GADT: A Probability Space ADT for Representing and Querying the Physical World*

Anton Faradjian, Johannes Gehrke
Department of Computer Science
Cornell University
{tony,johannes}@cs.cornell.edu

Philippe Bonnet†
Datalogisk Institut
Københavns Universitet
bonnet@diku.dk

Abstract

Large sensor networks are being widely deployed for measurement, detection, and monitoring applications. Many of these applications involve database systems to store and process data from the physical world. This data has inherent measurement uncertainties that are properly represented by continuous probability distribution functions (pdf's). We introduce a new object-relational data type, the Gaussian ADT GADT*, that models physical data as gaussian pdf's, and we show that existing index structures can be used as fast access methods for* GADT *data. We also present a measure-theoretic model of probabilistic data and evaluate* GADT *in its light.*

1 Introduction

Networks of radar, sonar, seismic, and thermal sensors are being deployed widely for measurement, detection, and monitoring applications. These sensor networks will create a flood of observational data of unprecedented scale [EGHK99]. Similarly, enormous quantities of physical data are, and will continue to be, generated by astronomical sky surveys [SKT+00]. A large class of these applications rely on database systems to store, filter, compare and aggregate large volumes of physical data [BS00].

Inherent to data that result from a physical measurement is *uncertainty* regarding the true value of the measured quantity. This uncertainty can properly be described by a *continuous probability distribution function (*$\mathrm{p.d.f.}$*)* over the possible measurement values. For example, consider a temperature sensor in your office that reports an estimate $\widehat{T}$ of the current temperature T; let this estimate be $\widehat{T} = 68°$ Fahrenheit (F). Given this measurement, do we believe that the temperature in your office is exactly $68°$ F? Assuming that the error introduced by the sensor has a gaussian distribution with a known standard deviation of $\sigma°$F, we can compute the probability that the true temperature T lies in the range $[T_1, T_2]$. In the context of a database application, a user should be able to submit a query that retrieves all temperatures whose true values lie in the range $[T1, T2]$ with a given probability p.

Note that we need to manage such uncertainties using probability theory, and not using fuzzy theory. There is no question here about fuzzy set membership or the definition of vague terms such as "tall" or "hot." Since the nature of our problem is fundamentally probabilistic, fuzzy relational models do not apply in our setting. [AR84, KF88, RM88].

In order to manage the uncertainty associated with physical data but at the same time take advantage of features of a modern database system, we need a data model for representing continuous $\mathrm{p.d.f.}$'s such as gaussians. Surprisingly, none of the numerous probabilistic data models described in the literature handles continuous $\mathrm{p.d.f.}$'s—all models deal with discrete $\mathrm{p.d.f.}$'s [CP87, BGMP92, DS96].

In this paper, we develop a data model for continuous $\mathrm{p.d.f.}$'s. Our first contribution is GADT, a concrete abstract datatype (ADT) for representing one-dimensional gaussian distributions. GADT is simple and expressive. We show that GADT is easy to implement as an extension to an existing object-relational DBMS, and we outline how we can access GADT data efficiently using indexing by linear constraint (QBLC) [GRSY97]. As a proof of concept, we have carried out a prototype implementation of GADT in the Cornell Predator ORDBMS [Ses98].

Our second contribution is a study of the theoretical aspects of probability space ADT's. Having started with the datatype GADT, we lift our level of abstraction to a measure-theoretic framework to reason about properties of datatypes that represent continuous as well as discrete probability distribution functions. We introduce probability spaces and events as the basic elements of any probabilistic data type.

*This work was supported by DARPA under contract F30602-99-2-0528, an IBM Faculty Development Award, and by gifts from Microsoft and Intel.

†Work done while at Cornell University.

We show that equality raises an interesting challenge for continuous distributions, and we introduce operations that overcome this challenge. This conceptual study does not only provide a framework for the future development of probabilistic ADT's, but also sheds light on several aspects of our one-dimensional gaussian model. Thus our measure-theoretic framework is not only an abstraction of given instantiations of probabilistic ADTs, it allows us to gain insights into the general functionality and methods that instantiations of probabilistic ADTs should encompass and what their semantics should be. The reader should therefore understand this paper as a trail that starts with a concrete instantiation, climbs to the abstract level, and then returns to the concrete instantiation with some insights from the abstract level.

The milestones along our trail are as follows. Section 2 introduces the gaussian ADT GADT and its methods. Section 3 outlines techniques for query processing using GADT data and queries. Section 4 studies the theoretical aspects of probability space ADTs, and Section 5 discusses the insights that our theoretical framework provides with respect to GADT. We discuss related work in Section 6 and conclude in Section 7.

2 GADT: The Gaussian ADT

In this section, we introduce GADT, the gaussian ADT with which we can represent physical measurements as continuous gaussian $\mathrm{p.d.f.}$'s. We first introduce gaussian $\mathrm{p.d.f.}$'s formally in Section 2.1. Section 2.2 introduces GADT, and Section 2.3 introduces the methods that GADT supports.

2.1 Preliminaries

A gaussian $\mathrm{p.d.f.}$ has the form

$$g_{\mu,\sigma}(x) \stackrel{def}{=} \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \tag{1}$$

The parameters μ and σ are the *mean* and *standard deviation* of the $\mathrm{p.d.f.}$, respectively. The definite integral of $g_{\mu,\sigma}$ is denoted $G_{\mu,\sigma}$ and gives the probability that the true value of x lies in the interval of integration:

$$G_{\mu,\sigma}([a,b]) \stackrel{def}{=} P\{x \in [a,b]\} = \int_a^b g_{\mu,\sigma}(t)\, dt. \tag{2}$$

For $z \geq 0$, we use $\epsilon(z)$ to denote $G_{0,1}([-z,z])$. It is related to the well-known error function $Erf(z)$ [Fel66, Vol. 1, Ch. 7]:

$$\epsilon(z) \stackrel{def}{=} G_{0,1}([-z,z]) = Erf\left(\frac{z}{\sqrt{2}}\right), \quad z \geq 0. \tag{3}$$

The function ϵ has an inverse, ϕ, defined on $[0,1]$ by:

$$\phi(\epsilon(z)) \stackrel{def}{=} z. \tag{4}$$

Both ϵ and ϕ are monotonically increasing.

2.2 The GADT Model

A measurement that is subject to many small and random errors is normally distributed and characterized by a gaussian $\mathrm{p.d.f.}$ A finite number of repetitions of a measurement also results in a normal distribution [Tay82]. We desire a data model that treats gaussians as first-class data values. GADT accomplishes this by defining a *gaussian ADT*: an instance of the ADT corresponds to a gaussian $\mathrm{p.d.f.}$, and, in terms of physical data representation, consists simply of the two real numbers μ and σ. GADT instances are by definition probabilistically independent of each other so that the joint $\mathrm{p.d.f.}$ of two gaussian instances is simply the product of their $\mathrm{p.d.f.}$'s. Statistical dependence between measured quantities can be represented using higher-dimensional gaussians; higher-dimensional gaussians are a topic for future research.

In order to evaluate the probability that a true physical value lies in a given interval, we need an *interval ADT*. The interval ADT represents intervals on the real line; it is ancillary to GADT. Due to space constraints, and for ease of explanation, we do not define the interval ADT formally, and we focus our attention on the case of single intervals, as opposed to unions of disconnected intervals.

We now use a simple denotational semantics to define GADT methods. The semantics make use of the following basic value mappings, which are generalized in Section 4. Given a GADT instance i having mean μ and standard deviation σ, we define the *gaussian mapping* $\mathcal{I}[\![i]\!]$ by

$$\mathcal{I}[\![i]\!] \stackrel{def}{=} G_{\mu,\sigma}. \tag{5}$$

Similarly, given an instance v of the interval ADT representing the real interval $[a,b]$, we define the *interval mapping* $\mathcal{V}[\![v]\!]$ by

$$\mathcal{V}[\![v]\!] \stackrel{def}{=} [a,b]. \tag{6}$$

Finally, given a real instance x representing the real number X, we define the *real mapping* $\mathcal{R}[\![x]\!]$ by

$$\mathcal{R}[\![x]\!] \stackrel{def}{=} X. \tag{7}$$

We use these three instance mappings to define the GADT methods $Prob$, $Diff$, and $Conf$, and we show how these methods can be used to pose queries involving data with continuous $\mathrm{p.d.f.}$'s.

2.3 GADT Methods

2.3.1 Selections with *Prob*

Computing the probability that a value lies inside an interval is the most fundamental GADT operation. The *Prob* ADT method provides this feature: It takes as argument an interval instance v and returns the probability that the true value of the measurement represented by a GADT instance i lies in $\mathcal{V}[\![v]\!]$:

$$\mathcal{R}[\![i.Prob(v)]\!] \stackrel{def}{=} \mathcal{I}[\![i]\!](\mathcal{V}[\![v]\!]). \quad (8)$$

Prob is useful for obtaining the likelihood of events. As an example, let R_1 be a relation having the GADT-valued attribute *Temp*, which stores a temperature measurement obtained from a temperature sensor. Using *Prob*, we can pose queries such as: Retrieve from R_1 all tuples whose *Temp* is within 0.5 degrees of 68 degrees with at least 60% probability:

SELECT *
FROM R_1
WHERE $R_1.Temp.Prob([67.5, 68.5]) \geq 0.6$

Another example is as follows: Retrieve from R_1 all tuples whose *Temp* is at least 75 degrees with probability at most 90%:

SELECT *
FROM R_1
WHERE $R_1.Temp.Prob([75, \infty]) \leq 0.9$

2.3.2 Comparisons with *Diff*

Another important operation is to compute the difference between two gaussians [Tay82]. Let i_1, i_2 be two GADT instances representing uncertain scalar quantities x_1 and x_2, respectively, and let $\mathcal{I}[\![i_1]\!] = g_{\mu_1,\sigma_1}(x_1)$ and $\mathcal{I}[\![i_2]\!] = g_{\mu_2,\sigma_2}(x_2)$. Because i_1 and i_2 are probabilistically independent, the p.d.f. of $x_1 - x_2$ is a gaussian g_{μ_-,σ_-}, where

$$\mu_- = \mu_1 - \mu_2, \text{ and} \quad (9)$$

$$\sigma_- = \sqrt{\sigma_1^2 + \sigma_2^2}. \quad (10)$$

We use DIFF to denote the difference between two gaussians:

$$\text{DIFF}(G_{\mu_1,\sigma_1}, G_{\mu_2,\sigma_2}) \stackrel{def}{=} G_{\mu_-,\sigma_-}.$$

Note that DIFF is not symmetric in its arguments. The *Diff* method computes DIFF:

$$\mathcal{I}[\![i_1.Diff(i_2)]\!] \stackrel{def}{=} \text{DIFF}(\mathcal{I}[\![i_1]\!], \mathcal{I}[\![i_2]\!]). \quad (11)$$

When used with *Prob*, *Diff* allows us to compare i_1 and i_2 by computing the probability that $a < x_1 - x_2 < b$:

$$P\{a < x_1 - x_2 < b\} = (i_1.Diff(i_2)).Prob([a, b]). \quad (12)$$

As an example, let R_1 and R_2 be two relations each having the GADT-valued attribute *Temp*, which stores a temperature. Consider the following query: Join R_1 and R_2 on the condition that $R_1.Temp$ is within 0.1 degrees of $R_2.Temp$ with probability at least 75%:

SELECT *
FROM R_1, R_2
WHERE $(R_1.Temp.Diff(R_2.Temp)).$
$Prob([-0.1, 0.1]) \geq 0.75$

2.3.3 Comparisons with *Conf*

In the context of astronomical data, C. Page shows that it is useful to compare gaussians by testing whether their confidence intervals overlap [Pag96]. Page calls this kind of join a "fuzzy join"[1] and recommends that it be implemented in all astronomical DBMS's. GADT provides the method *Conf* to do this. Given a GADT instance i and a probability $p \in [0, 1]$, $i.Conf(p)$ evaluates to the $100p$ % confidence interval. Specifically, if $\mathcal{I}[\![i]\!] = g_{\mu,\sigma}$, then

$$\mathcal{V}[\![i.Conf(p)]\!] \stackrel{def}{=} [\ \mu - \sigma \cdot \phi(p),\ \mu + \sigma \cdot \phi(p)\ ] \quad (13)$$

(recall the definition of ϕ from Equation 4). Let S_1, S_2 be two relations each having the GADT-valued attribute *Pos*, which stores the positions of stars along a certain dimension. Then we can ask the following query: Join S_1, S_2 on the condition that the 30% confidence interval of $S_1.Pos$ intersects the 35% confidence interval of $S_2.Pos$:[2]

SELECT *
FROM S_1, S_2
WHERE $S_1.Pos.Conf(0.3) \cap$
$S_2.Pos.Conf(0.35) \neq \emptyset$

2.4 Implementation

As a proof of concept, we performed a prototype implemention of GADT as an extension to the Cornell Predator object-relational DBMS [SP97]. We defined new abstract data types (ADTs) for gaussians and for intervals. We implemented the *Prob* and *Conf* methods of GADT. The computation of probabilities in *Prob* relies on an approximation of ϵ [FGB01]. An alternative is to rely on a pre-packaged implementation of ϵ, such as those found in Mathematica, Matlab, or the GNU C Compiler. The interval ADT is used to express ranges and the results of calls to the *Conf* method. In order to implement the Page join (in Section 2.3.3) we implemented a simple *Intersect* method that computes the intersection of two intervals. The gaussian and interval ADTs extend Predator's type subsystem;

[1] Arguably a misnomer, since it involves no fuzzy set theory.

[2] The interval ADT is assumed to provide a method to compute intersections of intervals. We use $\cap$ as informal notation for that method.

they do not rely on any features particular to this system and thus can be implemented in any ORDBMS.

3 Indexing GADT Relations

When dealing with large volumes of GADT data, queries cannot be efficiently processed by naively scanning relations; we need efficient access methods. Fast access to GADT data can be achieved by translating GADT queries into *queries by linear constraints (QBLC)*. Goldstein et al. [GRSY97] and Agarwal et al. [AAE98] have recently shown that QBLC can be processed efficiently using standard indexing structures such as the R-tree.

Let i be a GADT instance with $\mathcal{I}[\![i]\!] = G_{\mu,\sigma}$. Then i is logically equivalent to the pair (μ, σ), and any condition imposed on i is equivalent to a constraint on (μ, σ). If the condition is given by a boolean predicate b then we can visualize all instances satisfying b as a region B in the $\mu - \sigma$ plane (a subset of $\mathsf{R} \times \mathsf{R}^+$). We call B the *valid region of* b. We call any superset of B a *safe region for* b. A simple procedure for GADT query processing is the following two-stage process:

1. Compute a safe region that is expressible as a set of linear constraints; then
2. Use the constraints as input to a QBLC indexing engine such as the R-tree variant of Goldstein et al. [GRSY97].

Examples of safe region computation follow in the remainder of this section. More general questions of query processing and optimization for GADT are beyond the scope of this paper and await future research.

3.1 Safe regions for $Prob$

Here we show how to process efficiently a selection on the predicate

$$R.a.Prob(I) \geq p, \tag{14}$$

where I is the interval $I = [L, R]$. The adaptation of the procedure to other kinds of predicates is straightforward. Recall from Section 2 that ϵ and ϕ are related to the well-known error function.

3.1.1 Semi-infinite intervals

Suppose first that I is a semi-infinite interval. Without loss of generality, say $L = 0$ and $R = \infty$. We distinguish two cases: $p \geq 0.5$ and $p < 0.5$.

Case 1: $p \geq 0.5$. By the symmetry of gaussians, we must have $\mu \geq 0$. Then

$$G_{\mu,\sigma}([0, \infty]) = \frac{1}{2}(1 + \epsilon(\mu/\sigma)),$$

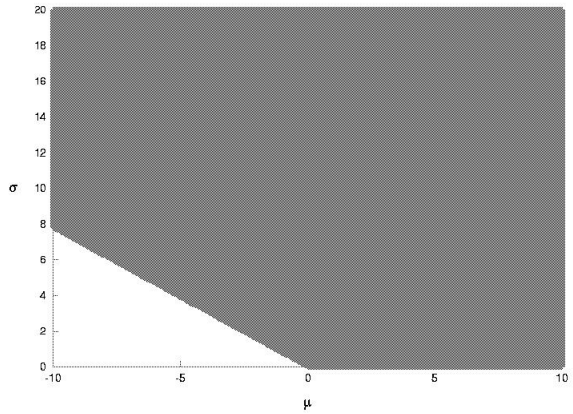

Figure 1. Valid region for $I = [0, \infty], p \geq 0.1$.

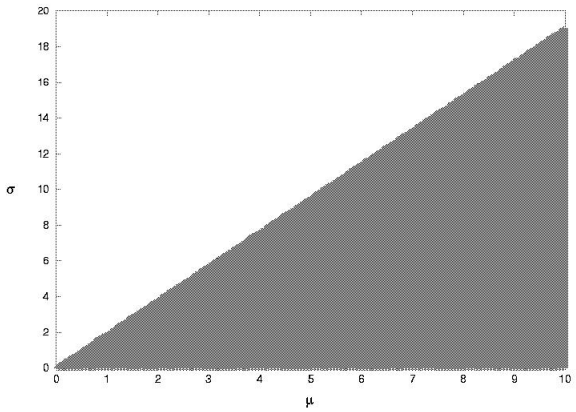

Figure 2. Valid region for $I = [0, \infty], p \geq 0.7$.

so Inequality 14 is equivalent to

$$2p \leq 1 + \epsilon(\mu/\sigma).$$

Since $\phi(x)$ is a monotonically increasing function of x, this last inequality is equivalent to

$$\mu \geq \sigma \cdot \phi(2p - 1). \tag{15}$$

Case 2: $p < 0.5$. Reasoning similarly to Case 1, we obtain

$$\mu \geq -\sigma \cdot \phi(1 - 2p). \tag{16}$$

Inequalities 15 and 16 are linear constraints that define exactly the valid region for the atomic predicate of Inequality 14 for the cases $p \geq 0.5$ and $p < 0.5$, respectively. Examples are shown in Figures 1 and 2.

3.1.2 Finite intervals

Suppose, without loss of generality, that $I = [-w, w]$ for some $w \geq 0$. We can again distinguish the two cases $p \geq 0.5$ and $p < 0.5$. It turns out that they too give rise to qualitatively different valid regions. But for finite intervals, the valid regions are not given by linear constraints. Consider, for example, the case $p \geq 0.5$. In order for $G_{\mu,\sigma}$ to lie in the valid region, μ must lie in the open interval $(-w, w)$. Inequality 14 is then equivalent to

$$\epsilon\left(\frac{|\mu| + w}{\sigma}\right) + \epsilon\left(\frac{|\mu| - w}{\sigma}\right) \geq 2p.$$

Because $\phi(x)$ is a nonlinear function of x, we cannot obtain a linear constraint involving μ and σ by applying ϕ to both sides of the last inequality, as we did above. Figures 3 and 4, which show plots of valid regions for $p = 0.1$ and $p = 0.7$, illustrate that the valid regions are indeed nonlinear. Observe, however, that for any p the valid region is enclosed by a *bounding box* given by the four linear constraints

$$\mu \geq -\mu^\star, \quad \mu \leq \mu^\star, \quad \sigma \geq 0, \quad \text{and} \quad \sigma \leq \sigma^\star. \tag{17}$$

The parameters $\mu^\star$ and $\sigma^\star$ are functions of p. We obtain $\sigma^\star$ by noting that, for fixed σ, $G_{\mu,\sigma}([-w, w])$ is maximum at $\mu = 0$. This follows from the symmetry of $g_{\mu,\sigma}$, and is illustrated in Figures 3 and 4. The parameter $\sigma^\star$ is therefore defined by the equation

$$G_{0,\sigma^\star}([-w, w]) \;=\; \epsilon(w/\sigma^\star) \;=\; p, \quad p \in [0, 1],$$

which is equivalent to

$$\sigma^\star = \frac{w}{\phi(p)}. \tag{18}$$

As for $\mu^\star$, we distinguish the two cases $p \geq 0.5$ and $p < 0.5$. When $p \geq 0.5$, $\mu^\star$ is easily seen to be w itself. When $p < 0.5$, however, the situation becomes more interesting. Consider, without loss of generality, a gaussian $G_{\mu,\sigma}$ with $\mu > w$. In the limit both of very large *and* of very small σ, we have $G_{\mu,\sigma}([-w, w]) = 0$. There is therefore a unique value of σ, which we denote σ_{max}, and which is a function of μ and w, that maximizes $G_{\mu,\sigma}([-w, w])$. Formally, $\sigma_{max}$is defined by

$$\left[\frac{\partial}{\partial\sigma} G_{\mu,\sigma}([-w, w])\right]_{\sigma=\sigma_{max}} \;=\; 0\,, \quad (\mu > w),$$

whose solution is

$$\sigma_{max} = \sqrt{\frac{2\mu w}{\ln\left(\frac{\mu+w}{\mu-w}\right)}}. \tag{19}$$

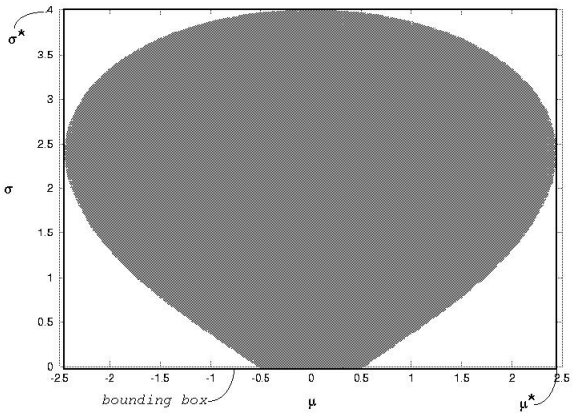

Figure 3. Valid region for $I = [-0.5, 0.5], p \geq 0.1$**.**

The maximum of $G_{\mu,\sigma}([-w, w])$, a function of μ and w, is denoted $G^w_{max}(\mu)$:

$$\begin{aligned} G^w_{max}(\mu) &\stackrel{def}{=} G_{\mu,\sigma_{max}}([-w, w]) \\ &= \frac{1}{2}\left[\epsilon\left(\frac{\mu + w}{\sigma_{max}}\right) - \epsilon\left(\frac{\mu - w}{\sigma_{max}}\right)\right] \\ &= \frac{1}{2}\left[\epsilon\left(\sqrt{\alpha\beta} + \sqrt{\frac{\alpha}{\beta}}\right)\right. \\ &\qquad \left. - \epsilon\left(\sqrt{\alpha\beta} - \sqrt{\frac{\alpha}{\beta}}\right)\right], \end{aligned}$$

where

$$\beta \stackrel{def}{=} \frac{\mu}{w}\,, \quad \alpha \stackrel{def}{=} \ln\left(\sqrt{\frac{\beta+1}{\beta-1}}\right).$$

There are infinitely many values of μ which are so large that no value of σ can satisfy Inequality 14. Such μ satisfy $G^w_{max}(\mu) < p$. But, in the upper halfplane U, there is clearly a unique μ that gives $G^w_{max}(\mu) = p$. This is $\mu^\star$. Formally,

$$G^w_{max}(\mu^\star) = p. \tag{20}$$

To the best of our knowledge, Equation 20 does not admit an analytical (closed-form) solution for $\mu^\star$. The problem can be recast as that of finding the root (zero) of the function $\zeta(\beta) \stackrel{def}{=} G^w_{max}(\mu^\star) - p$. The function ζ is very shallow for large β, however, which suggests that a root-finding algorithm (such as a variant of Newton's method) will struggle to find a good solution if p is small. In other words, the problem is ill-conditioned in that regime. A better solution is simply to tabulate a few values of $G^w_{max}(\mu)$ and to use

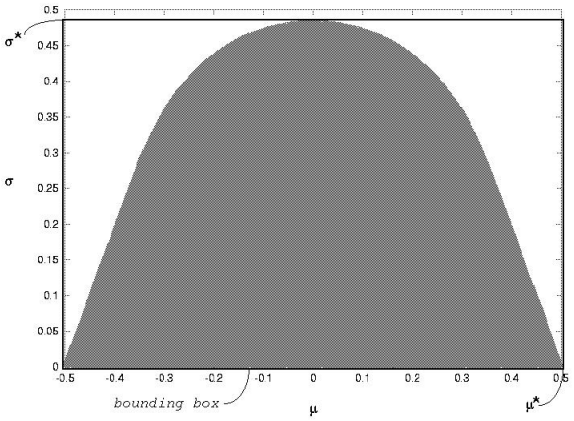

Figure 4. Valid region for $I = [-0.5, 0.5], p \geq 0.7$**.**

the resulting table to look up a conservative estimate of $\mu^\star$. The conservatism introduced decreases with increasing table size, but the table need not be very large; a few extra false positives would only negligibly impair performance. To improve on this scheme, we can interpolate the tabulated values using, for example, a multi-dimensional cubic spline.

3.2 Joins using *Diff*

The following query can be processed using index nested loops (INL) where the index is used to probe the inner relation S:

SELECT *
FROM R, S
WHERE $(R.a.Diff(S.b)).Prob([-w, w]) \geq p$.

Let t and s be tuples of R and S such that $t.a$ is the gaussian g_{μ_a,σ_a} and $s.b$ the gaussian g_{μ_b,σ_b}, respectively. Then $R.a.Diff(S.b)$ corresponds to the gaussian $g_d = g_{\mu_-,\sigma_-}$, from Equations 9 and 10. The results of Section 3.1.2 can thus be used with g_d as the gaussian in question. That is, Equations 17, 18, and 20, apply with the substitutions

$$\mu \to \mu_a - \mu_b, \quad \sigma \to \sqrt{\sigma_a^2 + \sigma_b^2}$$

which implies

$$\sigma_b \leq \sqrt{(\sigma^\star)^2 - \sigma_a^2}.$$

Since σ_a belongs to the outer tuple, it is a constant with respect to the inner index probe, and the last constraint is therefore linear in σ_b.

3.3 Safe regions for *Conf*

The test for confidence overlap also reduces to linear constraints on μ and σ. Let $g_1 = G_{\mu_1,\sigma_1}, g_2 = G_{\mu_2,\sigma_2}$ be two gaussians. Suppose we wish to test whether the c_1 confidence-interval C_1 of g_1 overlaps with the c_2 confidence-interval C_2 of g_2. For convenience, put $C_1 = [L_1, R_1]$ and $C_2 = [L_2, R_2]$. Then it is easy to show that:

$$\begin{aligned} L_1 &= \mu_1 - \phi(c_1)\sigma_1, \;\; R_1 = \mu_1 + \phi(c_1)\sigma_1, \\ L_2 &= \mu_2 - \phi(c_2)\sigma_2, \;\; R_2 = \mu_2 + \phi(c_2)\sigma_2. \end{aligned}$$

The condition for overlap is $L_1 \leq R_2 \;\wedge\; L_2 \leq R_1$, which is equivalent to

$$\mu_1 \leq R_2 + \phi(c_1)\sigma_1 \tag{21}$$
$$\mu_1 \geq L_2 - \phi(c_1)\sigma_1 \tag{22}$$

If we view g_2 as fixed, Equation 21 and Equation 22 are linear constraints involving μ_1 and σ_1, and the valid region is a curtailed cone.

4 Probability Space ADTs

Having presented an ADT for gaussian data, we now begin to explore a more general theory of probabilistic data in the ADT context. The goal is to define a framework (concepts, operations, ADT methods) that is independent of any particular $\mathrm{p.d.f.}$, so that we would not need to undertake a separate study for data whose uncertainty is given by other distributions, for example, a Gamma distribution. The model we present is not only $\mathrm{p.d.f.}$-agnostic, but also subsumes both continuous *and* discrete $\mathrm{p.d.f.}$'s under one general framework. To accomplish this goal, our model uses the language of measure theory.[3] In what follows, we use the term *probability space ADT (PSADT)* to refer to any datatype that aims to model probabilistic data using the ADT approach; the gaussian ADT GADT is an example of such a PSADT.

4.1 Spaces and Events

Let R be a database relation with an attribute a. Before attribute a can be typed as probabilistic and populated with PSADT instances, we must declare the *sample set* Ω in which may lie the true values of the quantities represented by those instances. This is analogous to specifying the domain of a regular attribute. Once Ω is specified, a PSADT instance m can then be modeled as a *probability measure* on the measurable space $(\Omega, \mathcal{F}_\Omega)$, where $\mathcal{F}_\Omega$ is a

[3] We assume the reader is already familiar with the basics of measure theory. See the books by Bartle [Bar95] or Billingsley [Bil95] for an introduction.

suitably chosen σ-algebra of subsets of Ω. We call a measurable set $E \in \mathcal{F}_\Omega$ an *event*, we call the measurable space $(\Omega, \mathcal{F}_\Omega)$ a *sample space*, and we call the triple $(\Omega, \mathcal{F}_\Omega, m)$ a *probability space*. The *domain* of attribute a is thus the set of all measures on the sample space $(\Omega, \mathcal{F}_\Omega)$. The *density* or *Radon-Nikodym derivative* of a probability measure with respect to some underlying measure on $(\Omega, \mathcal{F}_\Omega)$ is called a *probability density function* or *probability distribution function*, and is abbreviated "p.d.f."

As an example, if attribute a is to contain PSADT instances that represent uncertain integer data, then the domain of a is the set of all probability measures on the sample space $(\mathsf{Z}, 2^{\mathsf{Z}})$, where Z denotes the set of integers. In this example, the probability measures corresponding to PSADT instances will be *discrete*: the p.d.f.'s are with respect to the well-known counting measure [Bar95]. GADT, on the other hand, deals with *continuous* probability measures (see Section 5). Both examples fit neatly into the PSADT framework.

We assume that the DBMS supports the abstraction of a set, and we refer to it as the *event ADT*. It is ancillary to the PSADT, and should support basic set operations such as union, intersection, etc. In GADT the event ADT represented intervals over the real line. We need to generalize Equations 5 and 6 to accomodate the probability space abstraction. Given a PSADT instance i representing a measure m on a sample space $(\Omega, \mathcal{F}_\Omega)$, we define the *probability measure mapping* $\mathcal{I}[\![i]\!]$ by

$$\mathcal{I}[\![i]\!] \stackrel{def}{=} m. \tag{23}$$

Given an instance v of the event ADT representing an event $E \in \mathcal{F}_\Omega$, we define the *event mapping* $\mathcal{V}[\![v]\!]$ by

$$\mathcal{V}[\![v]\!] \stackrel{def}{=} E. \tag{24}$$

Unless stated otherwise, we assume throughout this section that the sample space is $(\Omega, \mathcal{F}_\Omega)$. We can now define PSADT methods.

4.2 Event probabilities

The most fundamental operation we can perform with probability spaces is to evaluate the probability assigned by a measure to an event. Let i be a PSADT instance with $\mathcal{I}[\![i]\!] = m$. Let v be an event instance with $\mathcal{V}[\![v]\!] = E$. Then the probability of E under m is given by

$$P\{E\} = m(E).$$

Accordingly, the most basic method of a PSADT is $Prob$, which takes an event instance as argument and computes its probability under a PSADT instance. Formally,

$$\mathcal{R}[\![i.Prob(v)]\!] \stackrel{def}{=} \mathcal{I}[\![i]\!](\mathcal{V}[\![v]\!]), \quad \mathcal{V}[\![v]\!] \in \mathcal{F}_\Omega. \tag{25}$$

4.3 Conditional measures

Let i be a PSADT instance with $\mathcal{I}[\![i]\!] = m$, and let v be an event instance with $\mathcal{V}[\![v]\!] = F$ such that $m(F) > 0$. The conditional probability measure m_F gives the conditional probability of an event E, given the event F:

$$m_F(E) \stackrel{def}{=} \frac{m(E \cap F)}{m(F)}, \ \forall E \in \mathcal{F}_\Omega. \tag{26}$$

The conditional measure can be used for updating: if F is new information then m_F represents the updated probability measure. The PSADT method $Cond$ computes conditional measures:

$$\mathcal{I}[\![i.Cond(v)]\!] \stackrel{def}{=} \mathcal{I}[\![i]\!]_{\mathcal{V}[\![v]\!]}. \tag{27}$$

4.4 Marginalization

Let the sample set Ω consist of the (ordered) cross product $\Omega_1 \times \cdots \times \Omega_n$. We can project out certain dimensions, obtaining a *marginal measure*. Let S be a subset of $\{1, \ldots, n\}$. Without loss of generality, let $S = \{1, \ldots, k\}$ with $k \leq n$. Let m be a measure. Then the *projection of* m *on* S is denoted π_m^S and defined as

$$\pi_m^S(E) \stackrel{def}{=} m(\Omega_1 \times \cdots \times \Omega_k \times E), \tag{28}$$

$$\forall E \subset \Omega_{k+1} \times \cdots \times \Omega_n \text{ s.t. } \Omega_1 \times \cdots \times \Omega_k \times E \in \mathcal{F}_\Omega.$$

The PSADT method $Proj$ computes projections: given a PSADT instance i, we have[4]

$$\mathcal{I}[\![i.Proj(S)]\!] \stackrel{def}{=} \pi^S_{\mathcal{I}[\![i]\!]}. \tag{29}$$

4.5 Comparisons

PSADT must provide a way to compare instances. Such comparisons would be at the heart of natural joins, for example. It turns out that, if we desire a model that treats discrete and continuous measures on the same footing, then the most basic and familiar kind of comparison, equality, needs to be reexamined.

4.5.1 Similarity: A Generalization of Equality

Suppose x_1, x_2 are two uncertain quantities that are known to lie in Ω. Let $S_2 = (\Omega^2, \sigma^2(\mathcal{F}_\Omega))$ be the product sample space, where $\sigma^2(\mathcal{F}_\Omega) \stackrel{def}{=} \{E_1 \times E_2 \mid E_1, E_2 \in \mathcal{F}_\Omega\}$, and let m be a joint probability measure on S_2 for x_1 and x_2. The following discussion applies in either of the following two situations:

[4] For simplicity, we do without a "set mapping" that maps an instance of a set to the set it represents. Hence S appears on both sides of Equation 29.

- There are PSADT instances i_1, i_2 such that m is the product measure $m = \mathcal{I}[\![i_1]\!] \times \mathcal{I}[\![i_2]\!]$.[5]
- There is a single instance i such that $m = \mathcal{I}[\![i]\!]$. (This allows for the possibility of non-factorizable product measures, i.e., attributes that are not probabilistically independent.)

We wish to compute the probability $P\{x_1 = x_2\}$ that x_1 and x_2 are equal. We might proceed as follows. Let $\mathcal{Q} \subseteq \Omega^2$ be the equality relation

$$\mathcal{Q} \stackrel{def}{=} \{(x_1, x_2) \in \Omega^2 \mid x_1 = x_2\}$$

(and assume that $\mathcal{Q} \in \sigma^2(\mathcal{F}_\Omega)$). Then the probability that the values of x_1 and x_2 are equal is simply

$$P\{x_1 \, \mathcal{Q} \, x_2\} \stackrel{def}{=} P\{(x_1, x_2) \in \mathcal{Q}\} = m(\mathcal{Q}). \quad (30)$$

The problem with this approach is that it works when m is discrete but not when it is continuous: in the latter case $m(\mathcal{Q})$ is generally zero. This is just a multi-dimensional analogue of the familiar fact that, given a continuous $\mathrm{p.d.f.}$ on R, there is zero probability that the true value equals any one real number.

The natural solution is to generalize the notion of equality, by replacing $\mathcal{Q}$ with a larger relation $\mathcal{E} \subseteq \Omega^2$. $\mathcal{E}$ is the set of all pairs (x_1, x_2) such that x_1 and x_2 are considered to be *similar* to one another, in whatever sense is appropriate to the application at hand. We require that $\mathcal{E}$ be reflexive, symmetric, measurable (i.e., $\mathcal{E} \in \sigma^2(\mathcal{F}_\Omega)$), and a superset of $\mathcal{Q}$. We do not require that $\mathcal{E}$ be transitive. We call such a relation $\mathcal{E}$ a *similarity event*. It is a relation on Ω and an event in Ω^2. The probability of equality in Equation 30 then becomes a *probability of similarity under* $\mathcal{E}$:

$$P\{x_1 \mathcal{E} x_2\} = m(\mathcal{E}). \quad (31)$$

As an example of a similarity relation, suppose Ω is a metric space with metric d.[6] Let $x \in \Omega$. Then the *neighborhood of radius* r *centered on* x is denoted by $B_r(x)$ and defined by $B_r(x) \stackrel{def}{=} \{y \in \Omega \mid d(x, y) \leq r\}$. Let $\Delta : \Omega \to \mathsf{R}$ be a function, called a *radius function*. Let

$$\mathcal{E}_\Delta \stackrel{def}{=} \{(x, y) \in \Omega^2 \mid y \in B_{\Delta(x)}(x) \ \vee \ x \in B_{\Delta(y)}(y)\} \quad (32)$$

be a relation on Ω. For reasonable choices of the σ-algebra $\mathcal{F}_\Omega$, $\mathcal{E}_\Delta$ will be a measurable set and thus will be a similarity event. We call it *metric similarity under the radius function* Δ. The simplest radius function is constant:

$$\exists \, \delta \in \mathsf{R}, \ \forall x \in \Omega, \ \Delta(x) = \delta. \quad (33)$$

We can obtain a more sophisticated radius function if the metric space Ω is also a *norm space*, that is, a *vector space* with an associated *norm* $\| \cdot \| : \Omega \to \mathsf{R}$. The norm induces a metric $d_{\|\cdot\|}(x, y) \stackrel{def}{=} \|x - y\|, \ \forall x, y \in \Omega$. Now let $\alpha \in \mathsf{R}$ be a small positive real number. Then the radius function $\Delta_\alpha(x) \stackrel{def}{=} \alpha\|x\|$ gives rise to a metric similarity $\mathcal{E}_{\Delta_\alpha}$ (substitute Δ_α for Δ in Equation 32). This similarity event judges two quantities (vectors) to be similar if their difference is small compared to their norms.

Choosing a similarity relation involves a degree of arbitrariness and subjectivity. This is not surprising. A certain amount of subjectivity *should* be involved in deciding whether two real-valued attribute values are equal, even when there is no uncertainty associated with them. Is the difference between 1 and $1 + 10^{-6}$ so significant as to make the numbers unequal? Only the user can decide the answer, and the verdict will depend on the situation at hand. Users therefore rely on similarity relations even when the data are certain. The only additional restriction imposed in the presence of uncertainty is that the similarity relations be measurable.

4.5.2 Total Variation

Probability measures can be compared using the *total variation distance* (TVD). The TVD between the two measures m_1, m_2 is defined as half the measure assigned to the entire sample set Ω by the total variation of their difference [Bar95]:

$$\mathrm{TVD}(m_1, m_2) = \frac{1}{2}|m_1 - m_2|(\Omega).$$

TVD is symmetric in its arguments. It is used to quantify the difference between m_1 and m_2 *as measures*. An example of such a use can be found in Barbará et al. [BGMP92]. In some cases it is natural to interpret a small TVD as indicating a high probability of equality of the underlying data values. This is especially true in the case of physical measurement, where the $\mathrm{p.d.f.}$'s are gaussian and where their expected values are interpreted as best estimates of measurements. The TVD is also appealing because it is a metric and thus renders the vector space of measures a metric space.

The PSADT method TVD computes total variations:

$$\mathcal{R}[\![i.TVD(j)]\!] \stackrel{def}{=} \mathrm{TVD}(\mathcal{I}[\![i]\!], \mathcal{I}[\![j]\!]).$$

4.5.3 Confidence overlap

Probability measures often have canonical choices of *confidence sets*. For example, the interval $[-1, 1]$ is the 68% confidence interval of the gaussian $G_{0,1}$. We use CONF to denote this mapping, as in $\mathrm{CONF}(G_{\mu,\sigma}, 0.68) = [\mu - \sigma, \mu + \sigma]$. The notion of confidence gives rise to the following comparison. Let i_1, i_2 be PSADT instances with $\mathcal{I}[\![i_1]\!] = m_1$

[5] The $\times$ notation refers to the product measure; as always, we assume i_1 and i_2 are probabilistically independent.

[6] Recall that a set Ω is a metric space if there is a function $d : \Omega \to \mathsf{R}$, called a *metric on* Ω, satisfying: $d(x, x) = 0 \ \forall x \in \Omega$; $d(x, y) = d(y, x) \ \forall x, y \in \Omega$; and $d(x, z) \leq d(x, y) + d(y, z) \ \forall x, y, z \in \Omega$.

and $\mathcal{I}[\![i_2]\!] = m_2$. Given $p_1, p_2 \in [0,1]$, we say m_1 and m_2 are *confidence-equal* if their respective confidence sets intersect:

$$\text{CONF}(m_1, p_1) \cap \text{CONF}(m_2, p_2) \neq \emptyset.$$

Informally, for smaller p_1 and p_2, confidence-equality implies that the true values are "closer" to one another. As mentioned above, this is the comparison that underlies the "fuzzy join" proposed by Page [Pag96]. It is useful, for example, when joining astronomical tables based on the positions of stars. We emphasize that CONF is not defined for arbitrary probability measures, but usually only for measures with parametric p.d.f.'s.

The PSADT method $Conf$ computes confidence sets:

$$\mathcal{V}[\![i.Conf(p)]\!] \stackrel{def}{=} \text{CONF}(\mathcal{I}[\![i]\!], \mathcal{R}[\![p]\!]). \tag{34}$$

5 GADT revisited

We now demonstrate that GADT is an instance of the foregoing model: it is a PSADT that represents gaussian p.d.f.'s with respect to the Lebesgue measure λ on the real line. Specifically, if $\mathcal{B}$ is the Borel σ-algebra on R, then the domain of a GADT attribute is the set of all measures on the sample space $(\mathsf{R}, \mathcal{B})$ conforming to the following two restrictions:

1. Only intervallic events are supported. Compare Equation 6 with Equation 24.
2. All p.d.f.'s are with respect to λ and are given by Equation 1.

As Section 2 shows, GADT is useful in spite of restriction 1. That is, we can pose many interesting queries without needing to take unions and intersections of intervals. However, restriction 2 gives rise to at least two complications. The first is the following complication with $Cond$. Given an instance i such that $\mathcal{I}[\![i]\!] = G_{\mu,\sigma}$, and given a conditioning interval v, the instance $j \equiv i.Cond(v)$ would store v in its state, along with μ and σ, and would implement $Prob$ using Equation 26. The problem is that $\mathcal{I}[\![j]\!]$ is no longer gaussian, and, for example, the indexing techniques given in Section 3 would no longer apply. One solution is to use j only as an intermediate result and to forbid its being stored in a relation. This is clearly a disadvantage if we wish to use $Cond$ to update the database. Restriction 2 also gives rise to a second complication that we deal with in Section 5.1.

GADT provides two ways to compare values: $Conf$ and $Diff$. $Conf$ is obviously the same ADT method as that discussed in Section 4.5.3 (compare Equation 13 with Equation 34). $Diff$ can be understood as a change of variables followed by an integration [Tay82]. Referring to Equation 12, if $m = \mathcal{I}[\![(i_1.Diff(i_2))]\!]$, then $m([-\delta, \delta])$ corresponds to the probability assigned by the joint p.d.f. J to the set $\mathcal{E}_\delta = \{(x,y) \in \mathsf{R}^2 : |x-y| \leq \delta\}$. In other words, $\mathcal{E}_\delta$ is a similarity event, and $Diff$ is used to implement metric similarity (Equation 33). For example, the query in Section 2.3.2 uses metric similarity with radius $\delta = 0.1$.

Thus, GADT implements the PSADT notions of event, σ-algebra, probability, comparison, and similarity, and it endows each of these with semantics specific to gaussians. It is an instance of a PSADT.

5.1 On the possibility of arithmetic operations

Having defined $Diff$, a method which computes the difference between two uncertain quantities, the reader may well ask if it is possible to define a method that computes the sum, or the product, or the quotient, or even an arbitrary scalar function of uncertain quantities. This would amount to an arithemtic of uncertain quantities and would provide a way to propagate uncertainties from simple PSADT instances to compound PSADT instances. It would also enable us to manipulate uncertain data as naturally as if they were simple numbers.

There is some hope of achieving such an arithmetic in GADT. To illustrate, let $f : \mathsf{R}^n \to \mathsf{R}$ be an n-ary scalar function, and let $x_1, \ldots, x_n$ be uncertain quantities with gaussian p.d.f.'s $g_{\mu_1,\sigma_1}, \ldots, g_{\mu_n,\sigma_n}$, respectively. It is a fact that, *if σ_k is small compared to μ_k for all $k \in \{1,\ldots,n\}$*, then $f(x_1,\ldots,x_n)$ is normally distributed with p.d.f. g_{μ_f,σ_f} [Tay82], where[7]

$$\mu_f = f(\mu_1, \ldots, \mu_n), \quad \text{and} \tag{35}$$

$$\sigma_f = \sqrt{\sum_{k=1}^{n} \left(\sigma_k \left(\frac{\partial f}{\partial x_k} \right)_{(x_1,\ldots,x_n)=(\mu_1,\ldots,\mu_n)} \right)^2}.$$

When $f(x_1,x_2) = x_1 \pm x_2$, Equation 35 is exact regardless of the size of μ_1/σ_1 and μ_2/σ_2 [Tay82]. But for general f the best we can hope for is accuracy to first-order in μ_k/σ_k. Such an arithmetic extension to GADT is therefore useful only in applications that do not require the exact computation of probabilities. This also suggests that the correct p.d.f. for f is not gaussian, so that GADT boundaries have already been crossed (restriction 2 in Section 5). Such complications are manageable, however, and we see that arithmetic for gaussians is feasible, but only because gaussians have many special properties. Thus, although we could try to generalize our attempts at arithmetic by defining arithmetic methods for non-gaussian p.d.f.'s, there is probably no way to implement them, because neat formulas such as Equation 35 probably do not exist for those p.d.f.'s. In conclusion, there is hope that arithmetic can work with GADT, but only because it is special. Whether arithmetic can also work with a general PSADT is a matter for future research.

[7] These formulae are used in scientific data analysis to propagate measurement error.

6 Related Work

Probabilistic data models (PDM's) have been investigated extensively in the literature, but to the best of our knowledge all of the previous models support only discrete $\mathrm{p.d.f.}$'s. The beginnings of the field can be viewed as extensions of early work on data incompleteness [Lip79, IL84, AKG99].

Wong treats data values as random variables, and regards query processing on uncertain data as a matter of statistical inference [Won82]; his paper has strong connections to the ideas of Lipski [Lip79]. The PDM's of Cavallo and Pittarelli extend each record by a probability stamp such that the sum of all probability stamps over a relation equals one; thus a relation directly encodes a probability distribution [CP87, Pit94].

Lakshmanan and Sadri include probabilities into the rule system of a deductive database through an algebra of confidence-intervals and a probabilistic calculus [LS94]. They also provide results on soundness, completeness, termination, and complexity of their model. Lakshmanan et al. give a probablistic relational model that aims for maximum flexibility by supporting, multiple strategies for combining basic events into complex events [LLRS97].

There is also a considerable body of work on fuzzy relations [AR84, KF88, RM88]. A number of authors have already observed, however, that the fuzzy approach to uncertainty in data is signficantly different from the probabilistic approach [BGMP92, DS96, LLRS97]. Generally, fuzzy logic is not concerned with uncertainty, but with compensating for the lack of expressivity in a language.

The work of Barbará et al. has been particularly influential for us [BGMP92]. Their model represents a discrete probability distribution as a first-class value, in the form of a nested relation. Dey and Sarkar present a model that is a hybrid of the PDM's by Barbará et al. [BGMP92] and Cavallo and Pittarelli[CP87] (see [DS96]). Their relations incorporate probability stamps that are not required to add up to unity.

7 Conclusions

We introduced GADT, a new probabilistic ADT that is especially suitable for representing data in the emerging class of applications that monitor the physical world. Our solution relies on ORDBMS ADT technology and supports continuous $\mathrm{p.d.f.}$'s. We also demonstrated that fast access methods exist for GADT. We believe that GADT is an important step towards general database support for data whose uncertainty is represented by continuous $\mathrm{p.d.f.}$'s. We also presented the general notion of a probability space ADT (PSADT) and showed how GADT conforms to it. The PSADT model is defined in terms of measure-theory and thus encompasses both discrete and continuous $\mathrm{p.d.f.}$'s.

This paper represents our intial work on probabilistic data models, and there are numerous avenues for future work:

- Physical measurements often involve more than one dimension. For instance, most astronomical data are represented as two-dimensional gaussians. We intend to study such multi-dimensional $\mathrm{p.d.f.}$'s.
- Gaussians are not the only relevant $\mathrm{p.d.f.}$ for modelling physical measurements. For instance, heavy-tailed non-gaussian distributions have been introduced to model phenomena with impulsive background noise [Mid99]. For these reasons and those given in Section 5.1, we are interested in the challenge of supporting arbitrary $\mathrm{p.d.f.}$'s.
- Since we are interested in sensor data reduction, we would like to extend the model by introducing aggregate operators.
- We are currently investigating how we can use GADT to represent the results of approximate query answers, where the uncertainty associated with query incompleteness combines with the uncertainty inherent in the measurement data.
- Interesting questions regarding the processing and optimization of general queries on uncertain data await further exploration.
- Since most continuous $\mathrm{p.d.f.}$'s represent real-valued data, it is worth inquiring into the possibility of a general "probabilistic arithmetic."

Acknowledgments. We thank Alin Dobra, Alexandre Evfimievski, Adam Florence, Steve Vavasis, Divesh Srivastava, and Dexter Kozen for helpful discussions.

References

[AAE98] Pankaj K. Agarwal, Lars Arge, and Jeff Erickson. Efficient searching with linear constraints. In *PODS*, pages 169–178, 1998.

[AKG99] Serge Abiteboul, Paris C. Kanellakis, and Gösta Grahne. On the representation and querying of sets of possible worlds. *Theoretical Computer Science*, 78(1):158–187, 1999.

[AR84] M. Anvari and G. F. Rose. Fuzzy relational databases. In J. C. Bezdek, editor, *Proceedings of the 1st International Conference on Fuzzy Information Processing*, pages B–6–3. CRC Press, 1984.

[Bar95] Robert G. Bartle. *The Elements of Integration and Lebesgue Measure*. Wiley, 1995.

[BGMP92] Daniel Barbará, Hector Garcia-Molina, and Daryl Porter. The management of probabilistic data. *TKDE*, 4(5):487–502, 1992.

[Bil95] P. Billingsley. *Probability and Measure*. Wiley, 1995.

[BS00] Philippe Bonnet and Praveen Seshadri. Device database systems. In *ICDE 2000, San Diego, California, USA*, page 194. IEEE Computer Society, 2000.

[CP87] Roger Cavallo and Michael Pittarelli. The theory of probabilistic databases. In Peter M. Stocker, William Kent, and Peter Hammersley, editors, *VLDB 1987, Brighton, England*, pages 71–81. Morgan Kaufmann, 1987.

[DS96] Debabrata Dey and Sumit Sarkar. A probabilistic relational model and algebra. *TODS*, 21(3):339–369, 1996.

[EGHK99] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: scalable coordination in sensor networks. In *Proceedings of the fifth annual ACM/IEEE International Conference on Mobile Computing and Networking August 15 - 19, 1999, Seattle, WA USA*, pages 263–270, 1999.

[Fel66] W. Feller. *An Introduction to Probability Theory and its Applications*. Wiley, 1966.

[FGB01] Anton K. Faradjian, Johannes Gehrke, and Philippe Bonnet. A measure-theoretic probabilistic data model. Technical report, Cornell University, 2001.

[GRSY97] Jonathan Goldstein, Raghu Ramakrishnan, Uri Shaft, and Jie-Bing Yu. Processing queries by linear constraints. In *PODS*, pages 257–267, 1997.

[IL84] Tomasz Imieliński and Witold Lipski Jr. Incomplete information in relational databases. *Journal of the ACM*, 31(4):761–791, October 1984.

[KF88] G. J. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty and Information*. Prentice Hall, New Jersey, 1988.

[Lip79] Witold Lipski Jr. On semantic issues connected with incomplete information databases. *TODS*, 4(3):262–296, 1979.

[LLRS97] Laks V. S. Lakshmanan, Nicola Leone, Robert Ross, and V. S. Subrahmanian. Probview: A flexible probabilistic database system. *TODS*, 22(3):419–469, 1997.

[LS94] Laks V. S. Lakshmanan and Fereidoon Sadri. Probabilistic deductive databases. In Maurice Bruynooghe, editor, *Logic Programming, Proceedings of the 1994 International Symposium, November 13-17, ISBN 0-262-52191-1*, pages 254–268, 1994.

[Mid99] D. Middleton. Non-gaussian noise models in signal processing for telecommunications: New methods and results for class a and class b noise models. *IEEE Transactions on Information Theory*, 45:1129–1149, May 1999.

[Pag96] Clive G. Page. Astronomical tables, 2-d indexing, and fuzzy-joins. In Per Svensson and James C. French, editors, *SSDBM 1996*, pages 44–52. IEEE Computer Society, 1996.

[Pit94] Michael Pittarelli. An algebra for probabilistic databases. *TKDE*, 6(2):293–303, 1994.

[RM88] K. V. S. V. N. Raju and Arun K. Majumdar. Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. *TODS*, 13(2):129–166, 1988.

[Ses98] Praveen Seshadri. Enhanced abstract data types in object-relational databases. *VLDB Journal*, 7(3):130–140, 1998.

[SKT+00] Alexandar Szalay, Peter Z. Kunszt, Ani Thakar, Jim Gray, and Donald R. Slutz. Designing and mining multi-terabyte astronomy archives: The sloan digital sky survey. In Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein, editors, *SIGMOD 2000*, volume 29, pages 451–462. ACM, 2000.

[SP97] Praveen Seshadri and Mark Paskin. Predator: An ordbms with enhanced data types. In Joan Peckham, editor, *SIGMOD 1997*, pages 568–571. ACM Press, 1997.

[Tay82] John R. Taylor. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, 1982.

[Won82] Eugene Wong. A statistical approach to incomplete information in database systems. *TODS*, 7(3):470–488, 1982.