Alexander Dekhtyar

Department of Computer Science
University of Maryland, College Park, MD, 20742
email: dekhtyar@cs.umd.edu
web: http://www.cs.umd.edu/∼dekhtyar/academ

March 20, 2000

Research Statement

Current Research

My current research deals with probabilistic information in Databases and Logic Programming.

Probabilistic Temporal Databases.(*joint work with Robert Ross and V.S. Subrahmanian*)
A number of database applications (e.g., scheduling databases for transportation providers, time-series stock databases, video extraction applications) require the ability to make statements of the following kind: *Data tuple d from relation R is true at some point of time in the interval $[t_i, t_j]$ with probability between p and p'*. For example, a tuple in a scheduling database may contain information that *a package shipped from Washington, DC to New York via UPS arrives at the destination between 12 and 36 hours after being sent with probability 95% - 100%.* Additionally, the database may contain information that *the probability of arrival at a particular hour for that time interval is distributed uniformly.* Until now, such information could be only represented in *either* purely temporal *or* purely probabilistic databases. Such representations had a number of problems:

- *Problem of compact representation.* Some applications of temporal databases require *the chronon* to be rather small (seconds, or even milliseconds). To represent uncertain information in purely probabilistic databases, one would need to include a tuple with an approoapriate probability for each timepoint when an event could possibly occur. In situations where chronons are small, this would lead to massive amounts of data being stored.
- *Problem of keeping track of probabilistic distributions.* Temporal databases allow compact representation of temporal data. However, once data is represented in a compact form as a temporal tuple, the only probabilistic information accompanying it would be the probability of the tuple being true at the time period specified in the tuple. What is *missing* is the information about finding the probabilities of the tuple being true at any given sub-period.

We have introduced the concept of Probabilistic Temporal Databases, which are designed to address these problems. Our main contributions are summarized as follows.

- We have introduced the concept of a *temporal-probabilistic tuple* or TP-tuple, for short. Intuitively, a TP-tuple allows us to augment classical relational database tuples with temporal-probabilistic data, as well as arbitrary probability distributions. For example, not only can we say "Data tuple d is in relation r at some point of time in the interval $[t_i, t_j]$ with probability between p and p'" but we can also say that the probability mass is distributed over $[t_i, t_j]$ according to an *arbitrary* probability distribution.

- We have shown how given any TP-tuple tp, we may "flatten" tp into a set of *annotated tuples*. In general, the set of annotated tuples associated with a single TP-tuple can be very large — hence, annotated tuples serve as a *purely theoretical device*.

- We have defined a *Theoretical Annotated Temporal Algebra* (TATA) and showed how the classical relational algebra operations can be extended to the case of annotated tuples. Intuitively, the Theoretical Annotated Temporal Algebra provides a theoretical specification of how the TP-Algebra operations must be defined.

- We have introduced a *Temporal-Probabilistic Algebra* (TPA) which directly manipulates TP-tuples *without converting them to annotated tuples*. This has a great advantage, as TP-tuples are very succinct objects. We showed that for each operation α in the Theoretical Annotated Temporal Algebra, there is a corresponding operation α' in the Temporal-Probabilistic Algebra which precisely captures it. Thus, the Temporal-Probabilistic Algebra is a sound and complete way of implementing the declarative semantics for temporal probabilistic data prescribed by the Theoretical Annotated Temporal Algebra. Correctness results are formally proved for each operation.

- We have shown that each operator, whether in the TP-Algebra, or in the Theoretical Annotated Temporal Algebra, can be parameterized by the user's knowledge of the dependencies between events. This is important because, the probability of a complex event like $(e_1 \vee e_2)$ depends upon our knowledge of the dependencies between e_1 and e_2.

- We have established a number of query equivalence and query containment results in TATA and TPA. Work on query optimization is currently underway.

- A prototype Probabilistic Temporal Database System has been implemented. The system has a web interface (http://bester.cs.umd.edu/TP/index.htm) and communicates with a number of commercial database systems, which store the actual data (Paradox, Oracle) via ODBC. Experiments showed the benefits of storing and operating on data in compact (tp-) form as opposed to the expanded (annotated) form.

Temporal Probabilistic Logic Programs. ***(joint work with V.S. Subrahmanian)***

Concurrently with our research on Probabilistic Temporal Databases, we have been studying logical reasoning in situations involving temporal uncertainty. We have proposed the concept of a *Temporal Probabilistic Logic Program* (or TPLP for short). We have defined the syntax of TPLPs and provided a formal model theoretic and fixpoint semantics that are shown to coincide. In this work we considered three types of statements:

- statements about the events that can occur only once (such as letter/package deliveries)

- statements about the events that can occur multiple times or have a duration (e.g., occurrence of a character in movie frames)

- uncertain statements independent of time

Our syntax allows the user to explicitly state the type of each statement (formula) found in a TP-program and our model theory captures reasoning about the temporal probabilities of events of all three types. The fixpoint semantics uses linear programming to compute the tightest probability intervals possible.

We have also studied a subset of the general language where only statements about uniquely occurring events were allowed and established a more efficient fixpoint procedure which does not employ linear programming for this subclass of TP-programs.

We are currently working on query processing procedures for TP-programs, and on finding other subclasses of TP-programs for which the fixpoint semantics will not need linear programming, therefore making more efficient query processing possible.

Optimal Status Sets For Heterogeneous Active Agents ***(joint work with Thomas Eiter, Thomas Schwentick and V.S. Subrahmanian)***

This work is a part of the ongoing IMPACT (Interactive Maryland Platform for Agents Collaborating Together). A part of the framework is a declarative language for Agent Programs which allows agents express their "operating principles" and to determine the actions they must take in order to satisfy them.

The semantics of Agent Programs is given in terms of *Status Sets* - sets of actions which satisfy the rules of the program. Different semantics are possible, and Eiter and Subrahmanian describe a number of semantics.

However, only in very simple cases are we guaranteed that the status set is unique (i.e., that the agent's choice of actions at each step will be deterministic).

Our current work we proposes a way to overcome this. We assume that an agent has an *objective function* which allows it to evaluate each status set that satisfies its program for a given semantics. Out of all such status sets, we will select the one(s) which optimize this objective function.

The goals of this work are:

- to provide a simple but powerful language for describing the objective functions in IMPACT. This language should be compatible with the current IMPACT framework.
- to give complexity analysis of the problem of computing an optimal status set, given an agent state and an objective function. We expect the general case to be quite hard, in part due to the arbitrary nature of the objective function.
- to find a number of important classes of objective functions for which the problem of computing an optimal status set is solvable in polynomial time.
- to construct the algorithms for finding optimal status sets both in the general case, and for all the classes of objective functions for which we can show that the problem is solvable in better time.

Hybrid Probabilistic Logic Programs. (***joint work with V.S. Subrahmanian and M.I. Dekhtyar*****)**

Most current work on probabilistic logic programming, assumes the existence of a simple apiori relationship between the events described (e.g., independence or ignorance). However, this severely limits the applicability of the frameworks.

We have proposed a *hybrid probabilistic logic programming* language which was designed to allow the programmer to reason about the probabilities of event under any assumption about their relationship. In particular, our contributions are:

- First, we have introduced a general axiomatic notion of a *probabilistic strategy*, as the means of computing probabilities under given assumption about the relationship between the events. We showed that a number of well known probabilistic strategies (e.g., independence, positive correlation, ignorance) are special cases of our definition.

- We have defined the concept of a *hybrid probabilistic program* (*hp-program*, for short). If the user selects a set of probabilistic strategies $\rho_1, \ldots, \rho_k$ for use in an hp-program (s/he may select these in any way, as long as these selections satisfy the axioms defining probabilistic strategies), then this automatically defines a set of conjunction and disjunction connectives used to construct formulas from atoms.

- We have defined a fixpoint semantics for hp-programs, a model theoretic semantics for hp-programs, and a proof procedure, and prove that the fixpoint theory, model theory, and proof theory all lead to equivalent characterizations. *This applies to any selection of probabilistic strategies made by the user, as long as these selections satisfy the axioms defining probabilistic strategies.*

- We have defined a cache-based proof procedure that extends the well known tabling techniques in Logic Programming to handle hybrid probabilistic programs. This procedure is also proved to be sound and complete.

- We have studied the computational aspects of hp-programs. In particular, we have designed algorithms for computing the minimal model of an hp-program, determining whether an hp-program entails a formula and determining whether an hp-program is consistent. We have shown that entailment problem in general is *NP-complete*, however, when we restrict the heads of rules to be of size 2 or less, this problem becomes equivalent to a generalization of a weighted matching problem on graphs. The computational complexity in this case depends on the set of probabilistic strategies used in the program. We have established that for a set of eight most common probabilistic strategies (disjunctive and conjunctive strategies of independence, positive correlation, negative correlation and ignorance), the entailment problem is solvable in *polynomial time*. It is also solvable in *polynomial time* for programs having *atomic heads* for any set of probabilistic strategies

- We have also provided a sound and complete rule-based inference system for hp-programs. We showed that for all formulas derivable in this system there exists a "short" (polynomially bound) derivation.

  Though we have not implemented it, it should be mentioned that since our original results have been published, two implementations of Hybrid Probabilistic Logic programs appeared. One implementation, by Terrence Swift at University of Maryland uses the XSB system (developed at SUNY at Stony Brook) to implement hp-programs with atomic heads. This implementation has been used to solve some data mining problems. A somewhat different fragment of HPPs had been implemented by Stoffel at the University of Neuchatel in Switzerland.

Previous Research

During my undergraduate study, my research had been in the area of finite model theory and Linear Logic. I have studied *Resource Transformation Nets (RT-nets)* – a model of computation in which a number of resources can be stored, and transformed at the nodes of a graph and transferred from node to node. I showed that a logic based on the fragment of Multiplicative Linear Logic describes exactly the behavior of RT-nets.

Future Research

My future research will proceed in the following areas:

- *N-Dimensional Probabilistic Reasoning.* I am planning to extend our results for Probabilistic Temporal Databases and Temporal Probabilistic Logic Programs to the case when the probability is distributed over an N-dimensional (and possibly continuous) set, such 2D or 3D space, or 2D/3D space with time. This work will require the generalization of many notions, used in the TP-databases and TP-programs. Database design will also involve work with different spatial data structures such as quadtrees, R-trees etc...

- *Intelligent Agents* I propose to continue working on the theory and implementation of heterogeneous software agents. I intend to study how an agent can make decisions when it is reasoning about uncertain temporal events. For instance, an agent may wish to predict certain events in the future. While such predictions are (inevitably) likely to be uncertain, the agent may still base its actions and schedules on such predictions. I propose to develop methods by which an agent can represent information about multiple possible uncertain future works, and can make decisions based on such information based on one or more objective functions the agent has.

- *Application of probabilistic methods.*

  - *Information Retrieval* (IR) studies the problem of matching *the documents* stored in the database with the *document specifications contained in a given query*. A number of aspects of IR (document specifications, user query, document relevance) are of a highly uncertain nature. Probabilistic methods, together with other formalisms for uncertain reasoning, such as Dempster-Schafer belief functions and Fuzzy Logic have been applied to IR. However, in most cases, just as in Probabilistic Logic Programming, probabilistic methods make an apiori assumption about the relationship between different document descriptors. I would like to extend our framework of *hybrid* probabilistic reasoning onto the domain of Information Retrieval.
  - In *Data Mining*, a problem of finding different data dependencies in a large body of data is studied. The data dependencies found in the data mining process may be interpreted probabilistically. In this area, I see an opportunity to combine our probabilistic approach to reasoning with the approaches to dealing with objective functions, which can be used to specify different data parameters (such as the concept of an "interesting" data dependency) in the data mining framework.

- *Probabilistic Updates.* Given a body of probabilistic data (e.g., probabilistic database, probabilistic deductive database), and a collection of "newly discovered" probabilistic information, the question of updating the data with the new information arises. While working on Probabilistic Temporal Databases and Hybrid Probabilistic Programs, we have found that (a) there is no unique "correct" way to update the probabilistic information, (b) probabilistic updates are not *local*, i.e., updating a probability of one event, may result in a chain of updates of probabilities of other events in some cases and (c) may require application of non-probabilistic methods to preserve consistency of the data. My goal is to study the general properties of probabilistic updates in detail.