Embracing Uncertainty in Large-Scale Computational Astrophysics

Dan Suciu[1], Andrew Connolly[2], and Bill Howe[1]

[1] Department of Computer Science and Engineering
{suicu,billhowe}@cs.washington.edu
[2] Department of Astronomy
ajc@astro.washington.edu
University of Washington

Abstract. A revolution is underway in astronomy resulting from massive astrophysical surveys providing a panchromatic view of the night sky. The next generation of surveys and the simulations used to calibrate them can produce in two nights what the previous generation produced over many years. This enormous image acquisition capability allows the telescope to revisit areas of the sky with sufficient frequency to expose dynamic features and transient events; e.g., asteroids whose trajectories may intersect Earth. At least three such surveys are planned; their collective output must be integrated and calibrated against computational simulations, prior surveys, and each other.

Relational databases have been shown to be effective for astronomy at yesterday's scale, but new access to the temporal dimension and increased intercomparison of multiple sources generate new sources of uncertainty that must be modeled explicitly in the database. Conventional relational database management systems are not cognizant of this uncertainty, requiring random variables to be prematurely and artificially collapsed prior to manpiulation. Previous results in probabilistic databases focus on discrete attribute values and are unproven at large scale.

In this paper, we present concrete examples of probabilistic query processing from computational astrophysics, and use them to motivate new directions of research: continuous-valued attributes and queries involving complex aggregates over such attributes.

1 Introduction

The last decade has witnessed a revolution in how we approach knowledge discovery in an astrophysical environment. The completion of several massive astrophysical surveys provides a panchromatic view of the night sky; spanning the γ and X-ray spectrum (Fermi and Chandra satellites) through the optical and ultraviolet (the SDSS, GALEX surveys) to the measurements of the cosmic microwave background in the submillimeter and radio (the WMAP and PLANCK satellites). In conjunction with this, simulations of the Universe are becoming larger and more complex—a single simulation today can use as many as a billion resolution elements (Figure 1). While each of these massive data sources, both observational and simulated, provide insights into the highest energy events in our universe as well as the nature of the dark matter and dark energy that drives our accelerating universe, it is only when they are combined, by collating data from several different surveys or matching simulations to observations, that their full scientific potential will finally be realized. The scientific returns from the total will far exceed those from any one individual component.

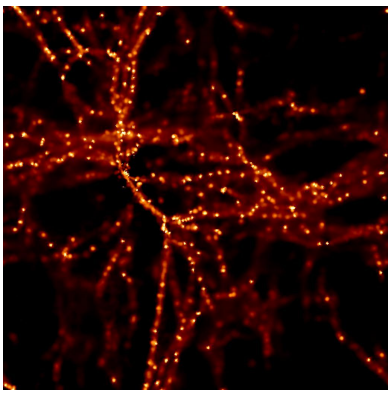


Fig. 1. The neutral hydrogen content of a simulated region of the Universe 25 million light-years wide as it existed 2 billion years after the Big Bang. The areas of highest density (yellow) are sites of galaxy formation.

The recognition of this need to federate massive astrophysical databases has led to initiatives designed to seamlessly interlace data distributed across the globe. Users will soon be able to return multi-frequency attributes of sources identified within regions of the sky without needing to worry about how different surveys or simulations interact or what are the underlying protocols for communicating between them. The virtual observatory (VO) is indicative of the growing awareness in the US and worldwide that efficient organization, distribution, and analysis of scientific data is essential to the continuing evolution of scientific inquiry.

As virtual observatories come online, the rate of data acquisition threatens to overwhelm our ability to access, analyze, and visualize these massive datasets. Further, observations provide only an imperfect, indirect representation of the observed phenomenon—we must build systems that not only tolerate uncertainty, but *embrace* uncertainty by providing probabilistic reasoning capabilities at the system level. Research in database and data management has developed techniques for scalable processing of large data sets, but do not attempt to capture uncertainty. All three modern commercial database systems (IBM's DB2, Oracle, and Microsoft's SQL Server) can optimize and execute SQL queries on parallel database systems, based on research that was done more than a decade ago [5, 14]. New trends in computing, such as the map-reduce programming model introduced by Google [12] and extensions [17, 21, 27] to process massive datasets, have been adopted and expanded as query processing paradigms in systems such as Pig Latin [27], Sawzall [28], Dryad [21], and SCOPE [7]. We are aggressively evaluating these frameworks for general utility in scientific data analysis, but these results are complementary to the development of a theory of scalable query processing over probabilistic science databases, for two reasons. First, the theoretical algorithmic complexity of query answering over probabilistic databases is an independent research question from the design of an effective parallel implementation, map-reduce or otherwise. Second, map-reduce and similar frameworks are increasingly supporting complete relational algebra expressions rather than just simple primitives citeolston:08,isard:07,abouzeid:09, so work on relational modeling and optimization does not preclude a map-reduce implementation.

In principle, relational databases offer the scalability and performance needed to process the data from astrophysical surveys; indeed, the Sloan Digital Sky Sur-

vey (SDSS; http://www.sdss.org), the largest current astronomical database, is powered by a relational database and supports SQL queries. The challenge posed by the new generation of cosmology surveys, however, stems from a significantly larger scale compounded by higher dimensionality (measurements now have a temporal extent) and new sources of uncertainty. The growth in size can be understood if we consider the volume of data generated by the previous generation survey of the SDSS. That ten year experiment surveyed 8,000 sq degrees of the sky and detected $\sim 10^8$ stars and galaxies, forming a 40 TB data set. In contrast, the next decade will see the Dark Energy Survey (http://www.darkenergysurvey.org), PanSTARRS (http://panstarrs.ifa.hawaii.edu/public/home.html) and the Large Synoptic Survey Telescope (LSST; http://www.lsst.org) that will produce **nightly** data rates of $\sim$0.5 TB, 5 TB and 20 TB respectively. The first of these surveys, DES will begin operations in 2011 and cover 5,000 sq degrees over a five year period, PanSTARRS and LSST will cover 20,000 sq degrees **every three nights** and are expected to begin operation in 2014. Beyond their sheer scale, all of these surveys will open the temporal domain through repeated observations of the sky many times over the length of the surveys (up to a thousand times in the case of the LSST). This will enable the extraction of temporal patterns for 10^8 sources with tremendous scientific potential, ranging from detection of moving objects, classification of astrophysical sources, and monitoring of anomalous behaviour. Individual images can no longer be analyzed independently—objects too dim to be recognized in any single image are inferred probabilistically by studying multiple images at different timesteps. However, this inference requires one to reason about and manage multiple possible probabilistic interpretations of the data simulatneously—a capability missing in the formalisms underpinning existing data management software.

In order to extract interesting temporal patterns from the data one needs to characterize the nature of sources from data sets with inherently different noise properties—data may be missing due to incomplete observations and individual sources may drop below the detection threshold of the image. Therefore, we are working to extend and apply recently developed techniques for *probabilistic databases* in order to extract patterns from temporal astrophysical surveys. We model the probabilistic inference associated with the pattern extraction task as a SQL query, then apply techniques such as *safe plans* [10] to execute them in a relational database engine, which will enable the engine to (among other things) evaluate the query in parallel on a cluster.

2 Background and Running Example

A probabilistic database is a relational database where the rows in the database are random variables. In the simplest case, only the value of an attribute is a random variable. Consider the two probabilistic tables in Figure 2. The first table, `Objects` stores the type of each object. The attribute `Type` is a discrete random variable, and its distribution is given explicitly for each object id: thus, for object id `x2234`, `Type` is `Quasar`, `Main Sequence Star`, or `White Dwarf`, with probabilities 0.1, 0.6, and 0.3 respectively, and this is represented by storing three distinct rows in the table, all with the same object id and with the three different values together with their probabilities. The second table in the figure, `Properties`, stores the properties measured periodically (e.g. daily): thus, for each object there are several rows in `Properties`. All these measurements are noisy, and are normally given by continuous random variables (most of them are

Objects:

	OID	Type	P
$t_{1,1}$	x2234	Quasar	$p_{1,1} = 0.1$
$t_{1,2}$	x2234	Main Sequence Star	$p_{1,2} = 0.6$
$t_{1,3}$	x2234	White Dwarf	$p_{1,3} = 0.3$
$t_{2,1}$	x5542	Quasar	$p_{2,1} = 1.0$
$t_{3,1}$	xg413	Main Sequence Star	$p_{3,1} = 0.7$
$t_{3,2}$	xg413	Quasar	$p_{3,2} = 0.3$
$t_{4,1}$	y5553	White Dwarf	$p_{4,1} = 0.1$
$t_{4,2}$	y5553	Binary Star	$p_{4,2} = 0.9$

(a)

Properties:

	OID	Brightness	Color	P
s_1	x2234	19.7	0.31	$q_1 = 0.2$
s_2	x2234	19.7	0.12	$q_2 = 0.8$
s_3	xg413	21.2	0.32	$q_3 = 0.7$
s_4	xg413	19.7	0.24	$q_4 = 0.7$
s_5	x5542	21.2	0.13	$q_5 = 0.5$

(b)

Fig. 2. Example of a probabilistic database. This is a *block-independent-disjoint* database: the 8 tuples (rows) in `Objects` are grouped in four groups. The random variables corresponding to tuples in a group are disjoint, e.g., t_1^1, t_1^2, t_1^3 are disjoint, meaning that at most one can be true; so are t_4^1, t_4^2. Tuples from different blocks are independent, e.g., t_1^2, t_2^2, t_4^1 are independent; the five tuples in `Properties` are independent probabilistic events.

Normal distributions). In the figure, we have represented each row as a discrete event, whose probability indicates a confidence that the row is in the database. However, the natural representation of this data is to use a continuous probability distribution.

Query evaluation on probabilistic databases involves probabilistic inference. Consider, for example, the SQL query in Figure 4 (a), asking for all object types that had a measured brightness < 20. The query joins `Objects` and `Properties` on the object id, and returns the type of the object, provided the brightness is < 20. A probabilistic database needs to examine the *lineage* of each answer, and compute a confidence score for that answer. For example, consider the object type `Quasar`. It is in the answer because of contributions from the first and the third object, and because of three rows in `Properties`, thus, its probability is:

$$p(\texttt{Quasar}) = 1 - (1 - p_{1,1}(1 - (1 - q_1)(1 - q_2)))(1 - p_{3,2}q_4) \qquad \text{(safe result)}$$

The algebra plan in Figure 4(c) computes very efficiently the probabilities of all object types, by incorporating the operations of the formula above into standard relational algebra operations: a join computes the probability $p_1 p_2$ while a projection with duplicate elimination computes the probability $1 - (1 - p_1)(1 - p_2)(1 - p_3) \cdots$ It is possible to modify a relational database engine to compute these probabilities on-the-fly. Alternatively, it is possible to translate back this query plan into SQL (as shown in Figure 4 (d)) and have it evaluated in any standard relational database engine, without having to modify the engine: given the current performance of todays commercial database engines, such a query can be evaluated in a few seconds on a database of hundreds of GB. In our own implementation of a probabilistic database system `mystiq.cs.washington.edu` we took the latter approach, where we translated the relational plan back to SQL.

It is important to note that not every relational algebra plan computes the correct output probabilities. The algebra plan in Figure 4 (b) is equivalent (over standard databases) to that in (c), yet it computes the probabilities incorrectly. In our example it returns the following:

$$p(\texttt{Quasar}) = 1 - (1 - p_{1,1}q_1)(1 - p_{1,1}q_2)(1 - p_{3,2}q_4) \qquad \text{(unsafe result)}$$

The difference is that plan (b) first computes a join, thus making a copy of $p_{1,1}$, and later projects and eliminates duplicates, thus treating the two copies of $p_{1,1}$ as two independent probabilistic events, which is incorrect.

Observations

id	T	X	Y	sigmaX	sigmaY	sigmaXY
a1234	10	2.34	0.46	0.2	0.1	0.3
a1235	10	0.33	3.03	0.1	0.3	0.1
...	...	...				
a5466	11	2.36	0.44	0.2	0.2	0.2
a5467	11	0.33	3.03	0.1	0.3	0.1
...	...	...				

For each observation, the uncertain location of each object (id, T, X, Y, sigmaX, sigmaY, sigmaXY) is given by the two-dimensional Normal distribution $N(\mu, \Sigma)$, where:

$$\mu = \begin{pmatrix} \texttt{X} \\ \texttt{Y} \end{pmatrix} \quad \Sigma = \begin{pmatrix} \texttt{sigmaX} & \texttt{sigmaXY} \\ \texttt{sigmaXY} & \texttt{sigmaY} \end{pmatrix}$$

Fig. 3. The Observations table stores individual observations at each time stamp.

The complexity of query processing in probabilistic databases has been intensively studied [9, 10, 15, 22, 35]. It was proven that, in general, computing the exact output probabilities is a #P-hard problem in the size of the input database [9], due to the interaction of joins and duplicate eliminations. What this means in practical terms is that it is not possible to compute exactly the output probabilities for every SQL query. However, over databases with discrete random variables, certain queries *can* be computed efficiently, and their computation can be expressed as a relational algebra plan, which manipulates the probabilities explicitly: this is illustrated in Figure 4. Such queries are called *safe queries*. Interestingly, not every relational algebra plan computes the output probabilities correctly: plans that compute the probabilities correctly are called *safe plans*. The plan in Figure 4 (b) is unsafe, while the plan in (c) is safe. A survey of the state of the art of the theory of safe queries and safe plans can be found in [10].

3 Two Concrete Problems

We consider two problems in astrophysics to motivate requirements for probabilistic databases. the identification of moving objects, and probabilistic classification.

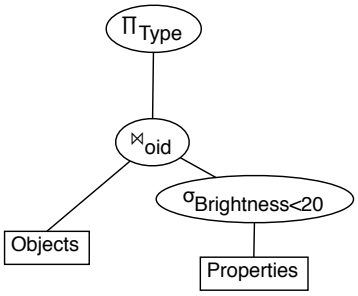
3.1 Tracking Moving Objects

As with the variable luminosities it is the dynamic range of motions of sources across the sky coupled with the confusion due to the many sources that are moving in our own Solar System that drives the complexity of the problem.

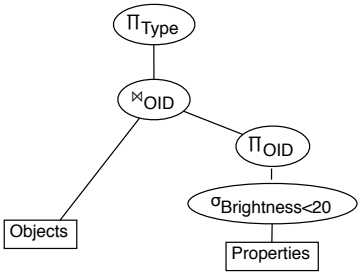
Within the Solar System there are approximately 10^7 sources that move relative to the Earth. The majority of these sources are the Main Belt Asteroids that reside between the orbit of Mars and Jupiter. These can be considered as the background noise that limits our ability to identify the more scientifically

```
SELECT x.Type, confidence( )
FROM Objects x, Properties y
WHERE x.OID = y.OID
 and  y.Brightness < 20
GROUP BY x.Type
```

(a)



(b)



(c)

```
SELECT x.Type, 1-prod(1-x.P*y.P)
FROM Objects x, (SELECT OID, 1-(1-prod(P))
                 FROM Properties
                 WHERE Brightness < 20
                 GROUP BY Type) y
WHERE x.OID = y.OID
GROUP BY x.Type
```

(d)

Fig. 4. A SQL query on the data in Figure 2(a) returning the types of all objects with a brightness below 20. Here `confidence()` is an aggregate operator returning the output probability. The figure shows an unsafe plan in (b) and a safe plan in (c). The safe plan re-written in SQL is shown in (d): the aggregate function `prod` is not supported by most relational engines, and needs to be rewritten in terms of `sum`, logarithms, and exponentiation.

compelling asteroids such as the Kuiper Belt Objects (KBOs) and the potentially hazardous Earth-crossing Near Earth Objects (NEOs).

NEOs with sizes in excess of 1km that strike the Earth have the potential to cause extinction level events through their impact and resulting climate change (similar to the events that may have led to the extinction of the dinosaurs). Such impacts are expected to occur every 500,000 years. On smaller scales (>140m) asteroid impacts would cause substantial damage (equivalent to 100 megatons of TNT) if impacting a populated area (with a 5% chance of impact over the next century). Because of this NASA has a high priority goal of mapping asteroids with sizes > 300m within the next 10 years to assess the potential risk to the Earth from impacts.

The challenge in finding these asteroids comes from the fact that we have multiple observations of the sky every three nights (i.e. we cannot continuously view one region of the sky as the asteroids are distributed over several thousand square degrees). For NEOs we will likely detect 50,000 sources against a background of 10^6 Main Belt Asteroids. Each of these sources moves with an average velocity of one degree per day. Sampling the orbits every three days and accounting for uncertainties in the positional and velocity measurements the combinatorics in connecting subsequent observations are daunting.

More distant asteroids, in particular those beyond the orbit of Neptune, have the potential to explain the origins of our Solar System. Kuiper Belt Objects are in orbits at distances of > 30 Astronomical Units (AU) and have a composition that is identical to the planetesimals that coalesced to form the planets.

Mapping their positions, distances, dynamics and colors (from which their composition can be determined) will constrain rate of accretion, collisions and orbital perturbations that led to the formation of the inner planets as well as providing statistical evidence for the possibility of additional existing and/or vanished planets in the outer Solar System.

There are currently ∼1000 known KBOs which compares to the expected 10-100,000 KBOs from surveys such as the LSST. Moving at 1 arcsecond per hour KBOs will move 2 degrees per year. Simple algorithms that search all of the available phase space for the orbital parameters would be prohibitive in computational time. The joint analysis of one year of data would increase the population of known KBOs by a factor of 50 and our sensitivity to asteroids a factor of 100 smaller in mass. This will enable studies of KBOs at distance in excess of 50 AU where we find a dramatic (and unexplained) drop in the asteroid population.

3.2 Working with Probabilistic Classifications

As described in Section 2, the next generation of astrophysics surveys will open the temporal domain probing a broad range of classes of sources, from the most energetic events in the universe to new classes of physics. Classifications will be derived based on repeated measurement of the same attributes for sources or by "coaddition" of the input images to provide a higher signal-to-noise measures. In each of these cases, the measurements and classifications will be inherently probabilistic. In the simplest case, these classifications will be uni-modal and can be approximated by a Gaussian (e.g. the likelihood of a source being a star or a galaxy). In more complex examples, such as an estimate of the redshift of a galaxy from its colors [8, 20, 25], the probability functions are non-Gaussian and often multimodal.

Understanding how the properties of galaxies depend on their class enables a better understanding of the physical processes that govern the evolution of the universe. Designing new ways of querying databases with probabilistic classifications and uncertain measurements is, therefore, a critical component of any future astrophysical survey. We provide two examples that will guide our development of probabilistic databases. In the initial example we will address the question of how galaxies are related to the dark matter halos in which they reside. Do the properties of galaxies depend simply on the mass of the dark matter halo or are galaxy properties influenced by larger scale structures? Locally we can address these questions using large spectroscopic surveys. We find, for example, that environment plays an important role in determining the properties of galaxies (e.g. their star formation and morphology; [19]. At higher redshifts, we do not have complete spectroscopic samples and, therefore, estimates of redshift and type must be undertaken probabilistically.

How do we use probabilistic classifications to determine and understand the relation between the properties of galaxies and their mass or environment? The classical approach is to subdivide a parent galaxy catalog into a series of subsamples (e.g. assuming categorical classifications based on luminosity or type of a galaxy) and to consider the clustering of these subsamples in isolation (e.g., [18, 26, 39]). This has been successful in characterizing simple relationships such as the morphology-density relation [29] but there are many reasons why this is not an optimal use of the data. The properties of galaxies (luminosity, morphology, star formation) are usually continuous in nature and how we choose to discretize a sample into subgroups is often arbitrary. Treating all galaxies within a subgroup as equal ignores the properties of the galaxies within

that group; we are discarding information. Finally, all of the classifications are inherently noisy so fixed classification thresholds will bias the derived relations.

To address these issues new statistics have been developed, marked correlation functions (MCFs), that do not require that we subdivide a sample of galaxies [36]. In their simplest form, the marked correlation functions, $M(r)$ are essentially, weighted correlation functions such that,

$$M(r) = \frac{\sum_i \sum_j w_i w_j \mathcal{I}(r_{ij} = r)}{\langle w \rangle^2 \sum_i \sum_j \mathcal{I}(r_{ij} = r)} = \frac{1 + W(r)}{1 + \xi(r)}, \tag{1}$$

where $\mathcal{I} = 1$ if the separation r_{ij} between galaxy i and galaxy j is r, and $\mathcal{I} = 0$ otherwise, so that the sum over pairs (i, j) includes only those pairs with separation $r_{ij} = r$. Here w_i is the weight or the mark (e.g. the luminosity or morphology) of galaxy i, $\langle w \rangle = \sum_i w_i / N_{gal}$ is the mean mark, and so $W(r)$ and $\xi(r)$ are the weighted and unweighted two-point correlation functions. Weighting galaxies by different marks yields datasets which are each biased differently relative to each other, and to the underlying dark matter distribution. In principle, we can determine which properties of galaxies result in a weighting of the galaxy distribution which minimizes the bias relative to the dark matter and over what redshift ranges this holds. In the context of a halo model that describes the mass and clustering of dark matter halos, marks provide a probe of the role of mass and environment in determining galaxy properties [37].

Extending these analyses to the clustering of the dark matter we can consider gravitational lensing signatures due to the growth of structure as a function of the age of the universe [3]. Foreground structures induce a lensing signal in background sources. By averaging the ellipticities of galaxies inside circular apertures, the coherent induced shear can be measured and can be used to estimate galaxy and cluster masses, the cosmological mass power spectrum, and higher order statistics. The size of the lensing distortions depends upon both the distances traveled, and upon the growth function which determines the amplitude of the deflecting mass concentrations. Weak lensing is an attractive cosmological probe because the physical effect, gravitational deflection of light, is simple and well understood. Furthermore, the deflecting masses are dominated by dark matter, the evolution of which is purely gravitational and hence calculable. Lensing is currently regarded as one of the most promising probes of the dark energy.

The uncertainties in this case comes from the use of colors to estimate the distances to the lens and background galaxies (i.e. photometric redshifts). As in all inversion problems: the data are both noisy and incomplete. A consequence of this is that photometric redshifts have broad error distributions as well as the presence of multiple minima. The scatter within the relation, its dependence on redshift and galaxy type, and the number of catastrophic outliers will all determine our ability to constrain the equation of state for dark energy. Given the prominent role that photometric redshifts play in current and planned cosmological surveys, it is both timely and necessary that we address how we account for these uncertainties when analyzing massive data sets and how can we, in the context of a database design, minimize the impact of the photometric redshift uncertainties to maximize our ability to constrain dark energy and dark matter [24].

4 Research Challenges

Our first aim is to store the temporal astrophysics data in a cluster of relational databases. Next, we will explore a theory of petascale relational query processing

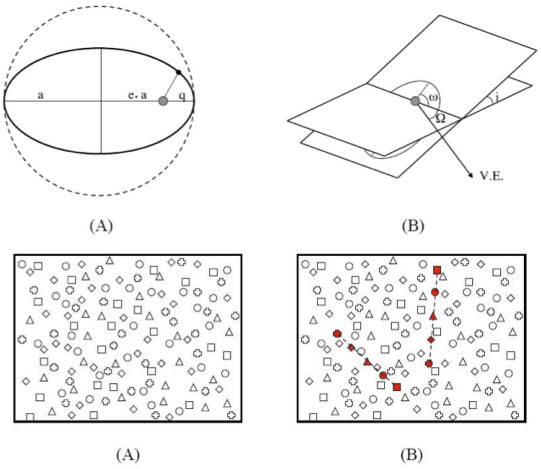


Fig. 5. To determine the orbital parameters of a moving source shown in the top panel requires six positional measures (i.e. three observations each with a Right Ascension and declination). To link three subsequent observations we must match moving sources over a period of three to 90 days. The lower panel shows a sequence of 4 sets of observations superimposed where the tracks of two moving objects have been highlighted in the right panel. the combinatorics associated with a naive linkage model that does a simple linear forward prediction results in a large number of false positives.

by addressing two specific challenges in computational, data-intensive astrophysics: Trajectory-Fitting Queries over Uncertain Paths and Scalable Analyses of Probabilistic Classifications.

In this section, we describe two challenges derived from the concrete scenarios described in Section 3.

4.1 Safe Queries over Continuous Attributes

The theory of safe queries was developed only for probabilistic databases represented by discrete random variables. In contrast, astrophysical data fundamentally requires continuous random variables expressed as probability density functions (pdf). Our second aim is to develop new, fundamental techniques for processing SQL queries over probabilistic databases that have both discrete and continuous random values. A naive solution is to simply replace the the continuous distribution with a discrete one by sampling. However, manipulating closed-form expressions is simpler, more accurate, and far more efficient than manipulating a set of possible worlds, and is therefore preferred when possible. In particular, we will identify and characterize the class of SQL queries that can be evaluated efficiently over probabilistic databases with continuous random variables. In the case of discrete attribute values, there exists a clear separation between *safe queries*, which can be computed efficiently using a safe plan and *unsafe queries*, which are provably #P-hard: we will study whether a similar dichotomy result holds for queries over continuous attribute values. A particularly hard challenge are fuzzy joins, where the join predicate is given by a user defined function that returns a confidence score. An index structure for fuzzy joins have been described for the special case when the function is the Jacquard similarity between two strings [2], with applications to data cleaning; we plan to explore extensions of that technique to user-defined similarity functions needed in the

continuous domain. Another challenge comes from the fact that continuous random variables may or may not be closed under certain algebraic operations. For example, the sum of two Normal distributions is always another Normal distribution, but the sum of a Normal distribution and a uniform distribution is not expressible by a standard pdf. In contrast, the sum of any two discrete numerical random variables is always a discrete random variable.

We approach this challenge by focusing on the first problem mentioned in Section 3, detecting moving objects. The challenge here is to transform a set of uncertain point measurements into a set of object trajectories by fitting curves subject to physical constraints (e.g. that each track must be able to be described by a set of orbital parameters). Each point is generally modeled as a two dimensional Gaussian distribution. Points at two different time stamps can either be the same fixed object, in which case their coordinates should be the same during all time stamps, or can be the same moving object, in which case their coordinates should evolve along a well defined trajectory, or are unrelated objects. By aggregating across many time stamps we expect to identify moving objects with high confidence. We describe this in some detail in the following example.

Example 1. Assuming we have 100 observations (over a period of, say, five months), each with 10^7 moving objects in `Observations(id, T, X, Y, ...)`. Starting with three time slices `T1,T2,T3`, at the beginning, the middle, and the end of the observation period, we will compute triples of observations `(id1,id2,id3)` that are close in space in the three time slices. Over this time period, the known distribution of orbital parameters constrains how far an object may move in a given time period, which will allow us to aggressively prune the set of candidate triples that need to be considered. For example, over an eight day period, an orbit can be approximated by a quadratic in x and y and asteroids are known to rarely move more than 1 degree per day. Moreover, given the endpoints `id1` and `id3`, the trajectory between these endpoints constrains the position in the middle, further reducing the number of candidates `(id1,id2,id3)`. In total, we expect to generate about 10^8 candidate triples, about 10 times more than the total number of objects observed.

Next, for each candidate triple, we will compute an approximate candidate trajectory, which is defined by six parameters $a, b, \ldots, f$ such that the quadratic trajectory is:

$$x = at^2 + bt + c$$
$$y = dt^2 + et + f$$

Furthermore, the errors in the coordinates translate into errors of the parameters, leading to six more parameters. All this information can be computed using a SQL Stored Procedure, and stored in a new relation, `Trajectories(tid, a, b, ...)`: with 10^8 records of (conservatively) 500 bytes each, for a total of 50GB. The attribute `tid` is a unique key.

At this point we need to *validate* the trajectories, by using the other time slices in the `Observation` table. A correct trajectory should predict correctly the position of its object in all time slices $T = 1, 2, 3, \ldots, 100$. A few misses are tolerable, due to measurement errors, but most predictions are expected to be fairly accurate. To do this, we first predict, for each candidate trajectory and each timestamp t, the position of its object at time stamp t. This results in a new probabilistic table, where each predicted position is (approximated by) a Normal distribution, and which has a foreign key to `Trajectories`:

```
CREATE MATERIALIZED VIEW
```

```
      Predictions(new(PID), C.TID, T.t, X, Y, probabilities...) AS
SELECT C.a*T.t*T.t + C.b*T.t + C.c AS X,
       C.d*T.t*T.t + C.e*T.t + C.d AS Y, probability...
FROM Trajectories C, TimeStamps T
```

Here TimeStamps is the active domain of the timestamps: e.g. the set of the 20 timestamps. PID is a unique identifier created for each prediction point.

To validate the trajectories, we compute for each predicted point, the probability that it is actually observed in `Observations`. This is a spatial join between two probabilistic tables, `Predictions` and `Observations`:

```
CREATE MATERIALIZED VIEW PredictionsConfidence(PID, ....) AS
SELECT P.PID,  confidence()  /* here we aggregate probabilities */
FROM Predictions P, Observations O
WHERE P.T = Observations.T AND closeEnough(P.x,P.y,O.x,O.y)
GROUP BY P.PID
```

This is a query with a fuzzy join, defined by the predicate `closeEnough`: we assume that the confidence score computed for the prediction depends on the closeness of the predicted point to the real point. Finally, we join this back with the trajectories, to get a confidence score on the trajectories:

```
CREATE MATERIALIZED VIEW TrajectoreisCOnfidence(TID, ...) AS
SELECT C.TID, confidence()
FROM Trajectories T, PredictionsConfidence P
WHERE T.TID = P.TID
GROUP BY C.TID
```

Here the $\approx$ 100 confidence scores for one trajectory (one per timestamp) are aggregated into a global confidence score for that trajectory (hence the role of the `GROUP BY`): a few misses are tolerable, but many misses will resulg in a low confidence score for that trajectory. Finally, the trajectories are sorted in decreasing order of confidence score and filtered by some threshold.

4.2 Complex Aggregates on Probabilistic Data

A second challenge is to develop general query processing techniques for computing complex aggregates over probabilistic data with continuous values. In SQL, aggregates come in two forms: *value aggregates* that are returned to the user in the `SELECT` clause, like in the query *"`COUNT` the number of galaxies in each region"*; and *predicate aggregates* that appear in the `HAVING` clause, like in the query *"find all regions where the number of galaxies is greater than 20"*. In the case of probabilistic databases, value aggregates are interpreted as expected values. For example if the `type` of an object is a discrete probability distribution with possible outcomes `star`, `quasar`, `galaxy` etc., then counting the number of galaxies in a region results in the expected value of that number given the joint distributions of `type` attributes of all objects in the region. In the case of discrete random attributes, linear aggregate functions such as `COUNT` and `SUM` can be computed straightforwardly, by using the linearity of expectation, but other aggregates, such as `MIN`, `MAX`, `AVG`, are more difficult to compute, and their complexity depends on the structure of the SQL query (e.g. how many joins there are); the case of `AVG` is particularly difficult, even for queries without joins [23]. Predicate aggregates, on the other hand, are interpreted as a probability representing a confidence score: for each region the system computes the probability that the number of objects of type `galaxy` is greater than 20. To compute this

confidence score one generally has to compute the entire probability density function of the aggregate value. Some techniques for predicate aggregate have been developed for probabilistic databases with discrete random variables [32].

This challenge requires new techniques to extend value aggregates with `MIN`, `MAX` to queries with joins, and to extend both value and predicate aggregates to continuous attributes. To explore the challenge further, consider the following two examples in astrophysics: clustering with intrinsic uncertainty and gravitational lensing analysis.

Example 2. As described in section 3.2, galaxies are classified by clustering on more than 20 measurable attributes. Specifically, the `type` of an observed object is a function of fluxes, wavelength, morphology, moments, and more. These attributes are collected repeatedly by the sky surveys and stored in the `Observation` table (Figure 3). Each measurement is inherently uncertain and may be measured thousands of times over the course of the survey. Consequently, these values are represented as a normal distribution (i.e., the mean and variance). To determine object type, one may learn a partition function based on training set of manually labeled data, converting a set of continuous random variable measurements into a single discrete random variable.

The uncertain type of the objects can help answer a variety of astrophysical questions. For example, we can reason probabilistically about objects that change type from one time step to the next or disappear completely; previously these cases would be handled as anomalies. Returning to examples of complex aggregates, we can find regions in the sky with a concentration of galaxies above a specified threshold but with a bound on the minimum luminosity:

```
SELECT Region.id, COUNT(*), MIN(o.luminosity) FROM  Object o, Region r
WHERE Object.type = 'galaxy' and  inRegion(Object, Region)
GROUP BY Region HAVING COUNT(*) > $c AND MIN(o.luminosity) < $l
```

Here `Region` is a collection of regions in the sky (defined by the two diagonally opposed points) that form a partition of the sky. `inRegion` is a user defined function checking if an object is in the given region: it is a deterministic predicate, hence it defines a spatial join, but not a fuzzy join. Luminosity and type are both uncertain values, complicating the semantics and evaluation of this query.

Example 3. As a second application, we plan to use aggregate queries for scalable model-fitting to study the evolution of the growth of structure in the universe through gravitational lensing. The uncertainty in classification in this case comes in two ways. The distances to galaxies (lenses and the lenses themselves) are derived based on the colors of galaxies. These distance estimates (photometric redshifts) are inherent uncertain. While the uncertainties are often Gaussian they can also have multimodal probability density functions and complex forms. The second classification uncertainty is the label for the measured ellipticity (due to the gravitational shear). These measures, for inherently low signal-to-noise galaxies, which are barely resolved relative to the telescope point-spread-function, must be averaged over a large number of galaxies to provide a statistically significant measure of lensing (e.g. by aggregating regions on the sky). To accomplish this we will cluster groups of galaxies into shells at various distances, and then compute for each shell and each region in the sky the average shape distortion of the galaxies in that shell and that region. By comparing this average distortion to one predicted by a random model, we can perform the gravitational analysis (i.e. a shear correlation function):

```
SELECT Shell.id, Region.id, avg(Object.distortion)
FROM Object, Shell, Region
```

```
WHERE Object.type='galaxy'
AND  inRegion(Object, Region) AND inShell(Object, Shell)
GROUP BY Shell.id, Region.id
```

There are two forms of uncertainty that must be handled. First, the redshift shell to which a galaxy belongs will be a discrete random variable rather than a fixed shell. Second, the distortion will be given by a continuous, multimodal random variable. Thus, the average aggregate operator needs to handle both continuous and discrete random variables in its input.

This challenge requires an effective representation of the probability density function (pdf) of the aggregate value for various patterns of aggregate operators and query structures. Representations for some patterns are known: For the `COUNT` over safe queries the pdf can be computed by a safe plan [32], but for `SUM` queries this is not possible even for queries without joins. For `SUM`, one approach is to examine lossy representations of the pdf, in terms of moments, which can be computed effectively for `SUM`. Effective representations for other query patterns are considered open problems.

5 Related Work

Probabilistic databases have been studied intensively in recent years [1, 4, 10, 22, 35], motivated by a wide range of applications that need to manage large, imprecise data sets. The reasons for imprecision in data are as diverse as the applications themselves: in sensor and RFID data, imprecision is due to measurement errors [13, 34]; in information extraction, imprecision comes from the inherent ambiguity in natural-language text [16]; and in business intelligence, imprecision is tolerated because of the high cost of data cleaning [6]. In some applications, such as privacy, it is a requirement that the data be less precise. For example, imprecision is purposely inserted to hide sensitive attributes of individuals so that the data may be published [30]. Imprecise data has no place in traditional, precise database applications like payroll and inventory, and so, current database management systems are not prepared to deal with it. In contrast, in a *probabilistic database management system*, is a system that can store probabilistic data and supports SQL queries over this data. The major challenge studied in probabilistic databases is the integration of query processing with probabilistic inference. A number of techniques have been described recently: lineage-based representations [4], safe plans [11], algorithms for top-k queries [31, 38], and representations of views over probabilistic data [33].

6 Conclusions

We have described concrete problems for probabilistic databases arising from a new generation of massive sky surveys and massive astrophysical simulations. To address these problems, we recommend extensions to existing probablistic theories to accommodate 1) continuous random variable attributes in the context of *safe plans*, 2) scalable evaluation strategies for complex aggregates over continuous attributes, 3) scalable implementations over parallel databases clusters. In general, we advocate exploration of domain science as a driver for applications and requirements for probabilistic databases, and we offer this initial treatment in Astronomy as an exemplar.

Acknowledgements

This work was partially funded by NSF IIS-0713576 and the eScience Institute at the University of Washington.

References

[1] L. Antova, T. Jansen, C. Koch, and D. Olteanu. Fast and simple relational processing of uncertain data. In *ICDE*, 2008.

[2] A. Arasu, V. Ganti, and R. Kaushik. Efficient exact set-similarity joins. In *VLDB*, pages 918–929, 2006.

[3] M. Bartelmann, L. J. King, and P. Schneider. Weak-lensing halo numbers and dark-matter profiles. A&A, 378:361–369, Nov. 2001.

[4] O. Benjelloun, A. D. Sarma, A. Halevy, M. Theobald, and J. Widom. Databases with uncertainty and lineage. *VLDBJ*, 17(2):243–264, 2008.

[5] D. Bitton, H. Boral, D. DeWitt, and K. Wilkinson. Parallel algorithms for the execution of relational database operations. *ACM Transactions on Database Systems*, 8(3):324–353, September 1983.

[6] D. Burdick, P. Deshpande, T. S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. Efficient allocation algorithms for olap over imprecise data. In *VLDB*, pages 391–402, 2006.

[7] R. Chaiken, B. Jenkins, P.-A. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou. Scope: easy and efficient parallel processing of massive data sets. In *Proc. of the 34th Int. Conf. on Very Large DataBases (VLDB)*, 2008.

[8] A. J. Connolly, I. Csabai, A. S. Szalay, D. C. Koo, R. G. Kron, and J. A. Munn. Slicing through multicolor space: Galaxy redshifts from broadband p hotometry. *Astronomical Journal*, 110:2655+, Dec. 1995.

[9] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, Toronto, Canada, 2004.

[10] N. Dalvi and D. Suciu. Management of probabilistic data: Foundations and challenges. In *PODS*, pages 1–12, Beijing, China, 2007. (invited talk).

[11] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[12] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. In *Proc. of the 6th USENIX Symp. on Operating Systems Design & Implementation (OSDI)*, 2004.

[13] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, pages 588–599, 2004.

[14] D. DeWitt and J. Gray. Parallel database systems: the future of high performance database systems. *Communications of the ACM*, 35(6):85–98, 1992.

[15] E. Grädel, Y. Gurevich, and C. Hirsch. The complexity of query reliability. In *PODS*, pages 227–234, 1998.

[16] R. Gupta and S. Sarawagi. Creating probabilistic databases from information extraction models. In *VLDB*, pages 965–976, 2006.

[17] Hadoop. `http://hadoop.apache.org/`.

[18] A. J. S. Hamilton. Evidence for biasing in the CfA survey. ApJ, 331:L59–L62, Aug. 1988.

[19] D. W. Hogg, M. R. Blanton, J. Brinchmann, D. J. Eisenstein, D. J. Schlegel, J. E. Gunn, T. A. McKay, H.-W. Rix, N. A. Bahcall, J. Brinkmann, and A. Meiksin. The Dependence on Environment of the Color-Magnitude Relation of Galaxies. ApJ, 601:L29–L32, Jan. 2004.

[20] O. Ilbert, S. Lauger, L. Tresse, V. Buat, S. Arnouts, O. Le Fèvre, D. Burgarella, E. Zucca, S. Bardelli, G. Zamorani, D. Bottini, B. Garilli, V. Le Brun, D. Maccagni, R. Picat, J.-P. an d Scaramella, M. Scodeggio, G. Vettolani, A. Zanichelli, C. Adami, M. Arnaboldi, M. . Bolzonella, A. Cappi, S. Charlot, T. Contini, S. Foucaud, P. Franzetti, I. Gavignaud, L. Guzzo, A. Iovino, H. J. McCracken, B. Marano, C. Marinoni, G. Mathez, A. Mazure, B. Meneux, R. Merighi, S. Paltani, R. Pello, A. Pollo, L. Pozzetti, M. Radovich, M. Bondi, A. Bongiorno, G. Busarello, Y. Ciliegi, P. an d Mellier, P. Merluzzi, V. Ripepi, and D. Rizzo. The VIMOS-VLT Deep Survey. Galaxy luminosity function per morpholo gical type up to z = 1.2. A&A, 453:809–815, July 2006.

[21] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: Distributed data-parallel programs from sequential building blocks. In *Proc. of the European Conference on Computer Systems (EuroSys)*, pages 59–72, 2007.

[22] R. Jampani, F. Xu, M. Wu, L. Perez, C. Jermaine, and P. Haas. MCDB: a Monte Carlo approach to managing uncertain data. In *SIGMOD*, pages 687–700, 2008.

[23] T. Jayram, S. Kale, and E. Vee. Efficient aggregation algorithms for probabilistic data. In *SODA*, 2007.

[24] Z. Ma, W. Hu, and D. Huterer. Effects of Photometric Redshift Uncertainties on Weak-Lensing Tomo graphy. ApJ, 636:21–29, Jan. 2006.

[25] B. Mobasher, P. Capak, N. Z. Scoville, T. Dahlen, M. Salvato, H. Aussel, D. J. Thompson, R. Feldmann, L. Tasca, O. Lefevre, S. Lilly, C. M. Carollo, J. S. Kartaltepe, H. McCracken, J. Mould, A. Renzini, D. B. Sanders, P. L. Shopbell, Y. Taniguchi, M. Ajiki, Y. Shioya, T. Contini, M. Giavalisco, O. Ilbert, A. Iovino, V. Le Brun, V. Mainieri, M. Mignoli, and M. Scodeggio. Photometric Redshifts of Galaxies in COSMOS. ApJS, 172:117–131, Sept. 2007.

[26] P. Norberg, C. M. Baugh, E. Hawkins, S. Maddox, D. Madgwick, O. Lahav, S. Cole, C. S. Frenk, I. Baldry, J. Bland-Hawthorn, T. Bridges, R. Cannon, M. Colless, C. Collins, W. Couch, G. Dalton, R. De Propris, S. P. Driver, G. Efstathiou, R. S. Ellis, K. Glazebrook, C. Jackson, I. Lewis, S. Lumsden, J. A. Peacock, B. A. Peterson, W. Sutherland, and K. Taylor. The 2dF Galaxy Redshift Survey: the dependence of galaxy clustering on luminosity and spectral type. MNRAS, 332:827–838, June 2002.

[27] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig latin: a not-so-foreign language for data processing. In *SIGMOD'08: Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 1099–1110, 2008.

[28] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan. Interpreting the data: Parallel analysis with Sawzall. *Scientific Programming*, 13(4), 2005.

[29] M. Postman and M. J. Geller. The morphology-density relation - The group connection. ApJ, 281:95–99, June 1984.

[30] V. Rastogi, D. Suciu, and S. Hong. The boundary between privacy and utility in data publishing. In *VLDB*, 2007.

[31] C. Re, N. Dalvi, and D. Suciu. Efficient Top-k query evaluation on probabilistic data. In *ICDE*, 2007.

[32] C. Re and D.Suciu. Efficient evaluation of having queries on a probabilistic database. In *Proceedings of DBPL*, 2007.

[33] C. Re and D.Suciu. Materialized views in probabilistic databases for information exchange and query optimization. In *Proceedings of VLDB*, 2007.

[34] C. Re, J. Letchner, M. Balazinska, and D. Suciu. Event queries on correlated probabilistic streams. In *SIGMOD*, Vancouver, Canada, 2008.

[35] P. Sen and A. Deshpande. Representing and querying correlated tuples in probabilistic databases. In *ICDE*, 2007.

[36] R. K. Sheth, A. J. Connolly, and R. Skibba. Marked correlations in galaxy formation models. *ArXiv Astrophysics e-prints*, Nov. 2005.

[37] R. K. Sheth and G. Tormen. On the environmental dependence of halo formation. MNRAS, 350:1385–1390, June 2004.

[38] M. A. Soliman, I. F. Ilyas, and K. C.-C. Chang. Probabilistic top- and ranking-aggregate queries. *ACM Trans. Database Syst.*, 33(3), 2008.

[39] I. Zehavi, D. H. Weinberg, Z. Zheng, A. A. Berlind, J. A. Frieman, R. Scoccimarro, R. K. Sheth, M. R. Blanton, M. Tegmark, H. J. Mo, N. A. Bahcall, J. Brinkmann, S. Burles, I. Csabai, M. Fukugita, J. E. Gunn, D. Q. Lamb, J. Loveday, R. H. Lupton, A. Meiksin, J. A. Munn, R. C. Nichol, D. Schlegel, D. P. Schneider, M. SubbaRao, A. S. Szalay, A. Uomoto, and D. G. York. On Departures from a Power Law in the Galaxy Correlation Function. ApJ, 608:16–24, June 2004.