


A Belief Revision Framework For Probabilistic Object-Oriented Databases

Zafer D. Ozdemir, (Email: ozdemir@muohio.edu), Miami University

ABSTRACT

While past research extended the conventional object-oriented data model using probability theory to better capture the uncertainty of the real world, database modification has not been addressed to date. This paper builds on previous work by Kornatzky and Shimony (1994) to address database modification for probabilistic object-oriented databases, and proposes a framework for belief revision. According to the framework, beliefs on data values are revised with the arrival of new information. The proposed belief revision framework can be used to update data stored in an object-oriented database.

INTRODUCTION AND MOTIVATION

 One of the important unsolved problems in databases is the ability to integrate models of uncertainty (Silberschatz et al. 1991), primarily because many real-world decision problems require the ability to represent, manipulate, and reason with uncertain knowledge (Sarkar et al. 1996). Such uncertainties arise from different sources such as measurement errors, cognitive errors, approximation errors, calculation errors, multiple data sources that conflict with one another, the dynamic nature of the real world, and partially unknown environment (Dey and Sarkar 1996, Lakshmanan et al. 1997). Therefore, today's database systems must handle uncertainties in the data they store.

Consider, for example, the case where a database is used to monitor the locations of battleships during war (Wong 1982). For every decision regarding deployment, battle, or other operations, it is essential to consider the approximate locations of the ships. Not having the ability to represent the inherent uncertain nature of the data, the conventional data models cannot be used. Similarly, a financial institution may be interested in storing some non-deterministic attributes of different companies for the purpose of making the right investment decisions (Barbará et al. 1992). Marketing and production decisions are mostly based on expected customer behavior patterns, which are rarely deterministic (Dey and Sarkar 1996). Consolidating heterogeneous databases presents another challenge. For example, in order to track all its alumni, a college may need to develop a database by consolidating data from several sources, including the registration system, the central alumni database, and departmental records within the college (Dey and Sarkar 2002). One may frequently encounter inconsistent attribute values stored in such heterogeneous sources. If the correctness of these conflicting attribute values cannot be easily verified, the designer would either retain only one of the values and discard the rest, or represent these multiple values in an ad hoc fashion (using multiple fields). Other examples are plentiful in the literature (Wong 1982, Buckles and Petry 1983, Prade and Testemale 1984, Zemankova and Kandel 1985, Barbará et al. 1992).

The relational and object-oriented data models offer no means to handle incomplete and uncertain data described in the preceding examples. They ignore all uncertain data and focus primarily on data values that are known with certainty; unknown or uncertain data items are represented with null values (Codd 1979, Date 1986). Therefore, the decision making process requires data models that can organize real-world information and update it consistently based on changes in the real world (Dey and Sarkar 2000). If a *probabilistic* data model were used for the alumni database in the above example, the uncertainty could be captured by assigning a probability distribution to the multiple values of attributes. This would allow the designer to retain all the information in a systematic and flexible manner.

Another important aspect of a data model is the modification of existing data. Most databases go through the normal process of addition, deletion, and update of data items. In a probabilistic database, these modifications can result from two sources. First, the states of objects in the database may continuously change due to the dynamic nature of the world. Also, more reliable information about the existing objects may be obtained. Obviously, any change in the belief about one or more objects should be incorporated into the database system. In a probabilistic database where data items are stored along with the user's degree of belief in them, the modification of data is equivalent to revision of belief about the data (Dey and Sarkar 2000). Interestingly, prior research on updating probabilistic databases is limited at best. The purpose of this research is to propose a framework for belief revision in a probabilistic object-oriented database. In doing this, we take the probabilistic approach since probability theory is mathematically the best understood paradigm for modeling and manipulating uncertain information (Lakshmanan et al. 1997).

The rest of the paper is organized as follows. Section 2 provides an overview of related work. Section 3 outlines a probabilistic object-oriented model, and Section 4 introduces the Bayesian framework for belief revision. Section 5 discusses and illustrates how the proposed belief revision scheme could be applied in different circumstances. Section 6 concludes and offers future research directions.

RELATED WORK

A significant amount of research has discussed the need for an extension of data models and focused on the representation of uncertainty. Cavallo and Pitarelli (1987) and Pittarelli (1994) extended the relational model to represent uncertainty using probability theory. Their model assigns every tuple a joint probability of all the attribute values in that tuple. Raju and Majumdar (1988) generalized the basic relational concepts by using fuzzy set theory. Barbará et al. (1992) proposed a probability-based extension of the relational model that adopts a non-first normal form view of probabilistic relations. Dey and Sarkar (1996) proposed a first normal form structure for probabilistic relations and developed the associated algebra as a consistent extension of the relational algebra. Lakshmanan et al. (1997) proposed an extended relational model based on probability intervals and developed a generalized algebra without making any assumption about interdependencies among events. Dey and Sarkar (2002) generalized the concept of functional dependency to stochastic dependency and used that to extend the scope of normal forms to probabilistic relational databases.

In contrast to the amount of research on probabilistic *relational* databases, there has been less work on probabilistic *object-oriented* databases. Kornatzky and Shimony (1994) outlined a probabilistic object-oriented data model to represent uncertainty at attribute and class hierarchy levels. Eiter et al. (2001) extended the work of Kornatzky and Shimony to develop an algebra to handle object bases with uncertainty. Finally, Biazzo et al. (2003) proposed a data model and algebra for temporal probabilistic object bases that let the designer to specify the probability with which an event occurs at a given point in time.

The issue of belief revision was first studied in the context of expert systems (Pearl 1988, Neapolitan 1990). Research in this domain used Bayesian networks for knowledge representation. More recently, Dey and Sarkar (2000) studied a belief revision scheme specifically for a probabilistic relational model. Both Kornatzky and Shimony (1996) and Eiter et al. (2001) stressed the importance of performing updates in probabilistic object-oriented databases and suggested the use of a probabilistic reasoning mechanism for belief revision. Building on the influential work of Kornatzky and Shimony, this research introduces a belief revision framework for the richer object-oriented model, as compared to the relational model.

THE PROBABILISTIC OBJECT-ORIENTED DATA MODEL

This section summarizes the key features of the probabilistic object-oriented data model developed by Kornatzky and Shimony (1994).

The Object-Oriented Data Model

In the deterministic object-oriented model, real-world data includes *values* and *objects*. A value has a *type*, which is recursively definable from atomic types and the set and tuple type constructors. Each object has an *identity* and state, and belongs to a *class*. Thus, an object identifier (oid) together with its state is called an *object*, and is an *instance* of the class to which the oid is assigned. The fact that an object o is an instance of a class C is denoted by *instance-of*(o, C). A class may inherit its type from other classes through a subclass hierarchy. The syntax and semantics of types are defined using a given set of atomic type and class names. Complex types are built using tuple constructors.

Example 1: An Employee type is defined as

Type Employee = [EmpID: integer, Name: text, Dept: text, Skills: {Skill}]

Example 2: A value of Employee type is

[EmpID: 100, Name: "Jane", Dept: "Marketing", Skills: {Java, Basic}]

The $<$ operator defines subclass (inheritance) relationships between classes. The set of immediate subclasses of a class C , denoted by \underline{C} , is $\{C' \mid C' < C \cup \{C'' \mid C'' < C' < C\}\}$. The subclass hierarchy is represented by a directed graph, whose nodes correspond to classes and whose edges point from a class to its immediate subclasses.

UNCERTAIN ATTRIBUTES OF OBJECTS

Uncertainty in an object's attribute values is described using *probabilistic attributes*. A probabilistic attribute is viewed as a probability distribution over the set of values of the attribute. Formally, a probabilistic attribute A of type t is a probabilistic distribution over the domain of type t . The probability of each possible tuple is represented by b . For ease of exposition, the names of all probabilistic attributes are underlined in the examples below. In the next example, salary is the only probabilistic attribute, and the employee with EmpID 100 has a salary of 40K with 0.4 probability and 50K with 0.5 probability. Note that if these are the only tuples about this employee, then the total probability is less than 1, indicating that this object does not exist with certainty. This is contrary to the original model of Korntazky and Shimony (1994), where objects are assumed to exist with certainty. We relax this assumption here to increase the flexibility of the model.

Example 3: Illustration of an Employee instance with the probabilistic attribute Salary.

[EmpID: 100, Name: "Jane", Dept: "Marketing", Salary: 40K; $\beta = 0.4$]

[EmpID: 100, Name: "Jane", Dept: "Marketing", Salary: 50K; $\beta = 0.5$]

When an object has multiple probabilistic attributes, β represents the probability of the event that all probabilistic attributes take the associated values.

UNCERTAINTY IN THE CLASS HIERARCHY

Object classes may also be uncertain. We consider only the uncertainty in subclass membership with no multiple inheritance.¹ Specification of conditional probabilities of subclass membership determines the class probability distribution of an object. For $C' \hat{\in} \underline{C}$, $\text{Pr}(\text{instance-of}(o, C') \mid \text{instance-of}(o, C))$ denotes

¹ The potential of multiple inheritance is not considered here since the consistency of probabilities cannot be preserved. See page 155 of Korntazky and Shimony (1994) for a proof.

the probability that the object o is an instance of the immediate subclass C' , given that it is an instance of the parent class C .

Example 4: Consider the class hierarchy given in Figure 1. The directed arc from Person to Employee and the value 0.5 next to the arc implies

$$\Pr(\text{instance-of}(o_1, \text{Employee}) \mid \text{instance-of}(o_1, \text{Person})) = 0.5.$$

That is, with 0.5 probability the object is an instance of the Employee subclass, given that it is an instance of the Person class.

Two definitions are in order. The *inclusive probability*, denoted by $a(o, C)$, is the probability that the object o is an instance of class C or any of its subclasses. The *exclusive probability*, denoted by $a^*(o, C)$, is the probability that the object o is an instance of class C , but is not an instance of any of its subclasses. The inclusive and exclusive probabilities can be calculated from conditional subclass probabilities, and they play an important role in the belief revision scheme presented in Section 4.

Example 5: Based on the information in Figure 1, the inclusive and exclusive probabilities for the object o_1 can be found as:

$$a(o_1, \text{Person}) = 1$$

$$a^*(o_1, \text{Person}) = 1 - 0.5 - 0.3 = 0.2$$

THE BAYESIAN FRAMEWORK

This section summarizes the Bayesian framework for belief revision proposed originally by Dey and Sarkar (2000), and explains the reasoning behind Jeffrey's rule and how it can be used for database modification. This framework is applied to the probabilistic object-oriented data model in Section 5.

Let $B_i, i = 1, 2, \dots, n$, be a set of exhaustive and mutually exclusive propositions. The belief in another proposition \bar{B} , which depends on propositions B_i , can be formulated by Baye's rule as:

$$\Pr(\bar{B}) = \sum_{i=1}^n \Pr(\bar{B} \mid B_i) \Pr(B_i).$$

Using this rule, the probability of \bar{B} can be updated when the probabilities of B_i change.

Jeffrey's Kinematics

Jeffrey (1983) proposed a procedure for updating the degree of belief in a proposition \bar{B} when a new evidence e is received, assuming that the new evidence affects the probability of \bar{B} indirectly through changing the degree of belief in B_i .² Using Jeffrey's rule of probability kinematics, the new degree of belief, $\Pr(\bar{B} \mid e)$, can be stated as:

² The conditional probabilities are assumed to remain the same.

$$\Pr(\bar{B} | e) = \prod_{i=1}^n \Pr(\bar{B} | B_i) \Pr(B_i | e).$$

The next section applies Jeffrey’s rule to database modification, assuming that the conditional distribution of an uncertain attribute or class of an object remains the same, unless the new information is about the conditional distribution itself.

THE BELIEF REVISION SCHEME FOR THE OBJECT-ORIENTED DATA MODEL

In consolidating heterogeneous data sources, some data items may overlap while some others conflict. For example, the salary information of an employee may have been stored differently in two different data sources. With conventional data models, given that the correctness of the two attribute values cannot be easily verified, the user would be forced to retain only one of the values and discard the other. A probabilistic database allows the representation of both values simultaneously, avoiding loss of some valuable information (perhaps the correct salary) in the process. Another nice feature of a probabilistic representation is that the user has the option of revising his or her belief about a particular data item. After the consolidation process is over, new information may become available that could change the user’s opinion about the probability assignments to the salary information. This section illustrates how the data values should be modified using the Bayesian framework in such circumstances. There are two types of data in the examples below: pre-existing data and new information. Pre-existing data shows the data items prior to the arrival of the new information, while the new information represents the changed opinion of the user. Notice that, in all the updates below, consistency is preserved such that the sum of probabilities for any given object is less than or equal to one.

Belief Revision In Attribute Values

There are three different ways by which new information could be processed for probabilistic attributes: modifications for atomic attributes, joint distribution of attributes, and set valued attributes.

Atomic Attributes

The simplest case is when only a single attribute takes a probabilistic, atomistic value.

Example 6:

| | |
|-----------------------|--|
| Pre-existing data | [EmpID: 100, Department: “Marketing”, Salary: 40K; β = 0.4] [EmpID: 100, Department: “Marketing”, Salary: 50K; β = 0.6] |
| New info | [EmpID: 100, Department: “Marketing”, Salary: 50K; β = 0.8] |
| Data after the update | [EmpID: 100, Department: “Marketing”, Salary: 40K; β = 0.2] [EmpID: 100, Department: “Marketing”, Salary: 50K; β = 0.8] |

In this case, a tuple similar to the incoming information already existed in the database, but with a different probability. The probability of this tuple is set to the new value and the probabilities of other tuples are modified to preserve consistency. Note that the likelihood of salary being equal to 40K decreases even though no new information is received about the likelihood of this event. This likelihood could increase only with the arrival of new information on this event.

Joint Distributions Of Attributes

The update mechanism for multiple probabilistic attributes is identical to that proposed for the relational data model by Dey and Sarkar (2000).

Example 7:

| | |
|-----------------------|--|
| Pre-existing data | [EmpID: 100, Department: “Marketing”, Salary: 40K; $\beta = 0.2$] [EmpID: 100, Department: “Production”, Salary: 50K; $\beta = 0.2$] [EmpID: 100, Department: “Production”, Salary: 60K; $\beta = 0.6$] |
| New info | [EmpID: 100, Department: “Production”; $\beta = 0.4$] |
| Data after the update | [EmpID: 100, Department: “Marketing”, Salary: 40K; $\beta = 0.2$] [EmpID: 100, Department: “Production”, Salary: 50K; $\beta = 0.1$] [EmpID: 100, Department: “Production”, Salary: 60K; $\beta = 0.3$] |

Note that the probabilities of other tuples do not change when the likelihood of an event is adjusted downward because of the new information, indicating that the sum of probabilities for a given object can be less than one.

Set Valued Attributes

The next example illustrates the belief revision process for a set valued attribute.

Example 8:

| | |
|-----------------------|---|
| Pre-existing data | [EmpID: 100, Skill: {Java, C++}; $\beta = 0.3$] [EmpID: 100, Skill: {Java, Basic}; $\beta = 0.3$] [EmpID: 100, Skill: {Cobol, Fortran}; $\beta = 0.4$] |
| New info | $\Pr(\text{EmpID} = 100, \text{Java} \in \text{Skill}) = 0.8^3$ |
| Data after the update | [EmpID: 100, Skill: {Java, C++}; $\beta = 0.4$] [EmpID: 100, Skill: {Java, Basic}; $\beta = 0.4$] [EmpID: 100, Skill: {Cobol, Fortran}; $\beta = 0.2$] |

Based on the previous conditional distribution, the new probability (0.8) is distributed among tuples that have “Java” as one of elements of the Skill attribute. The probability of the remaining tuple is adjusted to maintain consistency.

Belief Revision in Class Hierarchy

Uncertainty in class hierarchy can be represented in two alternative forms: using conditional subclass probabilities and using inclusive and/or exclusive probabilities. In both cases the class hierarchy is assumed to be stationary; that is, the new information about class membership always refers to an existing class in the hierarchy.

Distribution Of Instances Of A Class Among Its Subclasses

This section considers the case where the new information is concerned with the conditional subclass probabilities. Recall that if an object is an instance of class C , it is either an instance of one of the subclasses of C or it is not an instance of any of the subclasses, in which case the object is said to be an exclusive instance of class C .

According to Figure 1:

$$\Pr(\text{instance-of}(o_1, \text{HourlyEmployee}) \mid \text{instance-of}(o_1, \text{Employee})) = 0.4$$

$$\Pr(\text{instance-of}(o_1, \text{Consultant}) \mid \text{instance-of}(o_1, \text{Employee})) = 0.2$$

³ Note the difference between this tuple and [EmpID: 100, Skill: {Java}; $\beta = 0.8$]

The above probabilities imply that the object with oid o_1 neither belongs to the HourlyEmployee nor the Consultant class with probability 0.4, which is the exclusive probability of Employee subclass ($a^*(\text{Employee})$).

Example 9: Suppose that the new information is:

$$\Pr(\text{instance-of}(o_1, \text{Consultant}) \mid \text{instance-of}(o_1, \text{Employee})) = 0.4$$

The conditional subclass probabilities after the update are:

$$\Pr(\text{instance-of}(o_1, \text{HourlyEmployee}) \mid \text{instance-of}(o_1, \text{Employee})) = 0.3$$

$$\Pr(\text{instance-of}(o_1, \text{Consultant}) \mid \text{instance-of}(o_1, \text{Employee})) = 0.4$$

The belief revision process for this case is similar to the one used in updating attribute values. The exclusive and inclusive probabilities are calculated based on prior conditional distributions. In the context of Example 9, since the prior exclusive probabilities $a^*(\text{Employee})$ and $a^*(\text{HourlyEmployee})$ were equal (both 0.4), the posterior exclusive probabilities are also equal (both 0.3).

Partial Classification Of Instances

The new information can also be in the form of inclusive and exclusive class probabilities.

Example 10: In Figure 1, inclusive and exclusive probabilities for the Employee subclass can be stated as:

$$\begin{array}{ll} a(\text{Employee}) = 0.5 & a^*(\text{Employee}) = 0.2 \\ a(\text{HourlyEmployee}) = 0.2 & a(\text{Consultant}) = 0.1 \\ a(\text{PermanentConsultant}) = 0.06 & a(\text{TemporaryConsultant}) = 0.02 \end{array}$$

If, according to the new information, the probability of belonging to Consultant class or any of its subclasses were zero, the updated inclusive and exclusive class probabilities would be:

$$\begin{array}{ll} a(\text{Employee}) = 0.5 & a^*(\text{Employee}) = 0.25 \\ a(\text{HourlyEmployee}) = 0.25 & a(\text{Consultant}) = 0 \\ a(\text{PermanentConsultant}) = 0 & a(\text{TemporaryConsultant}) = 0 \end{array}$$

Belief Revision In Both Attribute Values And Class Hierarchy

While the proposed belief revision scheme can be used to handle uncertainty in attributes and class hierarchy, the interaction of the two types of uncertainties can complicate the process significantly. The next example illustrates two cases where the new information is about (i) an attribute and (ii) a class distribution.

Example 11:

| | |
|----------------------------|---|
| Pre-existing data | Employee: [EmpID: 100, Name: "Jane"; $\beta = 0.2$] Employee: [EmpID: 100, Name: "Judy"; $\beta = 0.3$] Consultant: [EmpID: 100, Name: "Jane", Project: "SAP"; $\beta = 0.1$] Consultant: [EmpID: 100, Name: "Judy", Project: "Oracle"; $\beta = 0.1$] |
| New info (i) | [EmpID: 100, Name: "Jane"; $\beta = 0.4$] |
| Data after the update (i) | Employee: [EmpID: 100, Name: "Jane"; $\beta = 0.4$] Employee: [EmpID: 100, Name: "Judy"; $\beta = 0.3$] Consultant: [EmpID: 100, Name: "Jane", Project: "SAP"; $\beta = 0.2$] Consultant: [EmpID: 100, Name: "Judy", Project: "Oracle"; $\beta = 0.1$] |
| New info (ii) | $a(\text{Consultant}) = 0.1$ |
| Data after the update (ii) | Employee: [EmpID: 100, Name: "Jane"; $\beta = 0.2$] Employee: [EmpID: 100, Name: "Judy"; $\beta = 0.3$] Consultant: [EmpID: 100, Name: "Jane", Project: "SAP"; $\beta = 0.05$] Consultant: [EmpID: 100, Name: "Judy", Project: "Oracle"; $\beta = 0.05$] |

CONCLUSIONS AND FUTURE RESEARCH

Despite the numerous situations where uncertain data play a critical role in managerial decision making, conventional databases lack the ability to model uncertainty in the data they store. Consequently, several extensions of the relational and object-oriented models have been proposed in the literature. However, with the exception of Dey and Sarkar (2000), these extensions ignore a very important aspect of a probabilistic data model: the modification of data. In a probabilistic database where data items are stored along with the user's degree of belief in them, the modification of data is equivalent to revision of belief about the data.

This paper makes a first attempt to deal with belief revision in a probabilistic object-oriented database. Building on the probabilistic object-oriented model of Kornatzky and Shimony (1994), we discuss the basic assumptions in applying the Bayesian framework and illustrate how the updates would be performed under various circumstances. The framework can accommodate the modifications resulting both from changes in degree of belief about one or more objects (due to better information) and from changes of state of one or more objects (due to an event).

Further work is needed to advance our understanding of this area. The first one is the definition of all relevant operations of a probabilistic object-oriented algebra. The availability of this toolset would structure the formulation of all the belief revision strategies described here, and could be developed along the lines of Dey and Sarkar (1996). Second, developing a prototype probabilistic object-oriented database with data update capability is also needed to assess the efficiency of belief revision mechanism described here.

REFERENCES

1. D. Barbará, H. Garcia-Molina, and D. Porter, The Management of Probabilistic Data, *Transactions on Knowledge and Data Engineering*, 4 (5) (1992) 487-502.
2. V. Biazzo, R. Giugno, T. Lukasiewicz, and V. S. Subrahmanian, Temporal Probabilistic Object Bases, *IEEE Transactions on Knowledge and Data Engineering*, 15(4) (2003) 921-939.
3. B. Buckles and F. Petry, Information-Theoretical Characterization of Fuzzy Relational Databases, *IEEE Transactions on Systems, Man and Cybernetics*, 13(1) (1983) 74-77.
4. R. Cavallo and M. Pitarelli, The Theory of Probabilistic Databases, In Proceedings of the 13th VLDB Conference, Brighton, England, (1987) 71-81.
5. E. Codd, Extending the Database Relational Model to Capture More Meaning, *ACM Transactions on Database Systems*, 4 (4) (1979) 397-434.
6. C. Date, *Relational Database: Selected Writings* (Addison-Wesley, Reading, MA, 1986).

7. D. Dey and S. Sarkar, A Probabilistic Relational Model and Algebra, *ACM Transactions on Database Systems*, 21 (3) (1996) 339-369.
8. D. Dey and S. Sarkar, Modifications of Uncertain Data: A Bayesian Framework for Belief Revision, *Information Systems Research*, 11 (1) (2000) 1-16.
9. D. Dey and S. Sarkar, Generalized Normal Forms for Probabilistic Relational Data, *IEEE Transactions on Knowledge and Data Engineering*, 14 (3) (1996) 485-497.
10. T. Eiter, J. Lu, T. Lukasiewicz, and V. Subrahmanian, Probabilistic Object Bases, *ACM Transactions on Database Systems*, 26 (3) (2001) 264-312.
11. R. Jeffrey, *The Logic of Decision* (University of Chicago Press, Chicago, IL, 1983).
12. Y. Kornatzky and S. Shimony, A Probabilistic Object-Oriented Data Model, *Data and Knowledge Engineering*, 12 (1994) 143-166.
13. L. Lakshmanan, N. Leone, R. Ross, and V. Subrahmanian, ProbView: A flexible probabilistic database system, *ACM Transactions on Database Systems*, 22 (3) (1997) 419-469.
14. R. Neapolitan, *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*, (John Wiley and Sons, New York, 1990).
15. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, (Morgan Kaufmann, San Mateo, CA, 1988).
16. M. Pitarelli, An Algebra for Probabilistic Databases, *IEEE Transactions on Knowledge and Data Engineering*, 6 (2) (1994) 293-303.
17. H. Prade and C. Testemale, Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries, *Information Science*, 34 (1984) 115-143.
18. K. Raju and A. Majumdar, Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems, *ACM Transactions on Database Systems*, 13 (2) (1988) 129-166.
19. Silberschatz, M. Stonebraker and J. Ullman, *Database Systems: Achievements and Opportunities*, *Communications of the ACM*, 34 (10) (1991) 110-119.
20. F. Tseng, A. Chen, and W. Yang, Answering Heterogeneous Database Queries with Degrees of Uncertainty, *Distributed and Parallel Databases: An International Journal*, 1 (3) (1993) 281-302.
21. E. Wong, A Statistical Approach to Incomplete in Database Systems, *ACM Transactions on Database Systems*, 7 (3) (1982) 470-488.
22. M. Zemankova and A. Kandel, Implementing Imprecision in Information Systems. *Information Science*, 37 (1985) 107-141.

Figure 1: Class Hierarchy And Conditional Probabilities For The Object o_1 .

