Can Probabilistic Databases Help Elect Qualified Officials?

Judy Goldsmith, Alex Dekhtyar and Wenzhong Zhao
Department of Computer Science
University of Kentucky,
Lexington, KY 40506-0046
{goldsmit, dekhtyar, wzhao0}@cs.uky.edu

Abstract

We present a flexible framework for implementing reasoning with uncertainty: Semistructured Probabilistic Databases. This framework bridges the gap between the process of obtaining probabilistic information from data and its use in AI applications by providing the facilities to store and query diverse and complex probabilistic data.

Introduction

The election commissioner of Anytown is facing the following problem: The Rhinoceros Party candidate has just won the election for the State Senate seat, despite a complete lack of apparent support in terms of monetary contributions, yard signs, pre-election or exit polls. The ballot initiative to legalize AI conferences, adamantly opposed by the Rhinoceros candidate, won in a landslide. The commissioner suspects voting irregularities. How can she argue her case in court[1]?

As we know, courts can be intimidated by mathematical formalism. Thus, the commissioner must produce mathematical evidence that the outcome of an election is not in keeping with trends and predictors. In order to do so, she must be able to refer to past data and to the probabilistic analysis of both the data and recent polls.

The commissioner's office has access to previous polls, voter registration records, previous votes in the district, votes in other electoral districts and demographics for her district and others. From this data, analyses must be performed to demonstrate the discrepancies between the predicted and the observed results. During this analysis, the statistical and probabilistic information, derived by the analysts, should be stored in a way that facilitates later comparisons. Due to small sample sizes, possible sample biases, etc., the probabilities obtained come with expected errors and are represented as probability intervals (e.g., 42% with a 3% error becomes the interval $[0.39, 0.45]$). The commissioner and her analysts may ask:

- What demographic groups voted Rhinoceros in the past?
- What other districts went Rhinoceros?
- What percentage did the Rhinoceros candidate receive in various polls, and how does that compare to the percentages the winners of previous elections received in corresponding polls?
- What were the joint probability distributions of voters voting for particular candidates in State Senate, State House and local Mayoral races?
- What is the likelihood of voters voting for both the Rhinoceros Senate candidate and the ballot measure to legalize AI conferences?

While finding the answers to these questions from the data is a matter of statistical analysis, once the results are obtained, they need to be preserved for future recall and reuse, as well as for integration with the results of other analyses. This particular part of working with uncertain information is what we concentrate on in this paper. We present the data model and the query language that allow convenient storage and efficient retrieval of complex probabilistic information.

Related Work

Interval probabilities have been a focus of a number of studies in recent years. (Walley 1991) and others (Kyburg Jr. 1998; Biazzo & Gilio 1999; Biazzo *et al.* 2001) use a behavioral semantics for interval probabilities based on *gambles*. (Biazzo & Gilio 1999; Biazzo *et al.* 2001) extend the theory of imprecise probabilities to incorporate logical inference and default reasoning. At the same time, (Weichselberger 1999) gives a possible world semantics for probability intervals. His semantics applies to Kolmogorov-style probability structures based on atomic events. It is reformulated in (Dekhtyar & Goldsmith 2002) to explicitly include joint interval probability distributions of discrete random variables. Interval probability distributions of discrete random variables generate a set of linear constraints on the acceptable probability values for individual instances. It is possible to extend possible-worlds semantics to more complex sets of constraints (Cano & Moral 2000). The terminology adopted here comes from the work on Temporal Probabilistic Databases by (Dekhtyar, Ross, & Subrahmanian 2001) via (Dekhtyar & Goldsmith 2002), the latter paper containing a detailed comparison of the frameworks and terminologies. The possible world semantics for interval probabilities also occurs in the discussion of Bounded Parameter Markov Decision Processes in (Givan, Leach, & Dean 2000).

[1]For the purpose of argument, we will ignore any recent Supreme Court decisions, and will concentrate on the aspects of the problem about which we can reason logically.

The work on probabilistic databases for the most part concentrated on relational representations (Barbara, Garcia-Molina, & Porter 1992; Cavallo & Pittarelli 1987; Lakshmanan *et al.* 1997; Dey & Sarkar 1996), although, more recently, an object-oriented probabilistic database framework have also been proposed (Eiter *et al.* 2001). Interval probabilities were introduced to databases in (Lakshmanan *et al.* 1997) and have been studied in (Eiter *et al.* 2001; Eiter, Lukasiewicz, & Walter 2001). The first semistructured framework for probabilistic databases have been proposed in (Dekhtyar, Goldsmith, & Hawkes 2001). That framework used point probabilities. (Hung, Getoor, & Subrahmanian 2003) proposed a framework for management uncertainty in the structure of XML documents. The work described in this paper incorporates the flexibility of the semistructured probabilistic object model of (Dekhtyar, Goldsmith, & Hawkes 2001) with the power of imprecise probabilities.

Extended Semistructured Probabilistic Objects (ESPOs)

The basic objects represented in the ESPO model are interval probability distributions. However, other data must be stored with them. For instance, for every probability table stored, we must be able to easily find information about its source (such as poll data), or whether it had been derived from other tables. We must record all probabilistic conditions under which the probabilities were obtained.

We propose a new model, Extended Semistructured Probabilistic Objects (ESPOs) for storage and management of interval probability distributions. ESPOs store two types of information: *stochastic*, namely, the random variables and their (joint) probability distribution and *context* — the *non-stochastic* information associated with the distribution.

More formally, let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be a universe of random variables with domains $dom(v_1), \ldots, dom(v_n)$. If $V = \{u_1, \ldots, u_k\} \subseteq \mathcal{V}$ then we write $dom(V)$ for $dom(u_1) \times \ldots \times dom(u_k)$. Let, $\mathcal{R} = \{A_1, \ldots, A_m\}$ be a list of relational attributes with domains $dom(A_i), 1 \leq i \leq m$ which represent context (non-stochastic) variables of the system. Finally, let C[0,1] denote our probability space: the set of all subintervals of $[0, 1]$.

An Extended Semistructured Probabilistic Object (ESPO) S is a tuple $\langle T, V, P, C\rangle$ where (i) $V \subseteq \mathcal{V}$ is a set of participating random variables; (ii) $P : dom(V) \rightarrow$ C[0,1] is a (possibly incomplete) probability table; (iii) $T = \{\langle A, a, W\rangle\}$, $A \in \mathcal{R}$, $a \in dom(A)$ and $W \subseteq V$ is context, (iv) $C = \{(u, x)\}$, $u \in \mathcal{V} - V$, $x \in dom(u)$ is the set of conditionals. A collection of ESPOs is called an ESP-relation, and a collection of ESP-relations forms an ESP-database.

Consider our election example. Voting in each election is represented as a separate random variable, its domain consisting of the set of possible choices. Information about voter groups, such as gender, race, education, party affiliation, as well as information about the source of the probability distribution (poll origin, date, question format) form the list of context variables. There are three concurrent races in Anytown: State Senate, State House and Mayor, plus a ballot initiative to legalize AI conferences. They are represented by random variables senate, house, mayor, legalize. The first three variables will have the domain {Rhino, Elephant, Donkey}. The legalize variable has a binary {yes, no} domain. Combining random and context variables we can store, for example, information about (i) the joint probability distribution of voting for specific Senate and House candidates for married women based on the independent poll conducted 2 weeks prior to the election date, or (ii) information about the probability that a University educated Elephant party member voted in the last election. The ESPO model combines random and context variables, and in addition, allows the following.

S

qNum:	12, {senate}	⟵ *context*
qNum:	17, {legalize}	
date:	October 23, 2002	
gender:	male	

senate	legalize	[*l*, *u*]	⟵ *random vars*
Rhino	yes	[0.04, 0.11]	⟵ *prob. table*
Rhino	no	[0.1, 0.15]	
Donkey	yes	[0.22, 0.27]	
Donkey	no	[0.09, 0.16]	
Elephant	yes	[0.05, 0.13]	
Elephant	no	[0.21, 0.26]	

mayor: Donkey ⟵ *conditional*

Figure 1: A Sample Extended Semistructured Probabilistic Object (left) and its XML representation (right)

• *Specifying conditionals.* Representing conditional probability distributions (such as the distribution of votes for Senate candidates by voters who intend to vote for the ballot initiative) is of utmost importance in the ESPO model. The model allows us to associate with each collection of random and context variables a list of conditions under which the distribution takes place. Thus, ESPOs feature a conditional part. In the example above legalize = yes would be the conditioning information that would distinguish the abovementioned probability distribution from a simple probability distribution of votes for Senate candidates.

• *Associating individual random and context variables.* Some context variables are associated with individual random variables, and this association needs to be preserved in joint distributions. For example with a joint probability distribution of votes for Senate and Mayor based on a poll by the Rhinoceros party, it may make sense to store information about the position of the questions on the survey form. Then we can associate the information that the Senate question was the 12th question of the survey, while the the Mayor question was the 15th by including (qNum:12) and (qNum:15) in the context of the ESPO and associating them with the senate and mayor random variables respectively. The latter is done by specifying the list of associations in the third part of each context triple.

Consider the ESPO in Figure 1. It stores the joint probability distribution of votes for the Senate candidate and the ballot initiative by men who intend to vote for the Donkey mayoral candidate based on a poll conducted on October 23,

2002. The Senate vote and the ballot initiative were, respectively, the 12th and 17th questions asked. The context part of the ESPO includes the date of the poll, the gender of the respondents, and the questions' order. (Whenever no list of associated random variables accompanies a context entry, we assume that the entry is associated with all random variables of the ESPO.) The joint probability distribution of votes for Senate and the ballot initiative is stored in the probability table. Finally, the conditional part contains information about the mayoral preferences of the subjects.

Semantics of Interval Probabilities

Consider the probability space $\mathcal{P} = \mathsf{C[0,1]}$, the set of all subintervals of the interval $[0,1]$, and a universe $\mathcal{V}$ of *discrete random variables* v with *finite* domains $dom(v)$. If $V = \{v_1, \ldots, v_k\} \subseteq \mathcal{V}$, we write $dom(V)$ for $dom(v_1) \times dom(v_2) \times \ldots \times dom(v_k)$.

An *interval probability distribution function (ipdf)* of random variables V is a function $P : dom(V) \rightarrow \mathsf{C[0,1]}$. If $\bar{x} \in dom(V)$, we write $P(\bar{x}) = [l_{\bar{x}}, u_{\bar{x}}]$. A *p-interpretation* over V is a point probability distribution $I : dom(V) \rightarrow [0,1]$ such that $\sum_{\bar{x} \in dom(V)} I(\bar{x}) = 1$. We say that I satisfies P (denoted $I \models P$) iff $(\forall \bar{x} \in dom(V))(l_{\bar{x}} \leq I(\bar{x}) \leq u_{\bar{x}})$. An ipdf is called *consistent* iff there exists a p-interpretation that satisfies it. Two ipdfs P and P' are called *equivalent* iff $\{I | I \models P\} = \{I' | I' \models P'\}$.

Let $P(\bar{x}) = [l, u]$ and let $\alpha \in [l, u]$. We say that α is *reachable* by P at $\bar{x}$ *iff* there exists a p-interpretation $I \models P$ such that $I(\bar{x}) = \alpha$. The reachability property is shown to be continuous (Dekhtyar & Goldsmith 2002), i.e., if $\alpha < \beta$ are both reachable by P at some $\bar{x}$, then so is any $\gamma \in [\alpha, \beta]$. P is called *tight iff* $(\forall \bar{x} \in dom(V))(\forall \alpha \in [l_{\bar{x}}, u_{\bar{x}}])$ α is reachable by P at $\bar{x}$. This notion corresponds to that of "coherence" in (Walley 1991).

Given an ipdf P', its *tight equivalent* is an ipdf P such that (i) P is tight and (ii) $P' \equiv P$. It can be shown that each consistent ipdf has a unique tight equivalent. A tightening operator $\mathcal{T}$ was introduced in (Dekhtyar & Goldsmith 2002), which maps each ipdf onto its tight equivalent:

$$\mathcal{T}(P)(\bar{x}) = \Big[\max(l_{\bar{x}}, 1 - \textstyle\sum_{\bar{x}' \in dom(V)} u_{\bar{x}'} + u_{\bar{x}}), \\ \min(u_{\bar{x}}, 1 - \textstyle\sum_{\bar{x}' \in dom(V)} l_{\bar{x}'} + l_{\bar{x}})\Big].$$

Extended Semistructured Probabilistic Query Algebra (ESP-Algebra)

ESPOs are complex objects that store probabilistic data and associated information in one place. The question of querying data stored in ESPOs is addressed in this section. Because probabilistic data exhibits certain important mathematical properties that need to be accounted for during query operations, standard relational, object or semistructured query languages are not immediately appropriate. Here, we provide a query algebra that specifies the semantics of different atomic query operations on ESPOs. This algebra can then be incorporated into any specific query language with appropriate structure. ESP-Algebra defines five operations on ESPOs: selection(σ), projection(π), conditionalization(μ), cartesian product($\times$) and join ($\rtimes, \ltimes$).

Selection operation finds ESPOs in an ESP-relation that satisfy a particular selection condition. Acceptable selection conditions are boolean combinations of atomic selection conditions for each type of information that can be stored in an ESPO. These types are:
(1)*Simple context:* expressions for checking the values of context variables of the form *var op value*. E.g., gender = male and date $\leq$ 11/01/2002 are valid simple context selection conditions.
(2) *Extended context:* expressions of the form c/V for checking *both* the values of context variables *and* their associations with the ESPO's participating random variables. Here, c is a simple context selection condition and $V \subset \mathcal{V}$ is a set of random variables. E.g. qNum = 12/$\{senate\}$ evaluates to *true* on an ESPO that has context entry qNum: 12 associated with random variable senate.
(3) *Participating random variables:* expressions checking that ESPOs contain particular random variables. The expressions are of the form $v \in V$ where v, is a name of a random variable. E.g., house $\in V$ evaluates to true on an ESPO S if house is in the list of participating random variables of S.
(4) *Conditioning information:* expressions that check for the presence of conditionals in ESPOs. These expressions have the form $u = \{a_1, \ldots, a_k\}$ where u is a random variable and $a_1, \ldots, a_k \in dom(u)$, e.g., house = {Donkey}.
(5) *Probability Table:* expressions that select rows of ESPOs' probability tables based on the values of random variables in them. E.g., legalize = yes.
(6) *Probabilities:* expressions that select rows of ESPOs' probability tables based on the probability values. These expressions have the form of l *op value* (check of lower bound value) or u *op value* (check of upper bound value). Examples are $u \leq 0.4$ and $l > 0.2$.

When an atomic condition c of one of the first four types is valid for some ESPO S, the selection operation $\sigma_c(S)$ returns S. When the atomic conditions are either on probabilities or the probability table, the ESPO returned retains the same context, participating variables and conditional information, but will only include probability table rows that match the selection condition. Figures 2.(a) and 2.(b) show the results of the queries $\sigma_{\mathsf{legalize=yes}}(S)$ and $\sigma_{u<0.16}(S)$, where S is the ESPO from Figure 1.

Projection is the operation of removing variables from an ESPO. It takes as its parameter the list $\mathcal{F}$ of variables to be kept in the output object(s). Projecting out context variables is a straightforward task: the unwanted entries are removed from the context of the resulting ESPO. A more interesting operation is the removal of unwanted random variables. When a random variable is removed from a joint probability distribution, the probability distribution is adjusted to reflect the marginal probability distribution of the remaining random variables. Given an ESPO $S = \langle T, V, P, C \rangle$ and a set $\mathcal{F} \subset V$, the projection $\pi_{\mathcal{F}}(S)$ is the ESPO $S' = \langle T, \mathcal{F}, P', C \rangle$, where P' is the *marginal interval probability distribution of P over the random variables from $\mathcal{F}$* computed as follows: let interval probability distribution $P'' : dom(\mathcal{F}) \longrightarrow \mathsf{C[0,1]}$ be defined as: $P''(x') = [\sum_{\bar{x''} \in dom(V-\mathcal{F})} l_{(\bar{x'}, \bar{x''})}, \min(1, \sum_{\bar{x''} \in dom(V-\mathcal{F})} u_{(\bar{x'}, \bar{x''})})]$.

(a) $\sigma_{\text{legalize=yes}}(S)$

qNum:	12, {senate}
qNum:	17, {legalize}
date:	Oct. 23, 2002
gender:	male

senate	legalize	$[l, u]$
Rhino	yes	[0.04, 0.11]
Donkey	yes	[0.22, 0.27]
Elephant	yes	[0.05, 0.13]

mayor:	Donkey

(a)

(b) $\sigma_{u<0.16}(S)$

qNum:	12, {senate}
qNum:	17, {legalize}
date:	Oct. 23, 2002
gender:	male

senate	legalize	$[l, u]$
Rhino	yes	[0.04, 0.11]
Rhino	no	[0.1, 0.15]
Elephant	yes	[0.05, 0.13]

mayor:	Donkey

(b)

(c) $\pi_{\{\text{senate}\}}(S)$

qNum:	12, {senate}
date:	Oct. 23, 2002
gender:	male

senate	$[l, u]$
Rhino	[0.18, 0.26]
Donkey	[0.35, 0.43]
Elephant	[0.31, 0.39]

mayor:	Donkey

(c)

(d) $\mu_{\text{legalize}=\{\text{yes}\}}(S)$

qNum:	12, {senate}
date:	Oct. 23, 2002
gender:	male

senate	$[l, u]$
Rhino	[0.09, 0.25]
Donkey	[0.48, 0.63]
Elephant	[0.12, 0.3]

mayor:	Donkey
legalize	yes

(d)

Figure 2: Selection (a,b), Projection (c), and Conditionalization (d) operations applied to the ESPO in Figure 1.

Then, $P' = \mathcal{T}(P'')$.

Suppose, our election commissioner is observing the ESPO S from Figure 1 and wants the probability distribution for votes in just the State Senate election. She can issue the query $\pi_{\{\text{senate}\}}(S)$. The result of this query is shown in Figure 2.(c). It is computed as follows: the legalize variable is removed from the ESPO, leading to the removal of context entries associated exclusively with legalize. Some rows of the probability table collapse; the lower and upper probability bounds of collapsed rows are added together to obtain new lower and upper probabilities for the rows in the new probability table (this produces distribution P''). Finally, the tightening operation is applied to assure that only reachable probabilities will be contained in the result. In our example, the tightening operation leads to adjustment of lower probability values for all three rows in the result.

Conditionalization is the operation of conditioning the joint probability distribution. Given an ESPO $S = \langle T, V, P, C\rangle$ and an atomic conditional selection condition $c : v = \{a_1, \ldots, a_k\}$, $v \in V$, this operation, denoted μ_c, replaces the probability table P with the conditional probability distribution of random variables $V - \{v\}$ given c, and adds c to the conditional part of the result.

The computation of conditional probabilities, straightforward for point probability distributions, turns out to be a non-trivial problem when interval probabilities are present. This question had been addressed in general form by a number of people, including Walley (Walley 1991), Jaffray (Jaffray 1992) and Weichselberger (Weichselberger 1999). Dekhtyar and Goldsmith (Dekhtyar & Goldsmith 2002) solve the problem of computing the result of conditionalization of interval probability distributions for the semantics used by ESPO model. We present here the final formula, while referring the reader to (Dekhtyar & Goldsmith 2002) for a more detailed discussion of its derivation. Let $X = \{a_1, \ldots, a_k\}$. Then $\mu_{\{}v = X\}(S) = S' = \ldots T, V - \{v'\}, P', C' \ldots$, where $C' = C \cup \{v = X\}$ and $P' : V - \{v\} \longrightarrow \mathsf{C[0,1]}$ is defined as follows:

$$P'(\bar{y}) = \left[\frac{l[X]_{\bar{y}}}{\min\left(1 - \sum_{x' \notin X} l_{(\bar{y'}, x')}\ ,\ \sum_{\bar{y^*} \neq \bar{y}\ ,\ x \in X} u_{(\bar{y^*}, x)} + l[X]_{\bar{y}}\right)}\ , \frac{u[X]_{\bar{y}}}{\max\left(\sum_{\bar{y^*} \neq \bar{y}\ ,\ x \in X} l_{(\bar{y^*}, x)} + u[X]_{\bar{y}}\ ,\ 1 - \sum_{x' \notin X} u_{(\bar{y'}, x')}\right)} \right],$$

where

$$l[X]_{\bar{y}} = \max\left(\sum_{x \in X} l_{(\bar{y}, x)}\ ;\ 1 - \sum_{\bar{y'} \neq \bar{y}\ \textbf{or}\ x' \notin X} u_{(\bar{y'}, x')}\right)$$

and

$$u[X]_{\bar{y}} = \min\left(1 - \sum_{\bar{y'} \neq \bar{y}\ \textbf{or}\ x' \notin X} l_{(\bar{y'}, x')}\ ;\ \sum_{x \in X} u_{(\bar{y}, x)}\right).$$

Suppose, instead of being interested in the probability distribution of Senate votes for all voters based on ESPO S, the commissioner is interested in finding the probability distribution of votes for Senate candidates for the voters who decided to vote for the ballot initiative to legalize AI conferences. To get the answer to this question, the commissioner issues the query $\mu_{\text{legalize}=\{\text{yes}\}}(S)$ which returns an ESPO containing the conditional probability distribution of random variable senate given that legalize=yes. The result of this query, shown in Figure 2.(d), contains a new entry legalize $= \{$yes$\}$ in its conditional part, and has the conditional probability distribution for the senate variable.

We note, however, that (Jaffray 1992) has shown that conditioning interval probabilities is a dicey matter: the set of point probability distributions represented by $P'(\bar{y})$ *will* contain distributions I' which do not correspond to any I in P. The unfortunate consequence of this is that conditionalizing is not commutative: $P((A|B)|C) \neq P(A|(B|C))$ for many A, B, and C. Thus, a conditionalization operation is included into ESP-Algebra with the caveat that the user must take care in the use of and interpretation of the result.

Cartesian Product ($\times_\alpha$). Cartesian product in the Extended SP-Algebra constructs a joint probability distribution from the input ESPOs. In order for Cartesian product operation to be applicable to a pair of ESPOs $S = \langle T, V, P, C\rangle$ and $S' = \langle T', V', P', C'\rangle$, S and S' must have disjoint sets of participating random variables ($V \cap V' = \emptyset$) and matching conditional parts ($C = C'$), in which case, they are *Cartesian product-compatible*. Finding the joint probability distribution of random variables from V and V' means computing the probability $P''((\bar{a}, \bar{b}))$ for each instance $\bar{a} \in dom(V)$ and $\bar{b} \in dom(V')$, given their respective probabilities $P(\bar{a})$ and $P'(\bar{b})$. This computes the probability of the conjunction of two events. Since the probability of the conjunction depends on the relationship between the events, there is no unique way to compute it. We employ *probabilistic conjunction strategies* (Lakshmanan *et al.* 1997), operations $\otimes_\alpha : \mathsf{C[0,1]} \times \mathsf{C[0,1]} \rightarrow \mathsf{C[0,1]}$, which compute the probability intervals for the conjunction of two events under specific assumptions about their relationships. In particular, under the assumption of independence between the two events (or the two respective sets of random variables), we get $[l,u] \otimes_{ind} [l',u'] = [l \cdot l', u \cdot u']$. When we have no information about the relationship between the events (ignorance assumption), the appropriate probabilistic conjunction strategy is $[l,u] \otimes_{ign} [l',u'] = [\max(0, l+l'-1), \min(u,u')]$.

Given two Cartesian product-compatible ESPOs S and S', the result of their cartesian product under the probabilistic conjunction strategy $\otimes_\alpha$, denoted $S \times_\alpha S'$, is an ESPO $S'' = \langle T'', V'', P'', C'' \rangle$ where (i) $V'' = V \cup V'$; (ii) $C'' = C = C'$; (iii) $P''((\bar{a},\bar{b})) = P(\bar{a}) \otimes_\alpha P'(\bar{b})$; (iv) $T'' = T \cup^* T'$ [2].

Join ($\rtimes_\alpha, \ltimes_\alpha$). Join in the ESP-Algebra is similar to cartesian product in that it computes the joint probability distribution of the input ESPOs. The difference is that join is applicable to the ESPOs that have *common participating random variables*. Let $S = \langle T, V, P, C \rangle$ and $S' = \langle T', V', P', C' \rangle$, and let $V^* = V \cap V' \neq \emptyset$ and $C = C'$. If these conditions are satisfied, we call S and S' *join-compatible*. Consider three value vectors $\bar{x} \in dom(V - V^*)$, $\bar{y} \in dom(V^*)$ and $\bar{z} \in dom(V' - V^*)$. The join of S and S' is the joint probability distribution $P''(\bar{x},\bar{y},\bar{z})$ of V and V', or, more specifically, of $V - V^*$, V^* and $V' - V^*$.

To construct this joint distribution, we recall from probability theory that under assumption α about the relationship between the random variables in V and V', $p(\bar{x},\bar{y},\bar{z}) = p(\bar{x},\bar{y}) \otimes_\alpha p(\bar{z}|\bar{y})$ and, symmetrically, $p(\bar{x},\bar{y},\bar{z}) = p(\bar{x}|\bar{y}) \otimes_\alpha p(\bar{y},\bar{z})$. $p(\bar{x},\bar{y})$ is stored in P, the probability table of S. $p(\bar{z}|\bar{y})$ is the conditional probability that can be found by conditioning $p(\bar{y},\bar{x})$ (stored in P') on $\bar{y}$. The second equality can be exploited in the same manner.

This gives rise to two families of join operations, left join ($\ltimes_\alpha$) and right join ($\rtimes_\alpha$) defined as follows. For $\bar{y} \in dom(V^*)$ let $S_{\bar{y}} = \mu_{V^*=\bar{y}}(S) = \langle T, V - V^*, P_{\bar{y}}, C_{\bar{y}} \rangle$ and $S'_{\bar{y}} = \mu_{V^*=\bar{y}}(S') = \langle T', V' - V^*, P'_{\bar{y}}, C'_{\bar{y}} \rangle$. Then $S \ltimes_\alpha S' = \langle T'', V \cup V', P'', C'' \rangle$, where $C'' = C = C'$ and $P''(\bar{x},\bar{y},\bar{z}) = P_{\bar{y}}(\bar{x}) \otimes_\alpha P'(\bar{y},\bar{z})$ and, $S \rtimes_\alpha S' = \langle T'', V \cup V', P'', C'' \rangle$, where $C'' = C = C'$ and $P''(\bar{x},\bar{y},\bar{z}) = P((\bar{x},\bar{y})) \otimes_\alpha P'_{\bar{y}}(\bar{z})$.

Conclusion

So, can Probabilistic Databases help elect qualified officials? The answer to this question is, of course, no.[3] However, the Semistructured Probabilistic Database framework proposed here *can* help researchers bridge the gap between obtaining statistical data and using that data in AI applications.

[2] $T \cup^* T' = \{(A, a, V^*)|$ (i) $(A,a,V^*) \in T$ and no $(A,a,V_*) \in T'$ or (ii) $(A,a,V^*) \in T'$ and no $(A,a,V_*) \in T$ or (iii) $(A,a,V_1^*) \in T$ and $(A,a,V_2^*) \in T'$ and $V^* = V_1^* \cup V_2^*\}$ In other words, $T \cup^* T'$ takes the union of the contexts in T and T' and merges the associated variables for common context.

[3] In order for qualified officials to be elected, qualified candidates must be willing to run, and they must run good campaigns.

References

Barbara, D.; Garcia-Molina, H.; and Porter, D. 1992. The management of probabilistic data. *IEEE Trans. on Knowledge and Data Engineering* 4:487–502.

Biazzo, V., and Gilio, A. 1999. A generalization of the fundamental theorem of de finetti for imprecise conditional probability assessments. In *Proc. 1st. Intl. Symposium on Imprecise Probabilities and Their Applications*.

Biazzo, V.; Gilio, A.; Lukasiewicz, T.; and Sanfilippo, G. 2001. Probabilistic logic under coherence, model-theoretic probabilistic logic, and default reasoning. In *Proc. ECSQARU'2001, LNAI*, volume 2143, 290–302.

Cano, A., and Moral, S. 2000. Using probability trees to compute marginals with imprecise probabilities. Technical Report DECSAI-00-02-14, Universidad de Granada, Escuela Técnica Superior de Ingenieria Infor mática.

Cavallo, R., and Pittarelli, M. 1987. The theory of probabilistic databases. In *Proc. VLDB'87*, 71–81.

Dekhtyar, A., and Goldsmith, J. 2002. Conditionalization for interval probabilities. In *Proc. Workshop on Conditionals, Information, and Inference*.

Dekhtyar, A.; Goldsmith, J.; and Hawkes, S. 2001. Semistructured probabilistic databases. In *Proc. Statistical and Scientific Database Management Systems*.

Dekhtyar, A.; Ross, R.; and Subrahmanian, V. 2001. Probabilistic temporal databases, i: Algebra. *ACM Transactions on Database Systems* 26(1):41–95.

Dey, D., and Sarkar, S. 1996. A probabilistic relational model and algebra. *ACM Transactions on Database Systems* 21(3):339–369.

Eiter, T.; Lu, J.; Lukasiewicz, T.; and Subrahmanian, V. 2001. Probabilistic object bases. *ACM Transactions on Database Systems* 26(3):313 –343.

Eiter, T.; Lukasiewicz, T.; and Walter, M. 2001. A data model and algebra for probabilistic complex values. *Annals of Mathematics and Artificial Intelligence* 33(2-4):205–252.

Givan, R.; Leach, S.; and Dean, T. 2000. Bounded parameter markov decision processes. *Artificial Intelligence*.

Hung, E.; Getoor, L.; and Subrahmanian, V. 2003. Probabilistic interval XML. In *Proc. International Conference on Database Theory*.

Jaffray, J.-Y. 1992. Bayesian updating and belief functions. *IEEE Transactions on Systems, Man, and Cybernetics* 22(5):1144–1152.

Kyburg Jr., H. E. 1998. Interval-valued probabilities. In de Cooman, G.; Walley, P.; and Cozman, F., eds., *Imprecise Probabilities Project*.

Lakshmanan, V.; Leone, N.; Ross, R.; and Subrahmanian, V. 1997. Probview: A flexible probabilistic database system. *ACM Transactions on Database Systems* 22(3):419–469.

Walley, P. 1991. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall.

Weichselberger, K. 1999. The theory of interval-probability as a unifying concept for uncertainty. In *Proc. 1st International Symp. on Imprecise Probabilities and Their Applications*.