived are determined by the rules of the underlying logic and the degrees to which they are believed are determined using the rules of belief functions. The logic also handles reasoning by cases and by allowing belief in the contradiction[5] provides a means of coping with partial inconsistency in a similar way to that proposed for possibilistic logic.

Further considerations are provided by Bacchus [7] who examines the semantics of probabilistic logic in some detail. He argues that a reasonably expressive language will need to be first order, and that a first order language will need a more sophisticated means of establishing the probabilities of its constituent sentences than generating a distribution over possible worlds. He then proposes a solution and discusses its use in the representation of defaults and statistical knowledge of the form "90% of birds fly". Halpern [62] takes this solution, and arguing that it is often useful to combine reasoning using statistical information and information about beliefs, which is essentially a distribution over possible worlds, shows that it is possible to find a common framework for both kinds of knowledge. The coupling of probability with logic has also been considered by Heinsohn [64], although his work is more concerned with drawing conclusions about members of terminological heirarchies than general first order techniques, and so is in some ways closer to Keißling et al. [71] than Bacchus and Halpern.

Another interesting advance in combining probabilistic methods with deductive techniques is made by Güntzer et al. [60]. They deal with rule-based reasoning where every rule in the knowledge base has an associated conditional probability, and provide a set of inference rules for deducing connections between various events that are implicit in the knowledge base. These inference rules enable the bounds on the conditional probabilities of these new connections to be deduced, and the soundness of these bounds is proved. Clearly the new connections and their associated probabilties allow new facts to be deduced along with a probabilistic quantifier. It seems at first as if this formalism is a new probabilistic logic.

However, the kind of links over which the probabilities are propagated are not logical implications, but a form of "causal" relation such as that handled by Pearl [113]. Thus the formalism provides an alternative notation for probabilistic causal networks, providing a means of performing correct probabilistic calculations without explicitly building the causal network. Like causal net-

[5] As suggested by Smets [145]

works, the representation is inherently propositional, and the proof procedure for any one query may not be used to answer another query. In [153] the same authors demonstrate that their approach can handle multiply connected networks to give the same results as Lauritzen and Spiegelhalter [78], while arguing that it is really just a more accurate version of Quinlan's INFERNO [129].

This work bears many similarities to that of Amarger et al. [4]. Although they tie their work in with the use of default rules, what Amarger et al. are essentially doing is to take a series of propositions, whose relationships are stated using imprecisely known probabilities, and inferring new probabilistic relations between them. Thus given the information that between 70 and 90% of students do sport, and between 85 and 90% of students are young, they show how to conclude what proportion of young people do sport. The most interesting point about the method that they propose, which is based upon local rules for calculating the tightest possible bounds just as Güntzer et al.'s is, is that they require no prior probabilities, and that the existence of loops in the graph of dependencies between propositions serves to improve the bounds.

4.4 Numerical approaches to uncertain information

The other main approach to handling uncertainty is to deal with purely numerical methods, concentrating on combining pieces of evidence for some conclusion, performing inference in ways that are derived from classical statistics. Thus there is no idea of deductive reasoning—rather there is an adjustment of some underlying distribution which makes particular options the correct conclusions. For a number of years the use of methods such as these were scorned by the artificial intelligence community, but due to pioneering work on a novel form of representation known as *Bayesian belief networks* [113], purely numerical methods have become well accepted.

4.4.1 Reasoning with belief networks

The major reason that probabilistic methods were initially shunned by the artificial intelligence community was because they were thought to be impractical. If we have a system that involves n variables we need 2^n probabilities to fully specify the probabilistic relationships between those variables. For expert systems in which n is reasonably large this suggests that vast numbers of probabilities need to be elicited and then updated during inference.

What Pearl [113][6] realised that although this is true in general, in practice one often needs many fewer probabilities. This is because the kind of knowledge that is represented in artificial intelligence systems does not usually involve inter-relationships between many variables. The relationships that do exist, and thus the probabilities that are required, may be exposed by the construction of a network in which variables are represented by nodes and the explicit dependencies between them by arcs. When two nodes are not connected it is because

[6] Pearl's original proposal was made in a series of papers in journals and conferences. The book [113] contains this material edited together and has become the benchmark reference for belief networks.

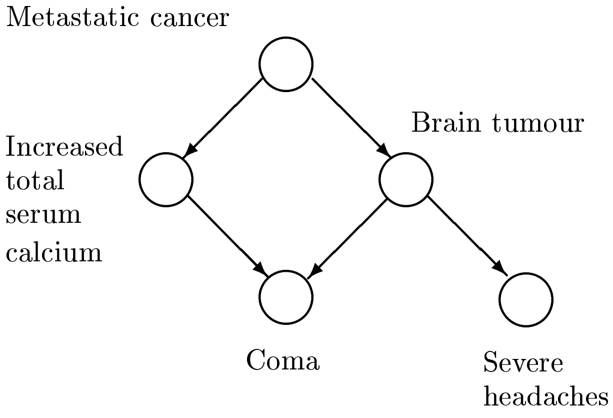

Figure 1: Some medical knowledge

the value of one node is known to be conditionally independent of that of the other. The resulting structure not only identifies the necessary probabilities but may be used as the basis for computing the updated values during inference, and a scheme for doing so is provided in [113].

As an example, consider the network in Figure 1 which represents medical knowledge about a set of related conditions. From the network we can tell that both the occurence of increased calcium and the occurence of brain tumours are dependent upon the occurrrence of metastatic cancer, while the occurence of severe headaches is dependent upon the occurence of a brain tumour and the occurence of coma is dependent jointly upon the occurence of increased calcium and a brain tumour. Thus when elicting the probabilities that concern coma we need not bother with metastatic cancer or severe headaches, reducing the necessary number of probabilities from 2^5 to 2^3. Similarly, the graph tells us that the probabilities of increased calcium and brain tumour are conditionally independent of one another once the existence of metastatic cancer and coma are established.

Pearl's method for updating values works for a large class of networks—all those in which there is at most one route between any pair of nodes, but it fails for networks such as those in Figure 1 in which loops occur. However, there are methods for handling such networks, the most celebrated of which is that of Lauritzen and Spiegelhalter [78]. It is also worth noting that, despite the fact that their name suggests that these networks are limited to using subjective probabilities, there is nothing to prevent them being used with experimentally determined objective probabilities as is done in the QUALQUANT [152] system. Indeed, as Neapolitan [100] has pointed out, most of the names for these networks are misleading, and they should perhaps be renamed as *independence networks* since what they encode is explicit conditional independencies between variables.

Since they were originally proposed, the use of belief networks has become widespread. There are numerous applications that make use of them, for example [2, 6, 29, 99, 108]. They have even been proposed as a means of establishing the best document to retrieve from a document database [155], an information retrieval application that clearly has connections with the needs of database users.

There are also a number of implementations of the various systems for performing inference with belief networks. For example, the commerical system Hugin [5] implements the Lauritzen and Spiegelhalter algorithm, while the IDEAL system [149] implements a wide range of different algorithms many of which are specialised for particular types of network. Similar schemes have been proposed for other methods of handling uncertainty. In particular, Shenoy and Shafer [142] identified a method for propagating belief functions [141] based on a underlying network representation that could be generalised to propagate probabilities [144] and possibility values [38, 143]. This work has been implemented as the Pulcinella system [136, 138].

4.4.2 Dynamic construction of belief networks

Now, there are a number of problems with applying belief networks in the kind of dynamic environment that exists within a database. The first is that to perform inference with a belief network, one needs a network, and in a database environment the usual method of obtaining one, which is to talk to a domain expert, is clearly not applicable. Instead, one has to provide a means of constructing the network automatically. In recent years there has been a considerable amount of work on this question [157], and in this section we discuss a number of relevant proposals.

One of the earliest attempts to provide for automated construction was that of Srinivas et al. [150] who take a number of different types of qualitative information, such as "A is a cause of B" or "A is independent of B given C", and use these, along with a black box that tests for independence, to create networks. Clearly this approach finesses one of the harder problems in ignoring the test for independence, but it is nevertheless obvious that the algorithm that they provide could be usefully linked with an automated reasoning system to build networks from logical statements. In addition, the system is implemented as part of the IDEAL [149] package.

In contrast with this "expert centered" approach, Cooper and Herskovits [26] have developed an algorithm that can deduce the most likely structure of a belief network linking a set of variables given a database of cases of the form *"in Case 1, A is present, B is absent, and C is present"*. The derivation of the network is based upon the assumptions that the database explicitly mentions all the relevant variables, the cases mentioned in the database are independent, and all the cases are complete. This algorithm has been tested on a moderately sized database of $10,000$ cases generated by an existing belief network. The algorithm took around 15 minutes to generate a network that was nearly identical to the original. This compares favourably with their initial experiements with a method based on finding the maximum entropy distribution [65] for a network based on a set of cases which took nearly 24 hours to handle the $10,000$ case database. The same authors [27] have also considered the problem of assessing the conditional probability values necessary to perform probabilistic inference in the belief network.

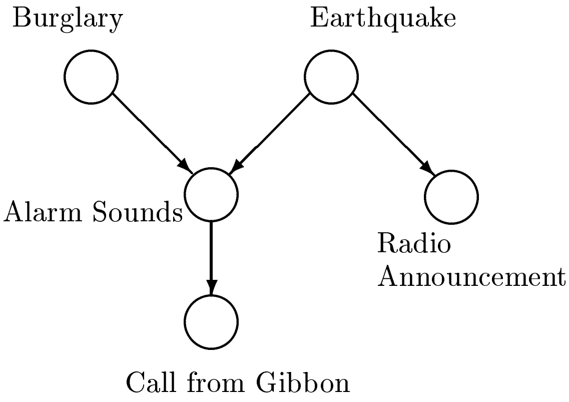

Figure 2: The events surrounding Mr Holmes

Wen [159] takes a slightly different approach, starting from a database which records statistical data of the form *"D and E occur with A, B and C on 2048 occasions"*. Wen discusses how to reduce sets of relations into fourth normal form, which correspond to the cliques of the equivalent belief network, and from which the necessary conditional probabilities may be learnt. He also discusses methods, based on the maximum entropy principle [70] for completing sets of conditional probabilities.

The other major disadvantage with network based formalisms is the fact that they are inherently propositional. Consider Pearl's [113] seminal example about Mr Holmes and his burglar alarm. Either an earthquake or a burglary would cause the alarm to sound, an earthquake would most likely result in a radio announcement, and the sounding of the alarm would cause Holmes' neighbour Mrs Gibbon to telephone him at work. This may be represented by a Bayesian network (see Figure 2) and the result used to predict the most likely cause of the alarm sounding given that Mrs Gibbon calls.

The problem with this model is how to extend it to cover the case, for instance, in which Dr. Watson, another neighbour who is more reliable than Mrs Gibbon, also telephones, and the case when Inspector Lestrade, who happens to be passing, telephones to report a suspicious person hanging around the Holmes residence, or even the case when Watson rather than Gibbon is the only one to call. The model as it stands does not allow universally quantified statements such as *"a ringing alarm would cause a neighbour to telephone"*, restricting expressiveness to statements to such as *"the alarm would cause Mrs Gibbon to telephone"*.

Saffiotti and Umkehrer [139] address the removal of this restriction, presenting a method for dynamically constructing networks suitable for their tool Pulcinella. Facts and relations are represented in first order logic, and resolution used to build a proof path from which a network can be constructed. The network may then be fed to Pulcinella for evaluation. The implementation described is proven sound and complete for the use of evidence theory [141] as a means of quantifying the uncertainty in the facts and relations, though it is possible to extend the approach to other uncertainty handling formalisms and representation languages.

A similar approach is adopted by Poole [118, 117] who uses first order horn clause logic to represent the variables in a belief network and the relationships between them, attaching an associated probability to each. The logical clauses are then used to deduce various facts such that the probability associated with the facts is the probability that would be established for them using the equivalent network. Thus the network is never built, but is implicit in the computation, and this differentiates the approach from the earlier work presented by Horsch and Poole [66] in which horn clauses were used to provide general statements which were then instantantiated at runtime and used to explicitly construct Bayesian networks.

Another related method is that of Goldman and Charniak [57], who are also interested in explicitly building Bayesian networks in dynamic environments. Their approach differs in that they use a specialised network-construction language rather than first order logic, and, being motivated by understanding natural language, is not goal directed in the same way as the methods listed above. Given that deductive database applications are likely to be somewhat goal directed, it seems that this approach is not necessarily the best, but its close relation to work on integrating probability values and truth-maintenance systems [77, 128] should be noted.

Given the dynamic nature of databases, it is important to remember that all of the network construction techniques mentioned so far build networks that are correct at a particular instant in time, but do not allow for changes in the network. Provan [127] addresses this problem, using a sensitivity analysis to determine when better decisions would have been taken using a different model, and gives an algorithm for performing the updating. The need to update networks is often due to the fact that the problem being diagnosed changes over time, and so the history of the problem becomes important. This time dependency is handled in Provan's system Dynasty, which also allows different levels of granularity of problem description to be considered during the diganosis. Similar issues are addressed by Dagum et al. [29] who synthesize belief networks with time-series analysis to create dynamic network models for use in forecasting.

Another factor that has been disregarded in all the systems considered so far is the problem of seperating model construction from evaluation. In a resource bound environment this could lead to the query-driven construction of a network that could not be evaluated in reasonable time. Goldman and Breese [56] consider how to alleviate this difficulty by integrating the two stages to give an "anytime algorithm" for query evaluation which gives successively better approximations of the answer. In addition to always providing a solution, the method allows useless solutions to be identified at an early stage, and its deductive style brings the use of numerical methods almost full circle and back to the logical methods discussed above.

4.4.3 Belief networks and relational databases

It should also be noted that there are some striking similarities between the representation of data in probabilistic networks and in relational databases. Pittarelli [114] pointed out that his probabilistic databases allow the computation of the same probability distributions as belief networks given information about the dependencies between different data. In other words, if the graphical structure is known, then the probabilistic information that is stored in his system is sufficient to establish a unique joint probability distribution for all the pieces of data in the database.

Similar findings were reported by Studeny [151] in his attempt to characterise the nature of conditionally independent pieces of information. Despite coming from a completely different direction, Studeny spotted a close analogy between his definition of conditional independence and the idea of embedded multi-valued dependencies which are a means of describing relational databases. However, he also showed that there were some differences between the ideas.

Finally, in a recent paper, Wong et al. [163] have shown that bayesian networks can be represented as relational databases. Rather as one might expect given the work discussed above by Poole [118, 117], and the close correspondance between predicates and the tuples in a relational table, it seems that if a probability distribution is given over a set of relational tables, it is possible to perform correct probabilistic inference using just the project and join operations that one would expect of a relational database. Thus when new evidence is obtained its effects may be propagated through the database in a manner consistent with the underlying dependencies but without building a network. However, the method does rely upon the prior structuring of the relations in order to represent the conditional independencies.

4.5 Argumentation

An approach that is somewhat related to the construction of belief networks is that of argumentation, which is discussed in detail by Krause et al. [75]. This approach also has important differences from most other methods for handling uncertainty. The basic idea behind argumentation is that it should be possible to say more about the certainty of a particular fact than may be expressed by quantifying it with a number between 0 and 1. In particular, it should be possible to assess the reason why a fact is thought to hold, and use this *argument* for the fact as its quantification.

An argument is thus rather like an endorsement [24], though it is more accurate to think of it as a tentative proof—the proof is tentative because argumentation allows the proof of a fact being true to co-exist with the proof of it being false, a state of affairs that is becoming acceptable amongst logicians [53]. The advantage of constructing arguments for facts, and using these arguments as quantifiers of the certainty of the facts is that it is possible to reason about the arguments themselves. This reasoning about arguments can be used to determine which facts are most acceptable, and this in turn firmly sets argumentation apart from the theory of endorsements.

Reasoning about arguments takes a number of different forms. Firstly, it is possible to use the logic of argumentation LA to combine different arguments together. Thus it is possible to combine an argument for a proposition A with one for B to get an argument for A *and* B, or to establish an argument for B from one for A and one for A *implies* B. Secondly it is possible to aggregate arguments together, so that a number of arguments for A can be combined to get a single argument with a suitable strength, for instance by applying an improper linear model [30] or by counting the number of steps in the argument. The result of the combination can then be used to rate A against competeing hypotheses.

Finally, and most interestingly, the structure of arguments can be analysed. In this process, an argument is classified into one of six classes based upon its "acceptability", which is determined by examining the arguments for and against given proposition to see whether any of the steps in the arguments themselves have any arguments against them. The advantage of this approach is that the degree of confidence of the proposition is determined by the structure of the reasoning rather than being imposed by the assignment of a numerical measure.

A number of other authors have proposed systems of argumentation, often as a way of formalising default reasoning, which have strong relations to the system described above.

Loui [85] describes just such a system. He incorporates rules that are explicitly denoted as being defeatable, and discusses ways of constructing arguments from such rules, along with criteria for resolving conflicts between arguments based on preferences between premises and the amount of evidence used by the arguments. These meta-level reasons for preferring one argument over another are interesting because they combine rules of thumb such as the shortest path heuristic equally with better founded notions of arguments attacking each other's premises.

A method that is more closely related to classical first order logic is that of Poole [119] who builds his system on top of earlier work on the Theorist [121] system which carries out default reasoning and diagnosis by identifying hypotheses which are consistent with what is known. This method of proceeding has been shown [120] to be equivalent to default logic. Poole considers what should be predicted when classically inconsistent information, such as the fact that both A and $\neg A$ are true, is deduced, and considers a number of different ways of interpreting the contradiction. Thus it is possible to say that the contradiction indicates that either A or $\neg A$ may be true, or that neither is definitely true. It is also possible to argue that even though there is a contradiction, some things may be still be predictable, for instance because B follows from A and C follows from $\neg A$ while D is true if B or C is true. These considerations produce a number of criteria for selecting particular predictions based on how much one is prepared to deduce from the contradictions, and these criteria are clearly close to the acceptability classes mentioned above.

In addition, Lin and Shoham [83] present a system which is also very similar to that of Krause at al., defining arguments as chains of inferences, and showing that their framework can capture more specific forms of nonmonotonic inference such as defualt logic, autoepistemic logic, and circumscription, as well as the kind of hypothesis formation found in Theorist. The use of argumentation as an integrating framework is also the theme taken up by Dung [44] who shows that his system can capture default logic and logic programming. Finally, Pollack [116] has introduced a system for suppositional reasoning, that is hypothesising something and then seeing if it can be justified from what is known. His system deals with the interactions between arguments, delving into the structure to identify which arguments are good and which are rebutted by others.

4.6 Summary

As with the handling of imperfect information in databases, there are four basic ways in which we can represent and reason with imperfect data. We can: (1) use a number or symbol to give the strength with which an attribute is known to take a given value, (2) use a number or symbol to indicate the strength of the relation between attribute values, (3) use a number or symbol to indicate the strength of inheritance between two objects, or (4) describe how to derive the number or symbol appropriate to a given attribute value from that known to be associated with some other attribute value. Thus, a probabilsitic logic provides a mechanism for implementing three out of the four ways—the first by associating probabilities with propositions, the second by associating probabilities with implications, and the last by specifying how to determine the probability of the consequent of an implication from the probabilities of its antecedent and the implication itself. Similarly, the use of a nonmonotonic logic provides a means of specifying (as true or false) the strength with which propositions and expressions relating propositions are known to hold, and given a set of formulae provides a means of determining whether or not various conclusions are true or false.

5 Discussion

Now, the distinction between the database view of uncertainty and the artifical intelligence view of uncertainty, and the use of the classification of different types of imperfection are not the only ways of obtaining some kind of perpsective on handling imperfect information. Other perspectives may be useful, and some of these are discussed in this section, along with some of the common themes from the different approaches. This section also mentions some other suggestions that have been made both in the database and artificial intelligence communities and which do not fit into the previous sections. Finally, a general approach to handling imperfect information is suggested.

5.1 Different perspectives

The survey carried out above was structured along two axes. Firstly a distinction was made between work carried out in the database community and work carried out in the artificial intelligence community in order to stress the contribution of both camps to the problem of representing and reasoning with imperfect information. The two bodies of work were then classified according to the kind of imperfection that was dealt with, splitting the contributions according to a heirachy of different types of imperfection that was synthesized from several that have been proposed in the literature.

The advantage of this approach is that, in general, it allows a clear distinction to be made between the different approaches since they are usually aimed at solving one particular problem caused by a particular form of imperfection. Thus null values [23, 81, 169] are intended to solve the problem of incomplete information, and probabilistic networks [113] are intended to handle information that is uncertain. Where it is possible to make this distinction, it is clear which methods are applicable to which problems, and thus which should be employed in given situations.

However, there are some approaches that do not easily fall into one category. For instance, consider the scheme proposed by Barbara et al. [11]. Although classified above as a scheme for handling uncertain information because of its basis in probability theory, the method is intended to model cases where the uncertainty is complicated by ignorance about the probabilties of certain attribute values. Thus it could be argued that the scheme should be classified as a system for handling ignorance. Given this kind of difficulty in making a clean classification, it is worth considering other perspectives that may be informative.

One obvious distinction that can be made is between approaches from the database side is between those intended for relational databases and those intended for object-oriented databases. Given the respective ages of the paradigms it is not surprising that the majority of the approaches are aimed at the relational model. However, those that use the object-oriented model raise some interesting points. For instance, as George et al. [54] point out, moving from to an object-oriented model involves more than just considering how to represent the imperfection in the values of the attributes in objects—it also involves considering imperfection in the inheritance hierachy, and that opens up a whole area of potential problems [52]. Thinking about imperfection in inheritance also highlights the fact that work on the relational model has failed to cover problems such as incompleteness of relations or the fact that data might not fit into the relations [170].

Another possible distinction is one that is based on the type of solution that is suggested for handling imperfection, rather than the kind of problem (whether it is handling incompleteness in a relational database or imprecision in an object-oriented database) that the solution is proposed for. Thus one could distinguish between fuzzy approaches and probabilistic approaches regardless of whether they were proposed to deal with imprecision or uncertainty or whether they were proposed in the context of work on databases, logic programming, or as part of some intelligent system.

5.2 Common themes

As well as suggesting these alternative perspectives, it is also possible to pick up on some of the themes that are common to a number of approaches. Perhaps the most obvious is the fact that some of the approaches are simply re-iterations of previous work. Thus, to take a prominent and well known example, Nilsson's probabilistic logic re-established results originally due to Smith [147] and de Finetti [48]. In many ways this is not surprising since the original results were buried deep in the statistics literature, and the idea in question was an inherently sensible one. However, it is notable that the study of imperfect information in databases throws up as many instances of repetition as it does.

This is especially noticible in the wide use of the concept of the 'possible' and 'necessary' values of attributes. The essential notion that when incomplete information is used to answer queries, it is possible to distinguish between those attributes that may satisfy the query and are thus possible and those that must satisfy the query and are thus necessary is invoked by a whole host of authors. The interpretations range from Lipski [84] with his totally symbolic approach, through the similar work of Williams et al. [161] and Morrissey [95] to the wholly numerical work of Prade [122].

This then points to another common theme, and one that was mentioned in passing in the previous section. This is the rather limited view of imperfection that almost every author has considered. Throughout the work on the subject there is an almost universal concentration on the problem of how to deal with imperfections in value, due to incompleteness, imprecision and uncertainty and an almost universal ignorance (in the sense of "a decision to ignore") of deeper issues such data that does not fit into tables, or data that has incompletely specified schemas. As has been remarked in the context of decision making [49], there is more to handling imperfect data than just manipulating numbers, and it is to be hoped that in the future some research will be directed at some of these other areas.

5.3 Other ideas

The discussion so far has been restricted to the work surveyed in the previous sections, but it is worth making some remarks on broader issues. The first extends the point about dealing with issues other than the value of attributes to dealing with issues other than those directly to do with the information system whose imperfect information is being dealt with. To echo the quote from Motro with which this paper opened, the reason for studying imperfect information is to build systems that deal with the real world. Thus it would be both interesting and useful to study the imperfections in data that real systems come up against from the perspective of actually building such a real system rather than studying the problems of imperfect data in a theoretical vacuum.

The second, somewhat related, point is that it seems worth considering how the imperfect data will be used. Just as Al-Zobaidie and Grimson [3] discuss how databases of perfect information and expert systems should be coupled together, there is plenty of room for discussion on the ways in which databases and knowledge-based systems can be used together when information is not perfect. For instance, as George et al. [54] point out, there is a choice as to whether to deal with uncertainty at the level of the database or at the level of the knowledge-based system, a question that to some degree has been addressed by

Saffiotti's work on separating knowledge and uncertainty representations [135].

The third and final idea is that the different approaches to representing imperfect information that have been discussed above should not be considered as mutually exclusive. In the last few years as number of researchers working in the area of representing uncertainty [50, 109, 136] have made the suggestion that the many different approaches are complementary. Thus the approaches should be used in combination [110], or at least should all be considered to see which best fits the problem [137, 138]. Thus several different approaches may be used in a single database either by having different parts of the database quantified using different methods, or by combining different methods as Zadeh [164] suggests. In this vein Umano [156] proposes the combined use of possibilistic and fuzzy models, with possibility distributions being used to model the uncertainty in the value of an attribute, and fuzzy degrees of membership being used to model the degree of association between values.

6 Conclusion

To complete this trawl through the literature a few words are in order. Firstly, it is clear that a large amount of work has been done on the problem of handling imperfection in information systems. This work has come from both the database community and the artificial intelligence community, and both camps can benefit from the successes and failures of the other, though at the moment they don't always seem to be aware that the other exists. Hopefully this paper will contribute to an increased awareness. Secondly, is also clear that we are still a long way from having a unified theory of imperfection, and that a lot of work remains to be done if such a theory is to be obtained. It can be argued that such a theory might not be necessary, since a significant amount can be achieved by the individual approaches that have been developed, but I believe it is desirable and I hope that the plea for an eclectic approach is heeded, at least in some quarters. Finally, and relatedly, I hope that the elements in common between the different approaches to handling imperfect information, as well as the areas that have not yet been addressed, are illuminated by this paper and that it thus contributes towards the eventual development of a general framework for handling imperfection.

Acknowledgements

This research was partially supported by Esprit Project 6333 IDEA (Intelligent Database Environments for Advanced Applications) and Esprit Basic Research Action 6156 DRUMS (Defeasible Reasoning and Uncertainty Management Systems). Many thanks to Tony Hunter for providing me with useful criticism and copies of many of the papers from the second UMIS meeting which I would otherwise have not had access to, and to Henri Prade for providing me with a veritable mountain of material which saved me a lot of time and the ICRF library a lot of money. Finally, I am grateful to John Fox, Paul Krause, and two anonymous referees for their comments on early versions of the paper.

References

[1] Abiteboul, S., Kanellakis, P., and Grahne, C. (1987) On the representation and querying of possible worlds, *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, ACM Press, 34–48.

[2] Abramson, B. (1991) ARCO1: An application of belief networks to the oil market, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, 1–8.

[3] Al-Zobaidie, A. and Grimson, J. B. (1987) Expert systems and database systems: how can they serve each other?, *Expert Systems*, **4**, 30–37.

[4] Amarger, S., Dubois, D. and Prade, H. (1991) Constraint propagation with imprecise conditional probabilities, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, 26–34.

[5] Andersen S. K., Olesen, K. G., Jensen, F. V. and Jensen, F. (1989) HUGIN—a shell for building belief universes for expert systems, *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Sydney, 1080–1085.

[6] Andreassen, S., Woldbye, M., Falck, B., and Andersen S. K. (1987) MUNIN—a causal probabilistic network for interpretation of electromyographic findings, *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, Milan, 366–372.

[7] Bacchus, F. (1990) On probability distributions over possible worlds, in *Uncertainty in Artificial Intelligence 4*, R. D. Shacter, T. S. Levitt, L. N. Kanal and J. F. Lemmer, eds., Elsevier Science Publishers.

[8] Baldwin, J. F. (1987) Evidential support logic programming, *Fuzzy Sets and Systems*, **24** 1–26.

[9] Baldwin, J. F. (1986) Automated fuzzy and probabilistic inference, *Fuzzy Sets and Systems*, **18** 219–235.

[10] Baldwin, J. F. and Zhou, S. Q. (1984) A fuzzy relational inference language, *Fuzzy Sets and Systems* **14** 155–174.

[11] Barbara, D., Garcia-Molina, H., and Porter D. (1990) A probabilistic relational data model, *Proceedings of the 1990 EDBT Conference*, 60–74.

[12] Beneferhat, S., Dubois, D., and Prade, H. (1992) Representing default rules in possibilistic logic, *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, Cambridge, MA, 673–684.

[13] Besnard, P. (1989) *Introduction to default logic*, Springer-Verlag, Berlin.

[14] Brewka, G. (1991) *Nonmonotonic reasoning: logical foundations of commonsense*, Cambridge Trends in Computer Science 12, Cambridge University Press.

[15] Bosc, P. and Prade, H. (1993) An introduction to fuzzy set and possibility theory based approaches to the treatment of uncertainty and imprecision in database management systems, *Proceedings of the 2nd Workshop on Uncertainty Management in Information Systems: From Needs to Solutions*, Catalina, CA.

[16] Bonissone, P. P. and Tong, R. M. (1985) Editorial: reasoning with uncertainty in expert systems, *International Journal of Man Machine Studies*, **22**, 241–250.

[17] Buckles, B. P. and Petry, F. E. (1987) Generalised database and information systems, in *Analysis of fuzzy information*, Volume 2, J. C. Bezdek, ed., CRC Press, 177–201.

[18] Buckles, B. P. and Petry, F. E. (1983) Extension of the fuzzy database with fuzzy arithmetic *Proceedings of the IFAC Symposium*, Marseille, 421–426.

[19] Buckles, B. P. and Petry, F. E. (1982) A fuzzy representation of data for relational databases, *Fuzzy Sets and Systems*, **5**, 213–226.

[20] Cavallo, R. and Pittarelli, M. (1987) The theory of probabilistic databases, *Proceedings of the 13th Very Large Database Conference*, Brighton, 71–81.

[21] Clark, K. L. (1978) Negation as failure, in *Logic and databases*, H. Gallaire and J. Minker eds., Plenum Press, New York.

[22] Clarke, M. R. B., Froidevaux, C., Grégoire, E. and Smets, P. (1991) *Journal of Applied Non-Classical Logics*, Special Issue on Uncertainty, Conditionals and Nonmonotonicity.

[23] Codd, E. F. (1979) Extending the database relational model to capture more meaning, *ACM Transactions on Database Systems*, **4**, 397–434.

[24] Cohen, P. R. (1985) *Heuristic reasoning about uncertainty: an artificial intelligence approach*, Pitman, London.

[25] Cooper, G. F. (1990) The computational complexity of probabiistic inference using belief networks, *Artificial Intelligence*, **42**, 393–405.

[26] Cooper, G. F. and Herskovits, E. (1991) A bayesian method for constructing bayesian belief networks from databases, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, 86–94.

[27] Cooper, G. F. and Herskovits, E. (1991) A bayesian method for the induction of probabilistic networks from data, Report SMI-1-93, Section of Medical Informatics, University of Pittsburgh.

[28] Dagum, P., Galper, A., (1993) Forecasting sleep apnea with dynamic netwrok models, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, Washington, 64–71.

[29] Dagum, P., Galper, A., and Horvitz, E. (1992) Dynamic network models for forecasting, *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Stanford, 41–48.

[30] Dawes, R. M. (1979) The robust beauty of improper linear models in decision making, *American Psychologist*, **34**, 571–582.

[31] Demolombe, R. and Fariñas del Cerro, L. (1988) An algebraic evaluation method for deduction in incomplete databases, *Journal of Logic Programming*, **5**, 183–205.

[32] Dubois, D, Lang, J and Prade, H. (1991) Towards possibilistic logic programming *Proceedings of the 8th International Conference on Logic Programming*, Paris, 581–595.

[33] Dubois, D., Lang, J and Prade, H. (1990) Handling uncertain knowledge in an ATMS using possibilistic logic *Proceedings of the TMS Workshop at the European Conference on Artificial Intelligence*, Stockholm, xxx–xxx.

[34] Dubois, D., Lang, J and Prade, H. (1990) Poslog: an inference system based upon possibilistic logic, *Proceedings of the North American Fuzzy Information Processing Society Congress*, Toronto, 177–180.

[35] Dubois, D., Lang, J and Prade, H. (1989) Automated reasoning using possibilistic logic: semantics, belief revision and variable certainty weights, *Proceedings of the 5th Workshop on Uncertainty in Artificial Intelligence*, Windsor, 81–87.

[36] Dubois, D., Lang, J and Prade, H. (1987) Theorem proving under uncertainty—a possibility theory based approach, *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, Milan, 984–986.

[37] Dubois D. and Prade, H. (1991) Certainty and uncertainty of (vague) knowledge and generalised dependencies in fuzzy databases, *Proceedings of the International Fuzzy Engineering Symposium '91*, Yokahoma, Japan, 239–249.

[38] Dubois, D. and Prade, H. (1990) Inference in possibilistic hypergraphs, in *Uncertainty in Knowledge Bases*, B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh eds., Springer-Verlag, Berlin, 250–259.

[39] Dubois, D. and Prade, H. (1990) Resolution principles in possibilistic logic, *International Journal of Approximate Reasoning*, **4**, 1–21.

[40] Dubois, D. and Prade, H. (1988) *Possibility theory: an Approach to the Computerized Processing of Uncertainty*, Plenum Press, New York.

[41] Dubois, D. and Prade, H. (1987) Necessity measures and the resolution principle, *IEEE Transactions on Systems, Man, and Cybernetics*, **17**, 474–478.

[42] Dubois, D., Prade, H., and Smets, Ph. (1994) Partial truth is not uncertainty: fuzzy logic vs. possibilistic logic, *IEEE Expert*, August 1994, 15–19.

[43] Dubois, D., Prade, H., and Testemale, C. (1988) Weighted fuzzy pattern matching, *Fuzzy Sets and Systems*, **28**, 313–331.

[44] Dung, P. M (1993) On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambéry, France, 852–857.

[45] Elkan, C. (1993) The paradoxical success of fuzzy logic, *Proceedings of the 11th National Conference on Artificial Intelligence*, Washington, D. C., 698–703.

[46] van Emden, M. H. (1986) Quantitative deduction and its fixpoint theory, *Journal of Logic Programming*, **3**, 37–53.

[47] Etherington, D. *Reasoning with incomplete information*, Pitman, London, 1988.

[48] de Finetti, B. (1974) *Theory of Probability*, Wiley, New York.

[49] Fox, J. (1991) Decision theory and autonomous systems, in *Decision Support Systems and Qualitative Reasoning*, M. G. Singh and L. Travé-Massuyés eds., North-Holland.

[50] Fox, J. (1986) Three arguments for extending the framework of probability, in *Uncertainty in Artificial Intelligence*, L. N. Kanal, J. F. Lemmer eds., North-Holland, Amsterdam, 447–458.

[51] Fox, J. (1981) Towards a reconciliation of fuzzy logic and standard logic, *International Journal of Man-Machine Studies*, **15**, 213–220.

[52] Froidevaux, C. and Kayser, D. (1988) Inheritance in semantic networks and default logic, in *Non-Standard Logics for Automated Reasoning*, P. Smets, E. H. Mamdani, D. Dubois and H. Prade eds, Academic Press, London.

[53] Gabbay, D. and Hunter, A (1991) Making inconsistency respectable: a logical framework for inconsistent reasoning, in *Fundamentals of Artificial Intelligence Research*, Lecture Notes in Artificial Intelligence 535, Springer-Verlag, Berlin, 19–32.

[54] George, R., Buckles, B. P. and Petry, F. E. (1991) An object-oriented data model to represent uncertainty in coupled artificial intelligence-database systems, in *The next generation of information systems: from data to knowledge*, Lecture Notes in Artificial Intelligence 611, Springer-Verlag, Berlin, 37–48.

[55] Ginsberg, M. L. (1987) *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, San Mateo, CA.

[56] Goldman, R. P. and Breese, J. S. (1992) Integrating model construction and evaluation, *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Stanford, 104–111.

[57] Goldman, R. P. and Charniak, E. (1990) Dynamic construction of belief networks, *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, Cambridge, MA, 90–97.

[58] Guardalben, G. and Lucarella, D. (1993) Information retrieval based on fuzzy reasoning, *Data and Knowledge Engineering*, **10**, 29–44.

[59] Gunter, E. and Libkin, L. (1994) OR-SML: a functional database programming language for disjunctive information and its applications, *Proceedings of the 5th International Conference on Database and Expert System Applications*, Athens, 641–650.

[60] Güntzer, U., Kießling, W. and Thöne, H. (1991) New directions for uncertainty reasoning in deductive databases, *Proceedings ACM SIGMOD International Conference on the Management of Data*, Denver, 178–187.

[61] Haack, S. (1979) Do we need fuzzy logic?, *International Journal of Man-Machine Studies*, **11**, 437–445.

[62] Halpern, J. Y. (1989) An analysis of the first-order logics of probability, *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Sydney, 1375–1381.

[63] Heckerman, D. E., Horvitz, E. J., Nathwani, B. N. (1992) Towards normative expert systems: Part I. The Pathfinder Project. *Methods of Information in Medecine*, **31**, 90–105.

[64] Heinsohn, J. (1991) A hybrid approach for modelling uncertainty in terminological logics, *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Uncertainty*, Marseille, 198–205.

[65] Herskovits, E. and Cooper, G. F. (1990) Kutató: an entropy driven system for construction of probabilistic expert systems from databases, *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, Cambridge, MA, 54–62.

[66] Horsch, M. C. and Poole, D. (1990) A dynamic approach to probabilistic inference using Bayesian networks, *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, Cambridge, MA, 155–161.

[67] Hughes, G. E. and Cresswell, M. J. (1968) *An introduction to modal logic* Methuen and Co., London.

[68] Imieliński, T. (1991) Incomplete deductive databases, *Annals of Mathematics and Artificial Intelligence* **3** 259–294.

[69] Imieliński, T. and Lipski, W. (1984), Incomplete information in relational databases, *Journal of the Association for Computing Machinery*, **31** (4), 761–791.

[70] Jaynes, E. T. (1979) Where do we stand on maximum entropy?, in *The maximum entropy formalism*, (R. D. Levine and M. Tribus, eds), MIT Press, Cambridge MA.

[71] Kießling, W., Łukasiewicz, T., Köstler, G., and Güntzer, U. (1994) The TOP database model—taxonomy, object-orientation and probability, *Proceedings of the Workshop on Deductive Systems and Databases*, Ithaca, NY.

[72] Konolige, K. (1989) On the relation between autoepistemic logic and cricumscription, *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Sydney.

[73] Konolige, K. (1988) On the relation between default and autoepistemic logic, *Artificial Intelligence*, **35**, 343–382.

[74] Kornatzky, Y. and Shimony, S. E. (1994) A probabilistic object-oriented data model, *Data and Knowledge Engineering*, **12**, 143–166.

[75] Krause, P., Ambler, S., Elvang-Gøransson, M., and Fox, J. (1993) A logic of argumentation for uncertain reasoning, *Computational Intelligence*, **11**, 113–131.

[76] Kwan, S. Olken, F. and Rotem, D. (1993) Uncertain, incomplete and inconsistent data in scientific and statistical databases, *Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems: From Needs to Solutions*, Catalina Island.

[77] Laskey, K. B. and Lehrner, P. E. (1990) Assumptions, beliefs and probabilities, *Artificial Intelligence*, **42**, 65–77.

[78] Lauritzen, S. L. and Spiegelhalter, D. J. (1988) Local computations with probabilities on graphical structures and their application to expert systems, *The Journal of the Royal Statistical Society*, **B 50**, 2, 157–224.

[79] Lee, S. K. (1992) An extended relational database model for uncertain and imprecise information, *Proceedings of the 18th Conference on Very Large Databases*, Vancouver, BC, 211–220.

[80] Leung, K. S., Wong, M. H. and Lam, W. (1989) A fuzzy expert database system, *Data and Knowledge Engineering*, **4** 287–304.

[81] Lien, Y. E., (1979) Multivalued dependencies with null values in relational databases, *Proceedings of the 5th International Conference on Very Large Databases*, Rio de Janeiro, 61–66.

[82] Lifschitz, V. (1986) Pointwise circumscription: preliminary report, *Proceedings of the 5th National Conference on Artificial Intelligence*, Philadelphia.

[83] Lin, F. and Shoham, Y. (1989) Argument systems: a uniform basis for nonmonotonic reasoning, *Proceedings of the International Conference on Knowledge Representation and Reasoning*, Toronto, 245–255.

[84] Lipski, W. (1979) On semantic issues connected with incomplete information databases, *ACM Transactions on Database Systems*, **4**, 262–296.

[85] Loui, R. P. (1987) Defeat among arguments: a system of defeasible inference, *Computational Intelligence*, **3** 100–106.

[86] Mamdani, E. H. and Gaines, B. R. (1981) (Editors) *Fuzzy reasoning and its applications*, Academic Press, New York.

[87] Martin, T. P., Baldwin, J. F. and Pilsworth, B. W. (1987) The implementation of FProlog—a fuzzy prolog interpreter, *Fuzzy Sets and Systems*, **23**, 119–129.

[88] McCarthy, J. (1986) Applications of circumscription to formalising commonsense knowledge, *Artificial Intelligence*, **28**, 89–116.

[89] McCarthy, J. (1980) Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence* **13**, 27–39.

[90] McDermott, D. V. Nonmonotonic logic II: nonmonotonic modal theories. *Journal of the ACM* **29**, 37–57 1982.

[91] McDermott, D. V. and Doyle, J. (1980) Non-monotonic logic I. *Artificial Intelligence* **13**, 41–72.

[92] McLeish, M. (1989) Nilsson's Probabilistic entailment extended to Dempster-Shafer theory, in *Uncertainty in Artificial Intelligence 3*, L. N. Kanal, T. S. Levitt, and J. F. Lemmer eds., Elsevier Science Publishers B. V., North-Holland.

[93] McLeish, M (1988) Probabilistic logic: some comments and possible use for nonmonotonic reasoning, in *Uncertainty in Artificial Intelligence 2*, J. F. Lemmer and L. N. Kanal eds. Elsevier Science Publishers B. V., North-Holland.

[94] Moore, R. C. (1985) Semantical considerations on nonmonotonic logic. *Artificial Intelligence* **25**, 75–94.

[95] Morrissey, J. M. (1992) Representing and manipulating uncertain data, *International Journal of Man-Machine Studies*, **36**, 183–189.

[96] Morrissey, J. M. (1990) Imprecise information and uncertainty in information systems, *ACM Transactions on Information Systems*, **8**, 159–180.

[97] Motro, A. (1993) Sources of uncertainty in information systems, *Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems: From Needs to Solutions*, Catalina Island, CA.

[98] Motro, A. (1990) Accomodating imprecision in database systems: issues and solutions, *SIGMOD RECORD*, **19**, 4, 69–74.

[99] Musman, S. A and Chong, L. W. (1993) A study of scaling issues in Bayesian belief nets for ship classification, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, Washington.

[100] Neapolitan, R. E. (1990) *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*, John Wiley & Sons Inc., New York.

[101] Ng, R. T. and Subrahmanian, V. S. (1992) Probabilistic logic programming, *Information and Computation*, **101**, 150–201.

[102] Ng, R. T. and Subrahmanian, V. S. (1992) Empirical probabilities in monadic deductive databases, *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Stanford, 215–222.

[103] Ng, R. T. and Subrahmanian, V. S. (1991) Non-monotonic negation in probabilistic deductive databases, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, 249–256.

[104] Ng, R. T. and Subrahmanian, V. S. (1991) A semantical framework for supporting subjective and conditional probabilities in deductive databases, *Proceedings of the 8th International Conference on Logic Programming*, Paris, 566–580.

[105] Nie, J-Y. (1992) Towards a probabilistic modal logic for semantic-based information retrieval, *Proceedings of the 15th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, N. Belkin, P. Ingerwersen and A. Pejtersen, eds., 140–151.

[106] Nilsson, N. (1993) Probabilistic logic revisited, *Artificial Intelligence*, **59**, 39–42.

[107] Nilsson, N. (1986) Probabilistic logic, *Artificial Intelligence*, **28**, 71–87.

[108] Olesen, K. G., Kjaerulff, U., Jensen, F., Jensen, F. V., Falck, B., Andraessen, S. and Andersen, S. K. (1989) A MUNIN network for the median nerve—a case study in loops, *Applied Artificial Intelligence*, **3**, 385–404.

[109] Parsons, S. (1996) *Qualitative approaches to reasoning under uncertainty*, MIT Press, to appear.

[110] Parsons, S. (1994) Hybrid models of uncertainty in protein topology prediction, *Applied Artificial Intelligence*, **9**, 335–351.

[111] Parsons, S. Kubat, M. and Dohnal, M. (1995) A rough set approach to reasoning under uncertainty, *Journal of Experimental and Theoretical AI*, **7**, 175–193.

[112] Pawlak, Z. (1991) *Rough sets: theoretical aspects of reasoning about data*, Kluwer Academic Publishers, Dordrecht.

[113] Pearl, J. (1988) *Probabilistic reasoning in intelligent systems—networks of plausible inference*, Morgan Kaufmann Publishers Inc.

[114] Pittarelli, M. (1994) An algebra for probabilistic databases, *IEEE Transactions on Knowledge and Data Engineering*, **6**, 293–303.

[115] Pittarelli, M. (1990) Probabilistic databases for decision analysis, *International Journal of Intelligent Systems*, **5**, 209–236.

[116] Pollack, J. (1992) How to reason defeasibly, *Artificial Intelligence*, **57**, 1–42.

[117] Poole, D. (1993) Probabilistic horn abduction and Bayesian networks, *Artificial Intelligence*, **64**, 81–129.

[118] Poole, D. (1991) Representing Bayesian networks within probabilstic horn clause abduction, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, 271–278.

[119] Poole, D. (1989) Explanation and deduction: an architecture for default and abductive reasoning, *Computational Intelligence*, **5**, 97–110.

[120] Poole, D. L. (1988) A logical framework for default reasoning, *Artificial Intelligence*, **36**, 27–47.

[121] Poole, D. L., Goebel, R. G. and Aleliunas, R. (1987) Theorist: a logical reasoning system for defaults and diagnosis, in *The Knowledge Frontei: essays in the representation of knowledge*, N. Cercone and G. McCalla eds., Springer Verlag, New York.

[122] Prade, H. (1984) Lipski's approach to incomplete information databases restated and generalized in the setting of Zadeh's possibility theory, *Information Systems*, **9**, 27–42.

[123] Prade, H. and Testemale, C. (1984) Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries, *Information Sciences*, **34**, 115–143.

[124] Prade, H. and Testemale, C. (1987) Fuzzy relational databases: representational issues and reduction using similarity measures, *Journal of the American Society for Information Science*, **38**, 118–126.

[125] Prade, H. and Testemale, C. (1987) Representation of soft constraints and fuzzy attribute values by means of possibility distributions in databases, in *Analysis of Fuzzy Information, Volume II, Artificial Intelligence and Decision Systems*, J. C. Bezdek, ed., CRC Press, Boca Raton, Florida, 213–229.

[126] Prade, H. and Testemale, C. (1987) Application of possibility and necessity measures to documentary information retrieval, in *Uncertainty in Knowledge-Based Systems*, B. Bouchon and R. R. Yager, eds. Springer-Verlag Lecture Notes in Computer Science 286, Berlin, 265–274.

[127] Provan, G. (1991) Dynamic network updating techniques for diagnostic reasoning *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, 279–286.

[128] Provan, G. (1988) Solving diagnostic problems using extended assumption-based truth maintenance systems: foundations, Technical Report 88-10, Department of Computer Science, University of British Columbia.

[129] Quinlan, J. R. (1983) INFERNO: a cautious approach to uncertain inference, *The Computer Journal*, **26** 255–269.

[130] Reiter, R. (1980) A logic for default reasoning. *Artificial Intelligence* **13**, 81–132.

[131] Reiter, R. (1978) On closed world databases, in *Logic and databases*, H. Gallaire and J. Minker eds., Plenum Press, New York.

[132] Van Rijsbergen, C. J. (1986) A non-classical logic for information retrieval, *The Computer Journal*, **29**, 481–485.

[133] Saffiotti, A. (1992) A belief function logic, *Proceedings of the 10th National Conference on Artificial Intelligence*, Boston, MA, 642–647.

[134] Saffiotti, A. (1992) A belief function logic: preliminary report, Technical Report, TR/IRIDIA/91-25, IRIDIA, Université Libre de Bruxelles.

[135] Saffiotti, A. (1990) A hybrid framework for representing uncertain knowledge,*Proceedings of the 8th National Conference on Artificial Intelligence*, Boston, MA, 653–658.

[136] Saffiotti, A. (1987) An AI view of the treatment of uncertainty, *The knowledge Engineering Review*, **2**, 75–97.

[137] Saffiotti, A., Parsons, S. and Umkehrer, E. (1994) Comparing uncertainty management techniques, *Microcomputers in Civil Engineering—Special Issue on Uncertainty in Expert Systems*, **9**, 367–380.

[138] Saffiotti, A. and Umkehrer, E. (1991) Pulcinella: a general tool for propagating uncertainty in valuation networks, *Proceedings of the Seventh Conference om Uncertainty in Artificial Intelligence*, Los Angeles, 323–331.

[139] Saffiotti, A. and Umkehrer, E, (1991) Automatic generation of valuation networks from general clauses, Technical Report, TR/IRIDIA/91-24, IRIDIA, Université Libre de Bruxelles.

[140] Schocken, S. and Hummel, R. A. (1993) On the use of the Dempster Shafer model in information indexing and retrieval applications, *International Journal of Man Machine Studies*, **39**, 843–879.

[141] Shafer, G. (1976) *A mathematical theory of evidence*, Princeton University Press.

[142] Shenoi, S. and Melton, A. (1989) Proximity relations in the fuzzy relational database model, *Fuzzy sets and systems*, **31**, 285–296.

[143] Shenoy, P. P. (1992) Using possibility theory in expert systems, *Fuzzy Sets and Systems*, **51**, 129–142.

[144] Shenoy, P. P. and Shafer, G. (1990) Axioms for probability and belief function propagation, in *Uncertainty in Artificial Intelligence 4*, R. D. Shacter, T. S. Levitt, L. N. Kanal, and J. F. Lemmer eds., Elsevier Science Publishers, North-Holland.

[145] Smets, Ph. (1988) Belief functions, in *Non-Standard Logics for Automated Reasoning*, Ph. Smets, E. H. Mamdani, D. Dubois and H. Prade eds., Academic Press, London.

[146] Smets, Ph. and Magrez, P. (1987) Implication in fuzzy logic, *International Journal of Approximate Reasoning*, **1**, 327–347.

[147] Smith, C. A. B. (1961) Consistency in statistical inference and decision, *Journal of the Royal Statistical Society*, **B23**, 218–258.

[148] Smithson, M (1989) *Ignorance and uncertainty: emerging paradigms.* Springer Verlag, New York.

[149] Srinivas, S. and Breese, J. (1990) IDEAL: a software package for analysis of influence diagrams, *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, Cambridge, MA, 212–219.

[150] Srinivas, S., Russell, S., and Agogino, A. (1989) Automated construction of sparse Bayesian networks from unstructured probabilistic models and domain information, *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, Windsor, Ontario, 343–350.

[151] Studeny, M. (1992) Conditional independence relations have no finite complete characterisation, in *Information Theory, Statistical Decision Functions, Random Processes: Transactions of the 11th Prague Conference*, S. Kubík and J. A. Víšek eds., Kluwer, Dordrecht.

[152] Sucar, L. E., Gillies, D. F. and Gillies, D. A. (1993) Objective probabilities in expert systems, *Artificial Intelligence*, **61**, 187–208.

[153] Thöne, H., Güntzer, U. and Kießling, W. (1992) Towards precision of probabilistic bounds propagation, *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, Stanford, 315–322.

[154] Turtle, H. R. and Croft, W. B. (1993) Uncertainty in information retrieval systems, *Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems: From Needs to Solutions*, Catalina Island, CA.

[155] Turtle, H. R. and Croft, W. B. (1990) Inference networks for document retrieval, *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, ACM, 1–24.

[156] Umano, M. (1983) Retrieval from fuzzy database by fuzzy relational algebra, *Proceedings of the IFAC Symposium*, Marseille, 1–6.

[157] Wellman, M. P., Breese, J. S., and Goldman, R. P. (1992) From knowledge bases to decision models, *The knowledge engineering review*, **7**, 35–53.

[158] Waugh, K. G., Williams, M. H., Kong, Q., Salvani, S., and Chen, G. (1990) Designing SQUIRREL: an extended SQL for a deductive database system, *The Computer Journal*, **33**(6), 535–546.

[159] Wen, W. X. (1991) From relational databases to belief networks, *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, Los Angeles, 406–413.

[160] Williams, M. H. and Kong, Q. (1991) Time and incompleteness in a deductive database, in *Uncertainty in Knowledge Bases*, B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh eds., Springer Verlag, Berlin, 443–455.

[161] Williams, M. H. and Kong, Q. (1988) Incomplete information in a deductive database, *Data and Knowledge Engineering*, **3**, 197–220.

[162] Wong, E. (1982) A statistical approach to incomplete information in database systems, *ACM Transactions on Database Systems*, **7**, 470–488.

[163] Wong, S. K. M., Xiang, Y. and Nie, X. (1994) Representation of bayesian networks as relational databases, *Proceedings of the International Conference on Information Processing and Management of Uncertainty*, Paris, 159–165.

[164] Zadeh, L. A. (1984) Fuzzy probabilities, *Information Processing & Management*, **3**, 363–372.

[165] Zadeh, L. A. (1983) Commonsense knowledge representation based on fuzzy logic, *IEEE Computer*, October, 1983, 61–65.

[166] Zadeh, L. A. (1983) The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets and Systems*, **11**, 199–227.

[167] Zadeh, L. A. (1978) Fuzzy sets as the basis for a theory of possibility, *Fuzzy Sets and Systems*, **1**, 1–27.

[168] Zadeh, L. A. (1965) Fuzzy sets *Information and Control*, **8**, 338–353.

[169] Zaniolo, C. (1982) Database relations with null values, *Proceedings of the ACM Symposium on the Principles of Database Systems*, Los Angeles, CA, 27–33.

[170] Zicari, R. (1993) Databases and Incomplete Information, *Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems: From Needs to Solutions*, Catalina Island, CA.

[171] Zicari, R. (1990) Incomplete Information in Object-Oriented Databases, *SIGMOD RECORD*, **19**, 3, 5–16.

[172] Zvelli, A. (1986) A fuzzy relational calculus, *Proceedings of the 1st International Conference on Expert Database Systems*, Charleston, South Carolina, 311–326.