Probabilistic Histograms for Probabilistic Data

Graham Cormode
AT&T Labs–Research
graham@research.att.com

Antonios Deligiannakis
Technical University of Crete
adeli76@yahoo.com

Minos Garofalakis
Technical University of Crete
minos@acm.org

Andrew McGregor
University of Massachusetts, Amherst
mcgregor@cs.umass.edu

ABSTRACT

There is a growing realization that modern database management systems (DBMSs) must be able to manage data that contains *uncertainties* that are represented in the form of probabilistic relations. Consequently, the design of each core DBMS component must be revisited in the presence of uncertain and probabilistic information. In this paper, we study how to build histogram synopses for probabilistic relations, for the purposes of enabling both DBMS-internal decisions (such as indexing and query planning), and (possibly, user-facing) approximate query processing tools. In contrast to initial work in this area, our *probabilistic histograms* retain the key *possible-worlds semantics* of probabilistic data, allowing for more accurate, yet concise, representation of the uncertainty characteristics of data and query results. We present a variety of techniques for building optimal probabilistic histograms, each one tuned to a different choice of approximation-error metric. We show that these can be incorporated into a general Dynamic Programming (DP) framework, which generalizes that used for existing histogram constructions. The end result is a histogram where each "bucket" is approximately represented by a compact probability distribution function (PDF), which can be used as the basis for query planning and approximate query answering. We present novel, polynomial-time algorithms to find optimal probabilistic histograms for a variety of PDF-error metrics (including variation distance, sum squared error, max error and EMD_1). Our experimental study shows that our probabilistic histogram synopses can accurately capture the key statistical properties of uncertain data, while being much more compact to store and work with than the original uncertain relations.

1. INTRODUCTION

The need to represent and manipulate data in the presence of uncertainty has recently led to the development of several new *probabilistic* database management system (PDBMS) architectures. Novel PDBMSs, such as MayBMS [3], Trio [5], Mystiq [6], and MCDB [19] are concerned with giving users the power of a general-purpose relational DBMS to store and query uncertain data (in the form of probabilistic relations). Most of these early systems rely on *tuple- and attribute-level uncertainty models*, where the attribute values for a data tuple are specified using a probability distribution over different mutually-exclusive alternatives (that might also include *non-existence*, i.e., the tuple is not present in the data), and assuming independence across tuples. A probabilistic database is a concise representation for a probability distribution over an exponentially-large collection of *possible worlds*, each representing a possible "grounded" (deterministic) instance of the database (e.g., by flipping appropriately-biased independent coins to select an instantiation for each uncertain tuple). This "possible-worlds" semantics also implies clean semantics for queries over a probabilistic database — essentially, the result of a query defines a distribution over query results across all possible worlds [9, 10].

In building such a probabilistic DBMS, one must revisit all the core components of a DBMS, to understand how the presence of uncertainty affects their design and requirements. In particular, traditional DBMSs have grown to rely on compact *synopses* of relations, typically in the form of histograms and other statistics, in order to both enable informed internal decisions (e.g., on which attributes to build indices, and how to plan and execute queries), and allow for fast, approximate query answers for interactive data exploration and visualization. Such synopses can play similar roles in probabilistic databases — in fact, given known results on the *#P-hard complexity* of simple PDBMS query-processing tasks [9], one could argue that the need for effective, compact data representations is even more pronounced in this new setting. Such hardness arises from performing joins and projections between (statistically) independent probabilistic relations, and the need to track the history of generated tuples.

Various pragmatic question immediately arise: What is the correct generalization of the notion of a histogram to represent uncertain data? And, how can these synopses be used within a probabilistic database system? In this work, we introduce the idea of using a novel, powerful class of compact probabilistic representations (termed *probabilistic histograms*), and design efficient algorithms that construct the optimal such summaries for a probabilistic data set. In contrast to earlier approaches, our probabilistic histogram synopses retain a clean probabilistic semantics (concisely capturing an *approximate "possible worlds"* distribution); at the same time, they are also significantly smaller in size than the original data. This allows them to better approximate a wider range of possible queries than summaries with only deterministic information.

Related Work. *Histograms* have proven to be a very effective summarization mechanism for conventional (deterministic) data, and are currently the key synopses deployed in commercial query engines. Assuming a one-dimensional data distribution (capturing

VLDB '09, August 24-28, 2009, Lyon, France

tuple frequencies over the domain of an attribute), a histogram synopsis partitions the data domain into a small number of contiguous ranges (the "buckets"), and stores only concise statistics to summarize the tuple frequencies in a bucket (such as the value of the average bucket frequency). The goal is to group together tuple values with similar frequencies: bucket boundaries are chosen to minimize a given error function that measures within-bucket dissimilarities and aggregates errors across buckets (using summation or maximum). Much research has produced a variety of techniques to construct (deterministic) histograms which are optimal or near-optimal relative to particular error metrics, including Sum-Squared error, sum absolute relative error, and max-absolute error histograms, to name a few [15, 16, 18]. Other variations arise by extending the space of possible representations, for example, by allowing some outlier points to be removed [17]. Finally, techniques for *multi-dimensional* histograms that approximate multi-attribute data distributions using hyper-rectangular buckets over the data domain have also been developed [22, 23]. Although the problem of building histograms to summarize probabilistic data can be viewed as a two-dimensional histogram problem, there are additional challenges due to the probabilistic semantics. As such, naively trying to build a two-dimensional histogram over (value, probability) data does not give meaningful results for the error metrics we consider.

The fundamentals of PDBMSs have grown up over recent years [1]. The concept of using histograms and other statistics within an uncertain data-management system is a natural one [25]. Singh *et al.* [27, 26] experiment with replacing a single PDF with a more compact histogram, although they do not describe with regard to what error metric the histogram is chosen. Other efforts have focused on building compact summaries over streams of probabilistic tuples [20, 7, 21]; however, that work is geared to answering very specific types of queries (e.g., order statistics and moment estimates), rather than providing more general-purpose synopses with useful error guarantees.

Closest to our work, Cormode and Garofalakis [8] present techniques for building histogram summaries of probabilistic relations. Their work adopts conventional histogram representations, so the data domain is split into buckets and a *single representative value* is chosen to minimize the *expectation* of a given error metric (over the possible-worlds distribution). While useful for computations based on expected values, such conventional histogram summaries are more limited when it comes to capturing complex probabilistic information. Using a single value to approximately capture the distributions of values inside a bucket loses the probabilistic, possible-worlds semantics of the original data. This makes the summary less effective for answering certain probabilistic queries, such as selections on frequency distributions. A related shortcoming is that, in contrast to the deterministic case, the histograms of [8] do not give a *complete* representation scheme: even with maximum space, the histogram cannot represent the full details of the input distributions, since it is restricted to representing tuple frequency PDFs with trivial distributions that take a fixed value with probability one. Lastly, the methods we develop here are quite different from those in [8]: although both develop a dynamic programming framework (in common with most prior work on histogram construction), the methods needed to find the optimal representation of a given bucket (the core technical problem) are quite different in this setting.

Our Contributions. In this work, we consider a more powerful representation of uncertain data. Instead of representing a bucket of probability distribution functions (PDFs) with a single value, we choose to use a single PDF. This PDF itself can be thought of as a histogram: it describes a representative PDF with a small number of piecewise-constant segments. This representation is now *complete* for the probabilistic data we consider: with a large enough space budget, it is possible to represent the original relation *exactly*. While previous work considered probabilistic extensions of deterministic similarity metrics to measure the quality of the representation, we can now adopt widely used *probabilistic* error metrics, such as variation distance, Kullback-Leibler divergence (relative entropy) and Earth Mover's Distance. These measure give a solid theoretical foundation for the resulting summaries, which can then be used for query planning, analysis, and so on.

In this paper, we give algorithms to effectively find probabilistic histogram representations of uncertain data. In particular:

- We introduce the notion of a *probabilistic histogram*, and define the probabilistic histogram problem for a variety of probabilistic error metrics (Section 2).
- We give algorithms based on dynamic programming to find optimal probabilistic histograms for probabilistic data in time polynomial in the size of the input, for each of the different error metrics considered (Section 3).
- We discuss the use of these summaries within probabilistic data management systems for query planning and management (Section 4).
- We perform a set of experiments to demonstrate the power of out algorithms to summarize probabilistic data, and the benefits over prior representations (Section 5).

2. PROBLEM DEFINITION

Probabilistic Data Model. We consider summarizing probabilistic data that is presented in an appropriate model of data uncertainty. In particular, let $\mathscr{U}$ denote an ordered domain indexing the uncertain relation; for simplicity, we will assume that $\mathscr{U}$ is the set of integers $\{1 \ldots N\} = [N]$, so $|\mathscr{U}| = N$. The probabilistic input defines a distribution of "possible worlds" over this domain, which we think of as vectors f. A single (N-dimensional) "grounded" vector f provides a value for each member of $\mathscr{U}$. Each value is chosen from some value domain $\mathscr{V}$, so that $f_i \in \mathscr{V}$ $(i = 1, \ldots, N)$. We also let V denote the number of values in $\mathscr{V}$, i.e., $V = |\mathscr{V}|$. For instance, $\mathscr{U}$ could correspond to a set of mile-markers along a highway, and f_i is the (uncertain) temperature measured at mile i on a particular day. In another interpretation, each f_i represents the frequency of item i within a given relation.

A probabilistic model defines a probability distribution over such vectors (the possible worlds). Different types of models are able to express more or less complex distributions, with the choice of model trading-off descriptive power for the size of the resulting description. A *fully-general model* is able to describe any possible N-dimensional probability distribution, e.g., by listing each possible world and its corresponding probability. Unfortunately, instantiating such a model is complex and time consuming, due to the enormous number of possible worlds (requiring a number of parameters that is exponential in N). Instead, probabilistic data is more typically described through a model which makes certain *independence assumptions* to reduce the number of parameters of the model. Moreover, even if such correlations exist within the input data, their impact can be low, so ignoring them when computing summaries may have minimal effect on the quality of the summary. Our results here are presented in one such model:

Definition 1. In the *Item-PDF Model (IPM)*, each item $i \in \mathscr{U}$ is assumed to behave independently of all others. A PDF X_i is provided to describe the distribution of item i. The probability of any given possible world f under this model can then be calculated directly as $\Pr[f] = \prod_{i \in \mathscr{U}} \Pr[X_i = f_i]$.

It is not difficult to show that the above model can naturally capture most forms of tuple- and attribute-level uncertainty that have formed the basis of early PDBMS prototypes (it can be viewed as capturing the "OR-sets" in Trio [5] and Orion [27, 26], and as a special case of models in other systems such as Maybms [3] and MCDB [19]). Equivalently, this model can be seen as giving the distribution of X conditioned on the value of i, which captures the correlations between the variable and i. By restricting the representation of correlations across item values, the item-PDF model gives a more compact representation of the possible-worlds distribution (using $O(NV)$ parameters) than an exponential general model. Still, for large N and V, even an IPM can be large and unwieldy to compute with, thus raising the need for effective summarization techniques.

Probabilistic Histograms: Definition and Construction. A basic observation is that, in practice, the distributions of items adjacent under the ordering of $\mathscr{U}$ often tend to be quite similar — this is a natural extension of the "smoothness" properties of conventional data distributions, and so leads to the notion of using contiguous buckets (i.e., *histograms*) to obtain effective compact representations of the data. The histogram partitions the domain $\mathscr{U}$ into buckets, and all items within the same bucket are considered to behave identically to a chosen *bucket representative*. This representation can then be much more convenient to work with in place of the original data, especially when the number of buckets is not too large; moreover, if the "smoothness" assumption does indeed hold, then the result of using the histogram in place of the original data will yield query answers which are very close to the result of those queries on the original data.

Earlier work on histograms for probabilistic data considered only conventional histogram summaries where each bucket representative is *a single value* chosen to minimize an *expected error* (over possible worlds) [8]. Unfortunately, such simplistic synopsis structures lose all the probabilistic semantics of the underlying data and are only useful for expectation-based estimations—this severely limits their utility, e.g., as approximate query processing tools. In particular, simply knowing that the expected frequency of some item tells us very little about the probability that this item will appear i times, which is needed to approximate selections and joins (see Section 4). In general, there are an unbounded number of distributions with the same expectation, leading to high representation error. In this work, we consider a much richer histogram representation (termed *probabilistic histograms*), where the bucket representative is itself a (compact) distribution over $\mathscr{V}$. By allowing compact PDFs as bucket representatives, probabilistic histogram synopses retain a natural (albeit, approximate) possible-worlds semantics for the underlying data. However, this more complex histogram structure means that novel synopsis-construction tools are needed. The goal will be to choose a set of B bucket boundaries, and a representative PDF for each bucket, so that some overall error function (with respect to the original IPM) is minimized.

More formally, each probabilistic histogram bucket $b = (s,e)$ consists of a start point s and end point e, and covers $|b| = e - s + 1$ item PDFs. To summarize the PDFs $X_s, X_{s+1}, \ldots, X_e$ inside b, we also choose a *representative* for b which is itself a compact PDF $\hat{X}(b)$ over $\mathscr{V}$ (using single-value bucket representatives, as in [8], is equivalent to employing only trivial PDF representatives that place a mass of 1 at a single value). The accuracy with which a bucket representative captures the item PDFs that it represents is measured by a *bucket-error metric*. Let $d()$ denote a *PDF distance function*, i.e., a measure of the overall dissimilarity across two PDFs (we consider specific examples below). Then, we can define the error of a bucket b as

$$Err(b) = \bigoplus_{i=s}^{e} d(\hat{X}(b), X_i).$$

where $\bigoplus$ is an appropriate aggregation (typically, either sum or maximum). The histogram is defined by a set of B buckets which span the data domain $\mathscr{U}$, so that the k^{th} bucket spans $b_k = (s_k, e_k)$, where $s_1 = 1, e_B = N$, and $s_k = e_{k-1} + 1$ for $2 \leq k \leq B$. The overall error of the histogram representation can be computed as a summation or a maximum over the individual bucket errors:

$$S = \sum_{k=1}^{B} \sum_{i=s_k}^{e_k} d(\hat{X}(b_k), X_i) \quad \textit{Sum-error}, \qquad \text{or}$$

$$M = \max_{k=1}^{B} \sum_{i=s_k}^{e_k} d(\hat{X}(b_k), X_i) \quad \textit{Max-error}$$

We can now formally define the probabilistic-histogram construction problem that is the main focus of this paper.

Probabilistic Histogram Construction. Given a distance function $d()$, a space-complexity bound $\mathscr{S}$, and an input set of item PDFs $X_1, \ldots, X_N$ over $\mathscr{V}$, construct a probabilistic histogram of space complexity at most $\mathscr{S}$ which minimizes the histogram Sum- or Max-error under PDF distance $d()$. ∎

We consider two distinct ways to specify a space-complexity bound on the constructed probabilistic histogram:

- **B-bucket Case:** The histogram consists of exactly B buckets, each of which is represented by a detailed, V-term PDF over values $\mathscr{V}$. Such a representation makes sense when the size of the frequency-value domain, $\mathscr{V}$, is relatively small, and so each of these PDFs is quite small. In the B-bucket case, the overall space requirement of the probabilistic histogram is $\mathscr{S} = O(BV)$.
- **T-term Case:** When V is large, it makes more sense to try to find a T-term summary: if we represent each bucket-representative PDF by a set of piecewise constant values (i.e., a conventional histogram) then the total description length is the total number T of such constant terms across all bucket representatives. The overall space requirement of the histogram in this case is $\mathscr{S} = O(T)$.

It follows that, assuming the same overall space (i.e., $T = BV$), the T-term problem generalizes the corresponding B-bucket problem and has to search over a much larger space of space allotments (potentially giving smaller overall approximation error). As we will see, this comes at the cost of more complex and more expensive construction algorithms. Figure 1 shows an example probabilistic histogram over $N = 5$ input distributions. The illustrated histogram has $B = 2$ buckets (with $b_1 = (1,3)$ and $b_2 = (4,5)$; the total number of terms is $T = 5$, since the two summary PDFs can be described by 5 piecewise constant values.

PDF Distance Metrics. The choice of the PDF distance metric $d()$ has significant impact on the resulting histogram. The metric d is a function which takes two probability distribution functions over the value domain $\mathscr{V}$ and returns a measure of their dissimilarity. In the literature, several measures of dissimilarity between probability distributions have been proposed, each of which is appropriate in certain situations, depending on the application, and the nature of the errors that should be penalized (proximity in the value domain, absolute or relative error on the probability values). We describe some of the most popular possible choices of d:

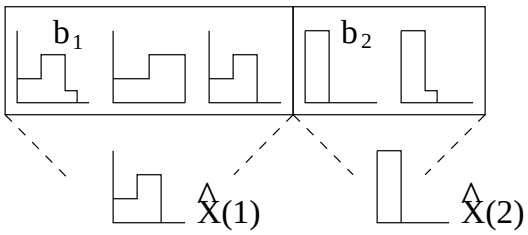


Figure 1: Summarizing a set of PDFs with compact PDF representatives.

- The *Variation Distance* (aka L_1) between two probability distributions over the same value domain $\mathcal{V}$ is the sum of absolute differences between the probabilities of each value. Formally, it is given by

$$d(X,Y) = \|X-Y\|_1 = \sum_{v\in\mathcal{V}} |\Pr[X=v] - \Pr[Y=v]|.$$

- The *Sum-squared error* (aka L_2^2) is similar to the Variation Distance, but takes the square of the difference of each pair of probabilities. It is defined by

$$d(X,Y) = \|X-Y\|_2^2 = \sum_{v\in\mathcal{V}} (\Pr[X=v] - \Pr[Y=v])^2.$$

- The *Kullback-Leibler divergence*, also known as the relative entropy, uses a more information theoretic approach to compare distributions. It is defined by

$$d(X,Y) = KL(X,Y) = \sum_{v\in\mathcal{V}} \Pr[X=v] \log_2 \frac{\Pr[X=v]}{\Pr[Y=v]}.$$

  Note that KL is not symmetric. It is natural to consider the second argument as the representative or approximation for the first argument.

- The *(Squared) Hellinger distance* is another commonly used measure of distribution similarity, given by

$$d(X,Y) = H^2(X,Y) = \sum_{v\in\mathcal{V}} \frac{(\Pr[X=v]^{\frac{1}{2}} - \Pr[Y=v]^{\frac{1}{2}})^2}{2}.$$

- The *Max-Error* measure (or L_∞) tracks the maximum difference between pairs of corresponding probabilities, and is given by

$$d(X,Y) = \|X,Y\|_\infty = \max_{v\in\mathcal{V}} |\Pr[X=v] - \Pr[Y=v]|$$

  Here, in contrast to other metrics, the error of a histogram bucket is taken to be the maximum of this value over the different PDFs, rather than the sum.

- The *Earth Mover's Distance in the L_p metric*, $EMD_p(X,Y)$ incorporates the "distance" between probabilities along the value domain. Conceptually, this measure represents the probability distributions as "piles of earth", and measures the difference as the total amount of work needed to convert one to the other, i.e. as the mass multiplied by the distance moved. This can be complex to compute for arbitrary distance metrics, but when used with L_p distances, it becomes simpler to work with. Further properties are described later.

Connection to Deterministic Case. This problem is more general than previous histogram problems that have been studied, since these can be seen as special cases of this one. Traditional histograms on deterministic data (such as V-opt histograms and Sum Absolute Error histograms) can be viewed as trying to find B-bucket histograms where all PDFs are restricted to be a fixed value, i.e. $\Pr[\hat{X}(b)=v]=1$ for some $v\in\mathcal{V}$. Measuring the error under Earth Mover's Distance with $p=1$ or $p=2$ corresponds to Sum Absolute Error and V-opt histograms respectively; other choices of the error metric generate other prior formulations of histograms. Similarly, prior work on summarizing probabilistic data [8] can be viewed as a restricted version of the above probabilistic histogram problem for versions of Earth Mover's Distance, where the B representative PDFs are limited to be fixed values. In the current formulation, there are fewer restrictions on the input and output, increasing the search space for possible histograms.

3. RESULTS

As with prior work, our solutions involve dynamic programming as a key component. Because the overall error objectives satisfy the *principle of optimality*, we can focus our attention on the central problem of finding the optimal representative for a given bucket. In the B-bucket case, if the final bucket spans $[i\ldots N]$, then the other $B-1$ buckets must form an optimal histogram for $[1\ldots i-1]$. Thus, Dynamic Programming (DP) over the choice of buckets will find the optimal solution: build a dynamic programming table over choices of parameters $k\le B$ and $i\le N$ recording the cost of the optimal k-bucket histogram covering the range $[1\ldots i]$. Similarly, for the T-term case, the same principle applies, to build a table giving the cost of constructing the optimal k-term histogram covering the range $[1\ldots i]$. However, this case is potentially more complex, since for each bucket considered, it is necessary to find the cost of representing it with 1 up to T terms.

In both cases, the heart of the solution is to determine the optimal representative for a particular bucket of PDFs defined by $b=(s,e)$. For the B-bucket case, the goal is to find the best unrestricted PDF that minimizes the cost of representing the PDFs in the bucket. In the T-term case, the goal is to find the best PDF that can be described with $1\ldots T$ piecewise constant terms representing the bucket. The solution to this core problem will depend on the choice of the error metric, so we will consider each in turn.

3.1 General Dynamic-Programming Scheme

We present the general DP formulation for constructing error-optimal probabilistic histograms in two parts: First, we give a dynamic program for computing the optimal (piece-wise constant) representative within a fixed bucket of PDFs. Then, we give the DP scheme that constructs the overall optimal probabilistic histogram (using the earlier, bucket-specific solution as a subroutine) for the T-term case. The same approach also holds for the B-bucket case, except that we use an "unrestricted" PDF (over $V=|\mathcal{V}|$ values) to represent the bucket.

Finding the Optimal Bucket Representative. Consider a particular bucket of PDFs $b=(s,e)$, where $s,e\in\mathcal{U}$ denote the two bucket boundaries. In the T-term case, we aim to find a T piece-wise constant representative (over $\mathcal{V}$) of the $e-s+1$ PDFs in b that minimizes the overall Sum/Max-error in the bucket for a particular error metric $d()$. Let $\textsc{ValErr}(b,v,w)$ denote the minimum possible value of the error in the approximation of all the probability values in the frequency range $r=(v,w)$ (where $v,w\in\mathcal{V}$) by the best single, constant representative for all the values in that set. (In a sense, this is the error in representing the set of values in a 2-dimensional $(e-s+1)\times(w-v+1)$ array by the best possible constant-value "centroid". The exact definition depends on the specifics of the un-

derlying PDF error metric $d()$.) Also, let B-OPT$^b[v,T]$ denote the optimal PDF approximation error up to point $v \in \mathcal{V}$ using at most T piece-wise constant segments. Based on the principle of optimality and assuming a Sum-error metric, we can write the following DP recurrence for computing B-OPT[] (the Max-error case is handled similarly):

$$\text{B-OPT}^b[w,T] = \min_{1 \le v \le w-1} \{\text{B-OPT}^b[v,T-1] + \text{VALERR}(b,v,w)\}.$$

The time complexity of the above DP clearly depends on the time required to compute VALERR(b,v,w), i.e., the optimal error for the constant approximation of a 2-dimensional $(e-s+1)\times(w-v+1)$ array of (frequency) values. Letting $t(e-s+1, w-v+1)$ denote that time, the complexity of our in-bucket DP is $O(V^2T\ t(e-s+1, w-v+1))$.

In the B-bucket case, we allow the representative PDF to fully describe the data. Here, the computational complexity can be lower, since the search space is smaller. The cost is $O(Vt(e-s+1,1))$ to find the optimal representative for all $v \in \mathcal{V}$.

Building the Optimal Probabilistic Histogram. Using H-OPT$[m,T]$ to denote the optimal error up to domain value $m \in \mathcal{U}$ and a total space budget of T terms, we can write the following DP recurrence:

$$\text{H-OPT}[m,T] = \min_{1 \le k \le m-1, 1 \le t \le T-1} \{\text{H-OPT}[k,T-t] + \text{B-OPT}^{(k+1,m)}[V+1,t]\}.$$

The naive cost of evaluating this recurrence over the $N = |\mathcal{U}|$ items is $O(N^2T)$ evaluations of B-OPT to find the costs of each bucket and T value. However, since B-OPT itself typically performs dynamic programming, computing B-OPT$^b[V+1,t]$ also generates the values of B-OPT$^b[V+1,t']$ for all $1 \le t' \le t$. Further, since a PDF has only V values, there is no benefit to assigning $t > V$ terms to a bucket PDF. Hence, we will usually only need to carry out $O(N^2)$ evaluations of B-OPT$^b[V+1,V]$ in the T-term case. The rest of the dynamic programming takes $O(N^2T\min(T,V))$ time, to compare all the possible choices of bucket boundaries and assignments of terms to a bucket.

Meanwhile, in the (simpler) B-bucket case, we have the same recurrence but without the ranging over t. This requires $O(BN^2)$ time to range over the bucket choices, and $O(N^2)$ evaluations of B-OPT$^b[V+1,V]$.

3.2 Sum-Squared Error

The sum-squared error metric is one of the most popular for building histograms on deterministic data — perhaps in part due to the fact that it can be computed efficiently. There, the optimum representative for a bucket is the mean of all the values that fall in the bucket. We show a similar result for probabilistic data.

LEMMA 1. *The optimal cost for representing a range of values in a particular bucket under Sum-Squared Error in the T-term case can be found in constant time using $O(VN)$ precomputed values.*

PROOF. Consider a range $r = (v,w)$ (where $v,w \in \mathcal{V}$) within a bucket $b = (s,e)$ that we wish to represent with a single value p. The contribution to the error is $\sum_{i=s}^{e}\sum_{j=v}^{w}(\Pr[X_i = j] - p)^2$. Differentiating with respect to p shows that this is minimized by setting

$$p = \bar{p} = \frac{\sum_{i=s}^{e}\sum_{j=v}^{w}\Pr[X_i = j]}{(e-s+1)(w-v+1)},$$

the average of the relevant probabilities. The cost VALERR(b,v,w) is then

$$\sum_{i=s}^{e}\Big(\sum_{j=v}^{w}(\Pr[X_i = j])^2 - 2\bar{p}\Pr[X_i = j] + \bar{p}^2\Big)$$
$$= \sum_{i=s}^{e}\Big(\sum_{j=v}^{w}(\Pr[X_i = j])^2\Big) - \bar{p}^2(e-s+1)(w-v+1) \quad .$$

This cost can be computed quickly based on $O(VN)$ precomputed values. Define

$$A[e,w] = \sum_{i=1}^{e}\sum_{j=1}^{w}\Pr[X_i = j] \quad B[e,w] = \sum_{i=1}^{e}\sum_{j=1}^{w}(\Pr[X_i = j])^2.$$

Then $\bar{p}[(s,e),(v,w)]\cdot(e-s+1)(w-v+1)$
$= (A[e,w] - A[s-1,w] - A[e,v-1] + A[s-1,v-1])$

and $\sum_{i=s}^{e}\sum_{j=v}^{w}(\Pr[X_i = j])^2$
$= B[e,w] - B[s-1,w] - B[e,v-1] + B[s-1,v-1]).$

From these, VALERR(b,v,w) can be computed in constant time. Last, note that this does indeed generate a valid PDF: clearly, each $\bar{p}$ is in $[0,1]$, since it is the mean of other probability values; and for a set of intervals $\mathcal{I} = \{(v,w)\}$ that partition $\mathcal{V}$, we have the cumulative probability,

$$\sum_{(v,w)\in\mathcal{I}}\sum_{j=v}^{w}\bar{p}[(s,e),(v,w)]$$
$$= \sum_{(v,w)\in\mathcal{I}}(w-v+1)\frac{\sum_{j=v}^{w}\sum_{i=s}^{e}\Pr[X_i = j]}{(e-s+1)(w-v+1)}$$
$$= \sum_{i=s}^{e}\sum_{j=v}^{w}\frac{\Pr[X_i = j]}{e-s+1} = \sum_{i=s}^{e}\frac{1}{e-s+1} = 1.$$

□

COROLLARY 2. *The optimal cost for a bucket in the B-bucket case under sum squared error can be found in constant time using $O(N)$ precomputed values.*

PROOF. In the B-bucket case, the cost is equivalent to the T-term case where we potentially choose a distinct probability for each $v \in \mathcal{V}$. This generates a representative PDF X for the bucket b where $\Pr[X = v] = \sum_{i=s}^{e}\Pr[X_i = v]/(e-s+1)$. Hence the cost is

$$\sum_{i=s}^{e}\sum_{j\in\mathcal{V}}(\Pr[X_i = j])^2 - \Pr[X = v]^2(e-s+1).$$

By precomputing $O(N)$ values in time $O(NV)$ as

$$A[e] = \sum_{i=1}^{e}\sum_{j=1}^{V}\Pr[X_i = j] \text{ and } B[e] = \sum_{i=1}^{e}\sum_{j=1}^{V}(\Pr[X_i = j])^2$$

the cost for the bucket can be found immediately as

$$B[e] - B[s-1] - \frac{(A[e] - A[s-1])^2}{e-s+1}.$$

□

In both cases the arrays A and B can be computed in $O(VN)$ time. Putting these results into the dynamic programming solution outlined in Section 3.1 allows us to conclude:

THEOREM 3. *The optimal T-term histogram of a probabilistic relation can be found in time $O(N^2T(\min(T,V)+V^2))$ under sum-squared error. The optimal B-bucket histogram of a probabilistic relation under the sum-squared error can be found in time $O(N(BN+V))$.*

Kullback-Leibler divergence and Sum-Squared Error. While L_2^2 and KL divergence appear very different measures of divergence, it follows from [11, Lemma 3] that the PDF X that minimizes $\sum_{i=s_k}^{e_k} KL(X_i,X)$ is the average of $\{X_i\}_{s_k\le i\le e_k}$, i.e., $(e_k-s_k+1)^{-1}\sum_{i=s_k}^{e_k} X_i$. Consequently the analysis of the B-bucket case for KL follows along identical lines to those in the SSE case, with the same costs.

3.3 Variation Distance

Recall that the variation distance between two PDFs is the sum of the absolute difference in probabilities for each value. In the T-term case, we are given a bucket b and a range of values $r=(v,w)$. We can write the contribution to the error when choosing a representative p as $\text{VALERR}(b,i,j)=\sum_{i=s}^{e}\sum_{j=v}^{w}|\Pr[X_i=j]-p|$. For minimization problems of this form, it is straightforward to show that the optimal solution is to choose the representative as

$$p = p_{med} = \text{median}_{\substack{s\le i\le e\\ v\le j\le w}} \Pr[X_i=j]$$

(Similar observations arise in other histogram problems [14]). Assuming for simplicity of notation that the median is unique and the number of items is even, we can write the error as

$$\text{VALERR}(b,v,w)=\sum_{i=s}^{e}\sum_{j=v}^{w}\Pr[X_i=j]-2I(i,j)\Pr[X_i=j]$$

where $I(i,j)$ is 1 if $\Pr[X_i=j]\le p_{med}$, and 0 otherwise.

Thus, the core problem is to be able to find the sum of a set of items smaller than the median. When applying the dynamic programming approach, we need the value of this quantity for *every* contiguous range of values and for every choice of bucket. We formalize this problem, and analyze its complexity.

LEMMA 4. *The* two-dimensional range-sum-median problem *is, given a two-dimensional array A of $m\times n$ values, to find*

$$\text{med}(a,b,c,d) = \text{median}_{\substack{a\le i\le c\\ b\le j\le d}} A[i,j]$$

$$\textit{and } \text{ms}(a,b,c,d) = \sum_{\substack{a\le i\le c\\ b\le j\le d\\ A[i,j]\le \text{med}(a,b,c,d)}} A[i,j]$$

for all $1\le a\le c\le m$ and $1\le b\le d\le n$. It can be solved in time $O((mn)^2\min(m,n)\log(mn))$. The one-dimensional range-sum-median problem *is the corresponding problem for a one dimensional array A of n values (equivalently, it is an instance of the two-dimensional problem with $m=1$). It can be solved in time $O(n^2\log n)$.*

PROOF. We first consider the one-dimensional version of the problem to find $\text{med}(a,c)$ and $\text{ms}(a,c)$ for all ranges. Note that this can be solved efficiently incrementally by fixing the value of a and stepping through the values of c. We can keep the (multi)set of values of $A[j]$ in a dynamic dictionary structure such as an AVL tree, from which we can find the desired quantities by tracking the number of items and sum of values within each subtree. Increasing c by one adds a new item to the tree, and so the total cost is $O(\log n)$ per update. Over the n^2 updates, the total cost is $O(n^2\log n)$. Note that, if all values are to be found, then the cost must be $\Omega(n^2)$, so this simple solution is near optimal.

The two-dimensional case is quite similar: assuming $m\le n$, we begin from each of the $O(mn)$ values of $[a,b]$, and fix a value of d. Then step through each possible value of c in turn. Each new value of c adds $O(m)$ new items into the tree, with cost $O(\log mn)$ per item. Again, the number of items and sum of values within each subtree is tracked, allowing the value of $\text{med}(a,b,c,d)$ and $\text{ms}(a,b,c,d)$ to be found. The total cost is then $O((mn)^2m\log(mn))$. For $m>n$, we perform the same operation but interchange the roles of c and d, giving cost $O((mn)^2n\log(mn))$. The claimed bound follows. □

Observe that in the T-term case, the full dynamic program has to find the cost of each range defined by a sub-bucket and a range of the value domain. Thus, the dynamic programming requires all the values generated by an instance of the two-dimensional range-sum-median problem. Once these have been computed in time $O((VN)^2\min(V,N)\log(VN))$, the process can find the cost of combination of bucket and value range in constant time. The dynamic program builds a table of size $O(NT)$ in time proportional to $O(N^2T\min(T,V))$.

In the B-bucket case, the process is somewhat simplified. Given a bucket b, the optimal representation is found by finding the median of the $(e-s+1)$ probabilities, for each of the V values. This can be aided by carrying out V parallel instances of the one-dimensional range-sum-median problem, one for each of the V values, in time $O(VN^2\log N)$. The dynamic programming then builds a table of size $O(N)$ in time $O(BN^2)$. Thus, in summary,

THEOREM 5. *The optimal T-term representation of a probabilistic relation under the variation distance can be found in time $O(N^2(T\min(T,V)+V^2\min(V,N)\log(VN)))$. The optimal B-bucket representation of a probabilistic relation under the variation distance can be found in time $O(N^2(B+\log(VN)))$.*

3.3.1 Normalization for Variation Distance.

While the representation generated minimizes the sum of absolute errors, the resulting representation of a bucket is not necessarily a PDF. That is, the sum of the "probabilities" may not be 1, as shown in the following example. Consider a bucket containing a single PDF over $\mathcal{V}=\{1,2,3,4,5\}$ given by

x	1	2	3	4	5
$\Pr[X=x]$	0	0	11/81	50/81	20/81

The optimal summary under variation distance with $T=2$ is

x	1	2	3	4	5
$\Pr[X=x]$	0	0	20/81	20/81	20/81

but this does not sum to 1. The optimal "normalized" summary is

x	1	2	3	4	5
$\Pr[X=x]$	0	0	0	1/2	1/2

This has implications for the use of this representation. While it is a good summary of the data, which minimizes a desired error metric, it is not normalized. Hence, it could cause unexpected results if passed on to other computations which expect a PDF as input.

Rescaling Solution. It is straightforward to rescale a representation so that it is normalized. But the example above shows that the optimal normalized summary is not necessarily a scaled version of the optimal unnormalized one. Nevertheless, let Y denote the optimal non-normalized bucket representative (e.g. the solution

found by the above dynamic programming solution), and consider the rescaled PDF $Z = Y/\mu$. Note that Z has the same space complexity as Y, and $\|Z-Y\|_1 = |1-\mu|$. Furthermore, for each summarized PDF X_i, we have $\|Y-X_i\|_1 \geq |\|Y\|_1 - \|X_i\|_1| = |1-\mu|$ since $\|X_i\|_1 = 1$. Therefore, by the triangle inequality,

$$\sum_{i=s}^{e} \|Z-X_i\|_1 \leq \sum_{i=s}^{e} \|Z-Y\|_1 + \|Y-X_i\|_1 \leq 2\sum_{i=s}^{e} \|Y-X_i\|_1$$

and so we find a solution whose error is at most a factor of two from optimal (since Y gives a lower bound on the error of the optimal normalized solution).

Discretized probability solution. Alternatively, we can try to find a tighter guarantee by adding a dimension to the DP table: let B-OPT$^b[w,T,\mu]$ denote the minimum error up to point $v \in \mathcal{V}$ using at most T terms such that the a values for points $1,\ldots,v$ sum up to μ. Note that B-OPT$^b[V,T]$ = B-OPT$^b[V,T,1]$ and can be found using the following recursion:

$$\text{B-OPT}^b[w,T,\mu] = \min_{1\leq v\leq w-1, 0<\nu<\mu} \{\text{B-OPT}^b[v,T-1,\mu-\nu] + \text{VALERR}(b,v,w,\nu)\}.$$

where VALERR(b,v,w,ν) is the error incurred by using value $(\mu-\nu)/(w-v+1)$ to approximate the values in the 2-dimensional $(e-s+1)\times(w-v+1)$ array. Unfortunately this recursion requires minimization of the continuous variable ν, which is not computationally feasible. So instead we work with values rounded to members of the following sets

$$S_1 = \{0, \frac{\varepsilon}{T}, \frac{2\varepsilon}{T}, \ldots, 1\}, \;\; S_2 = \{0, \frac{\varepsilon}{T}, \frac{(1+\varepsilon)\varepsilon}{T}, \frac{(1+\varepsilon)^2\varepsilon}{T}, \ldots 1\}$$

We compute a table $\Psi[v,t,\mu]$ for $v \in \mathcal{V}, t \in [T], \mu \in S_1$ so that

$$|\text{B-OPT}^b[w,t,\mu] - \Psi[w,t,\mu]| \leq 3\varepsilon t/T + \varepsilon\mu \; . \qquad (1)$$

Consequently we compute B-OPT$^b[v,t]$ while enforcing that the representative is normalized, and have additive error at most 4ε for any $t \leq T$. $\Psi[w,t,\mu]$ is defined by the following recursion:

$$\Psi[w,t,\mu] = \min_{1\leq v\leq w-1, \nu<\mu : \nu\in S_2} \{\Psi[v,t-1,f(\mu-\nu)] + \text{VALERR}(b,v+1,w,\nu)\}.$$

where $f(x) = \min\{x' \in S_1 : x \leq x'\}$. Let $g(x) = \min\{x' \in S_2 : x \leq x'\}$. For $0 \leq x \leq 1$, note that $f(x) - x \leq \varepsilon/T$ and $g(x) - x \leq \varepsilon/T + \varepsilon x$. We prove Eq. (1) by induction on t. For $t = 1$, $\Psi[v,1,\mu]$ = B-OPT$^b[v,1,\mu]$. For fixed $v < w \in \mathcal{V}, t \in [T], \mu \in S_1$, suppose $\nu = \nu^* \leq \mu$ minimizes

$$\text{B-OPT}^b[v,t-1,\mu-\nu] + \text{VALERR}(b,v+1,w,\nu)$$

$$\begin{aligned}
\text{Then } &\Psi[v,t-1,f(\mu-\nu^*)] + \text{VALERR}(b,v+1,w,g(\nu^*)) \\
&\leq \text{B-OPT}^b[v,t-1,f(\mu-\nu^*)] + 3\varepsilon(t-1)/T + \varepsilon f(\mu-\nu^*) \\
&\quad + \text{VALERR}(b,v+1,w,\nu^*) + \varepsilon(\nu^* + 1/T) \\
&\leq \text{B-OPT}^b[v,t-1,\mu-\nu^*] + \varepsilon/T \\
&\quad + 3\varepsilon(t-1)/T + \varepsilon(\mu-\nu^*+1/T) \\
&\quad + \text{VALERR}(b,v+1,w,\nu^*) + \varepsilon(\nu^*+1/T) \\
&= \text{B-OPT}^b[v,t-1,\mu-\nu^*] + \text{VALERR}(b,v+1,w,\nu^*) \\
&\quad + 3\varepsilon t/T + \varepsilon\mu
\end{aligned}$$

where the first inequality follows by the induction hypothesis and the triangle inequality in conjunction with a property of g. The second inequality uses the triangle inequality in conjunction with a property of f. Note that each of the $O(VT^2\varepsilon^{-1})$ values of $\Psi[\cdot,\cdot,\cdot]$ can be computed in $O(V\log(T\varepsilon^{-1}))$ time. Putting this into the DP recurrence gives:

THEOREM 6. *An ε-error (normalized) approximation to the optimal T-term probabilistic histogram of a probabilistic relation under variation distance can be found in time $O(N^2T^3V^2\varepsilon^{-1}\log(T\varepsilon^{-1}))$. An ε-error (normalized) approximation to the optimal B-bucket histogram can be found in time $O(N^2BV^4\varepsilon^{-1}\log(T\varepsilon^{-1}))$.*

3.4 Squared Hellinger Distance

The squared Hellinger distance is a commonly used metric for measuring the distance between probability distributions. Given a range $r = (v,w) \subset \mathcal{V}$ within a bucket $b = (s,e)$ that we wish to represent with a single value p, the squared Hellinger distance of the values within the bucket from p is given by:

$$\sum_{i=s}^{e}\sum_{j=v}^{w} \frac{(\sqrt{\Pr[X_i=j]} - \sqrt{p})^2}{2}.$$

Differentiating this expression with respect to p demonstrates that it can be minimized by setting p to the value:

$$p = \bar{p} = \left(\frac{\sum_{i=s}^{e}\sum_{j=v}^{w}\sqrt{\Pr[X_i=j]}}{(e-s+1)(w-v+1)}\right)^2,$$

LEMMA 7. *The optimal cost for representing a range of values in a particular bucket under the squared Hellinger distance in the T-term case can be found in constant time using $O(VN)$ precomputed values.*

PROOF. Consider a range $r = (v,w)$ (where $v,w \in \mathcal{V}$) within a bucket $b = (s,e)$ that we wish to represent with a single value p. The cost for the optimum p value is then

$$\begin{aligned}
&\sum_{i=s}^{e}\sum_{j=v}^{w} (\Pr[X_i=j] - 2\sqrt{\bar{p}}\sqrt{\Pr[X_i=j]} + \bar{p}) \\
&= \sum_{i=s}^{e}\sum_{j=v}^{w} \Pr[X_i=j] - (e-s+1)(w-v+1)\bar{p} \quad .
\end{aligned}$$

This is similar in form to the expression obtained for sum squared error. Hence, this cost can be computed quickly based on $O(VN)$ precomputed values in a similar way. Define

$$A[e,w] = \sum_{i=1}^{e}\sum_{j=1}^{w}\sqrt{\Pr[X_i=j]} \quad B[e,w] = \sum_{i=1}^{e}\sum_{j=1}^{w}\Pr[X_i=j].$$

$$\begin{aligned}
\text{Then } &\bar{p}[(s,e),(v,w)]\cdot(e-s+1)(w-v+1) \\
&= \frac{(A[e,w] - A[s-1,w] - A[e,v-1] + A[s-1,v-1])^2}{(e-s+1)(w-v+1)} \\
\text{and } &\sum_{i=s}^{e}\sum_{j=v}^{w}\Pr[X_i=j] \\
&= B[e,w] - B[s-1,w] - B[e,v-1] + B[s-1,v-1]).
\end{aligned}$$

□

COROLLARY 8. *The optimal cost for a bucket in the B-bucket case under the squared Hellinger distance can be found in constant time using $O(N)$ precomputed values.*

PROOF. This follows immediately, by only computing and storing the values of $A[e,V]$ and $B[e,V]$, and applying the above reduction. □

We note that in both cases the arrays A and B can be computed in $O(VN)$ time. In the T-term case this is dominated by the cost of the overall dynamic programming, and so can be ignored.

THEOREM 9. *The optimal T-term histogram of a probabilistic relation under squared Hellinger distance can be found in time $O(N^2T(\min(T,V)+V^2))$. The optimal B-bucket histogram can be found in time $O(N(BN+V))$.*

As in the variation error case, the resulting representation is not guaranteed to be a PDF, i.e., the probabilities do not necessarily sum to 1. Similar approaches can be used to find a solution that is normalized.

3.5 Max Error

The Max Error is defined as the maximum deviation between two distributions. Given a range $r=(v,w)$ within a bucket $b=(s,e)$ that we wish to represent with a single value p, it is

$$\max_{\substack{s\le i\le e\\ v\le j\le w}} |\Pr[X_i=j]-p|.$$

The above quantity is minimized by setting p to the value:

$$p=\frac{1}{2}(\max_{\substack{s\le i\le e\\ v\le j\le w}}\Pr[X_i=j]+\min_{\substack{s\le i\le e\\ v\le j\le w}}\Pr[X_i=j])$$

The dual problem is, given a deviation δ, to find a representation X of the PDFs $X_s\dots X_e$ so that $\max_{s\le i\le e}\|X-X_i\|_\infty\le\delta$ using as few subbuckets as possible. This can be solved easily with a single pass over the PDFs in the bucket. First, if there is any j such that $\max_{s\le i\le e}\Pr[X_i=j]-\min_{s\le i'\le e}\Pr[X_{i'}=j]\ge 2\delta$, then there is no solution for this choice of δ and bucket b. Otherwise, begin the first subbucket at value 1, and for each subbucket defined by a range $r=(v,w)$, track

$$\alpha=\max_{\substack{s\le i\le e\\ v\le j\le w}}\Pr[X_i=j]\quad\text{and}\quad\beta=\min_{\substack{s\le i\le e\\ v\le j\le w}}\Pr[X_i=j]\ .$$

If the current subbucket (v,w) has $\alpha-\beta>2\delta$, then we must terminate the current subbucket at $[v,w-1]$, and open a new subbucket at $[w,w]$. At the end of this process, we will have opened the smallest possible number of subbuckets while guaranteeing that the max error is at most δ, achieved by setting $p=(\alpha+\beta)/2$. The problem in the T-term case of finding a solution with at most T subbuckets can therefore be solved by (binary) searching over values of δ. This process can be made efficient by observing that this process only needs the maximum and minimum value for each $v\in V$. Using appropriate data structures, these can be found for any bucket in constant time per query after linear time preprocessing [13].

A more sophisticated argument can be applied to show that it suffices to search over only $O(V^2)$ different possible values of δ and moreover that this search can be done efficiently in total time $O(V)$, by adapting the search algorithm from [12].

For the B-bucket case, the smallest value of δ for a bucket b is

$$\max_{1\le v\le V}(\max_{s\le i\le e}\Pr[X_i=v]-\min_{s\le i\le e}\Pr[X_i=v]),$$

which gives the cost of picking that bucket. This value is found for a given bucket by finding for the minimum and maximum values in the range $s\dots e$, for each value $v\in\mathcal{V}$. Therefore, using appropriate range search structures [13], the the total query time is $O(BVN^2)$.

THEOREM 10. *The optimal T-term probabilistic histogram of a probabilistic relation under max-error can be found in time $O(TVN^2)$. The optimal B-bucket histogram can be found in time $O(BVN^2)$.*

We note that the resulting summary is not guaranteed to be normalized, since the total mass may not equal 1.

3.6 Earth Mover's Distance

The formal definition of the earth mover's distance between two distributions is based on a transfer function $\phi(v,w)$, which specifies how much "mass" to move from value v to value w. Then the cost for a given ϕ with L_p^p distance on the value domain is $\sum_{v,w\in\mathcal{V}}\phi(v,w)|v-w|^p$. The EMD_p cost between two distributions X and Y is the minimum over all ϕ such that applying ϕ to X generates Y (i.e., $\Pr[X=v]+\sum_{w\in\mathcal{V}}\phi(v,w)=\Pr[Y=v]$ for all v). This may appear complex, owing to the minimization over all possible transfer functions ϕ. However, because of the structure of PDF's over a value domain $\mathcal{V}$, the metric is considerably simplified. Given two PDFs over $\mathcal{V}=[1\dots V]$, X and Y, the Earth Mover's Distance EMD_p can be computed in a single pass. The procedure operates by considering each index i in turn: starting from $i=1$, if the difference $\Pr[X=i]-\Pr[Y=i]$ is positive, it is "moved" to index $i+1$, so that $\Pr[X=i+1]\leftarrow\Pr[X=i+1]+\Pr[X=i]-\Pr[Y=i]$, else $\Pr[Y=i+1]\leftarrow\Pr[Y=i+1]+\Pr[Y=i]-\Pr[X=i]$. $\mathrm{EMD}_1(X,Y)$ is given by the total amount of probability mass moved (i.e., the sum of the $|\Pr[X=i]-\Pr[y=i]|$ at each step).

Equivalently, this process can be thought of as operating on "atoms" of probability (sometimes also referred to as an "unfolded histogram"). For simplicity, assume that each probability in the PDF can be written as an integer multiple of some small quantity Δ. Then a PDF X can be written in terms of $1/\Delta$ such atoms: let $L_X[j]$ denote the position of the jth atom, so that $L_X[j]\le L_X[j+1]$ and $\Pr[X=i]=\Delta|j:L[j]=i|$. Then

$$\mathrm{EMD}_p(X,Y)=\sum_{j=1}^{1/\Delta}\Delta|L_X[j]-L_Y[j]|^p.$$

The correctness of this claim can be seen by observing that any transfer function ϕ defines a bijection between atoms defining X and Y. If $L_X[1]$ is mapped to $L_Y[j]$, and $L_X[j']$ is mapped to $L_Y[1]$, then the cost of the transfer is no more than if $L_X[1]$ is mapped to $L_Y[1]$, and $L_X[j']$ is mapped to $L_Y[j]$. By repeating this argument for each index in turn, we observe that the minimum cost mapping is when $L_X[j]$ is mapped to $L_Y[j]$, yielding the above cost formulation.

Given this characterization of the distance, the optimal unrestricted PDF to represent a collection of PDFs under EMD_p can be analyzed. If $\hat{X}$ is the representative PDF, then the cost for the bucket in the B-bucket case can be written as

$$\sum_{i=s}^{e}\mathrm{EMD}_p(\hat{X},X_i)=\Delta\sum_{j=1}^{1/\Delta}\sum_{i=s}^{e}|L_{\hat{X}}[j]-L_{X_i}[j]|^p$$

Thus we can minimize this cost by placing each atom of $\hat{X}$ in turn to minimize $\sum_{i=s}^{e}|L_{\hat{X}}[j]-L_{X_i}[j]|^p$.

We now present results specific to the EMD_1 case. By analogy with 1-median clustering [24], the optimal choice is to set $L_{\hat{X}}[j]=\mathrm{median}_{i\in b}L_{X_i}[j]$. This clearly gives a valid PDF: the total probability mass remains 1, since there is a location for each atom of probability. Further, the atoms are placed in increasing order along the value domain, since $L_{X_i}[j]\le L_{X_i}[j+1]$ for all i, j, and so $L_{\hat{X}}[j]\le L_{\hat{X}}[j+1]$. The cost of using this representative $\hat{X}$ is then

$$\Delta\sum_{j=1}^{1/\Delta}\sum_{i=s}^{e}|L_{\hat{X}}[j]-L_{X_i}[j]|=\Delta\sum_{j=1}^{1/\Delta}(\sum_{i=s}^{e}L_{X_i}[j]-2I(i,j)L_{X_i}[j])$$

where $I(i,j)$ is an indicator variable that is 1 if $L_{X_i}[j]<\mathrm{median}_i L_{X_i}[j]$, and 0 otherwise. Observe that this can be solved

with the answers to multiple instances of the one-dimensional range-sum-median problem (Lemma 4). For each atom, we need to find the sum of values below the median for the locations of the atom over the bucket b. In the B-bucket case, the dynamic programming algorithm considers all possible buckets in order to choose B optimal bucket boundaries. After the $O(\frac{N^2}{\Delta}\log N)$ cost of solving $1/\Delta$ instances of the range-sum-median problem, the cost of any bucket can be found in $O(1/\Delta)$ time. Therefore, when all probabilities are multiples of Δ,

THEOREM 11. *The optimal B-bucket representative of a probabilistic relation under the* EMD_1 *metric can be found in time* $O(N^2(B+\frac{\log(N)}{\Delta}))$.

For the T-term case, the process is more complicated. A natural approach is to apply the dynamic programming approach within a bucket, by choosing a single representative value for a range of atoms $[a,b]$. However, this results in placing all those atoms at a single location in the representative PDF: this generates a PDF which consists of a small number of "impulses" at particular locations. While the resulting histogram is optimal from the space of all histograms consisting of PDFs with a total of T impulses, it does not match our original requirements to find a T-term histogram.

Instead, we can make use of the observation that $\mathrm{EMD}_1(X,Y) = \|F(X)-F(Y)\|_1$: that is, the distance is equivalent to the L_1 distance between the cumulative probability distributions of X and Y, denoted by $F(X)$ and $F(Y)$ respectively. This follows from the previous analysis by allowing Δ to tend to 0. Note that the cumulative distribution of a PDF represented by a histogram with t terms on $\mathscr{V}$ is a non-decreasing t-piecewise linear function G that is (a) continuous in the sense that each consecutive pair of linear segments meet at a common point and (b) normalized in the sense that $G(0)=0$ and $G(V)=1$. Hence we wish to find such a function G that minimizes $\sum_i \|F(X_i)-G\|_1$. This can be done by simple modifications of the approximation algorithm of Aronov et al. [4] to ensure that f is normalized and non-decreasing. However, since the cost is quartic in the number of points (i.e. $O(((s-e+1)V)^4)$, this approach is unlikely to be practical for large instances.

4. QUERY ANSWERING WITH HISTOGRAMS

A probabilistic histogram (irrespective of which error metric it is built for) can be used to approximate a variety of queries. Extracting basic statistics such as expected values from the histogram is immediate. In this section, we discuss how to apply selections and joins; more complex queries can be naturally be evaluated in a similar manner, but are beyond the scope of this paper. A nice property is that many of these operations are *closed* for probabilistic histograms: applying an operation generates a new summary that is also a probabilistic histogram, with space cost $\mathscr{S}$ that is closely related to the original cost of the input histogram.

4.1 Selection

Any selection query on the item domain identifies a subset of tuples $\mathscr{C} \subseteq \mathscr{U}$. The distribution of these tuples is modeled by projecting the histogram onto this support set $\mathscr{C}$. When this selection is simple, such as a range selection, the result of the projection is itself a probabilistic histogram.

The case when the selection is on the value domain $\mathscr{V}$ is more interesting. The probabilistic histogram retains enough information to summarize the full distribution of tuples which pass the selection: essentially, the result is a histogram where each bucket PDF is the conditional distribution, conditioned on the predicate P. That is, the new conditional distribution $\hat{X}(b,P)$ for bucket b has

$$\Pr[\hat{X}(b,P)=v|P(v)] = \frac{\Pr[\hat{X}(b)=v]}{\sum_{v|P(v)}\Pr[\hat{X}(b)=v]},$$

and zero otherwise (i.e. $\Pr[\hat{X}(b,P)=v|\neg P(v)]=0$). Moreover, when $\hat{X}(b)$ is given by some small number of terms t and P corresponds to a range predicate, the resulting PDF is also represented by at most $t+2$ terms (at most two new terms may be needed at the extreme ends of the distribution).

When combined with an aggregation, information can be found quickly. For instance, the expected number of distinct tuples selected is computed easily from the buckets. The expected number of tuples passing a predicate P is given by $\sum_{k=1}^{B}(e_k-s_k+1)\sum_{v\in\mathscr{V}|P(v)}\Pr[\hat{X}(b_k)=v]$, where $\hat{X}(b_k)$ indicates the PDF representing the kth bucket, b_k. But more than this, the distribution of the number of tuples selected has a simple form. Let $P(\hat{X}(b))$ be shorthand for $\sum_{v\in\mathscr{V}|P(v)}\Pr[\hat{X}(b)=v]$. Then, for a bucket b, the distribution of the number of distinct tuples selected by P is $\mathrm{Bin}((e-s+1),P(\hat{X}(b)))$, the binomial distribution with $n=(e-s+1)$ and $p=P(\hat{X}(b))$ (since we treat each item as independent). Consequently, the distribution over the whole histogram is $\sum_{k=1}^{B}\mathrm{Bin}((e-s+1),P(\hat{X}(b)))$.

4.2 Join

We focus on the question of joining two probabilistic relations where the join is an equijoin on $\mathscr{U}$ with an additional join condition on the common uncertain domain $\mathscr{V}$. In this case, the input is two probabilistic histograms representing two relations. However, we do not need to assume that the two histograms share the same bucket boundaries. We make the observation that, given two histograms with B_1 and B_2 buckets respectively, together this defines a partition of $\mathscr{U}$ with at most B_1+B_2-1 non-overlapping ranges. There is a unique bucket from each histogram, say b^1 and b^2 which covers all items in each of a given range. These two buckets define a distribution over items in the range which, by assuming independence between the two relations is the product distribution, can be written as $\Pr[X=(v_1,v_2)]=\Pr[\hat{X}(b^1)=v_1]\Pr[\hat{X}(b^2)=v_2]$.

We assume for simplicity that the join is an equijoin on $\mathscr{V}$ (other join types are similar). Then the join tuple(s) within the overlap of buckets b_1 and b_2 are distributed as

$$\Pr[\hat{X}(b_1,b_2)=v]=\Pr[\hat{X}(b_1)=v]\Pr[\hat{X}(b_2)=v].$$

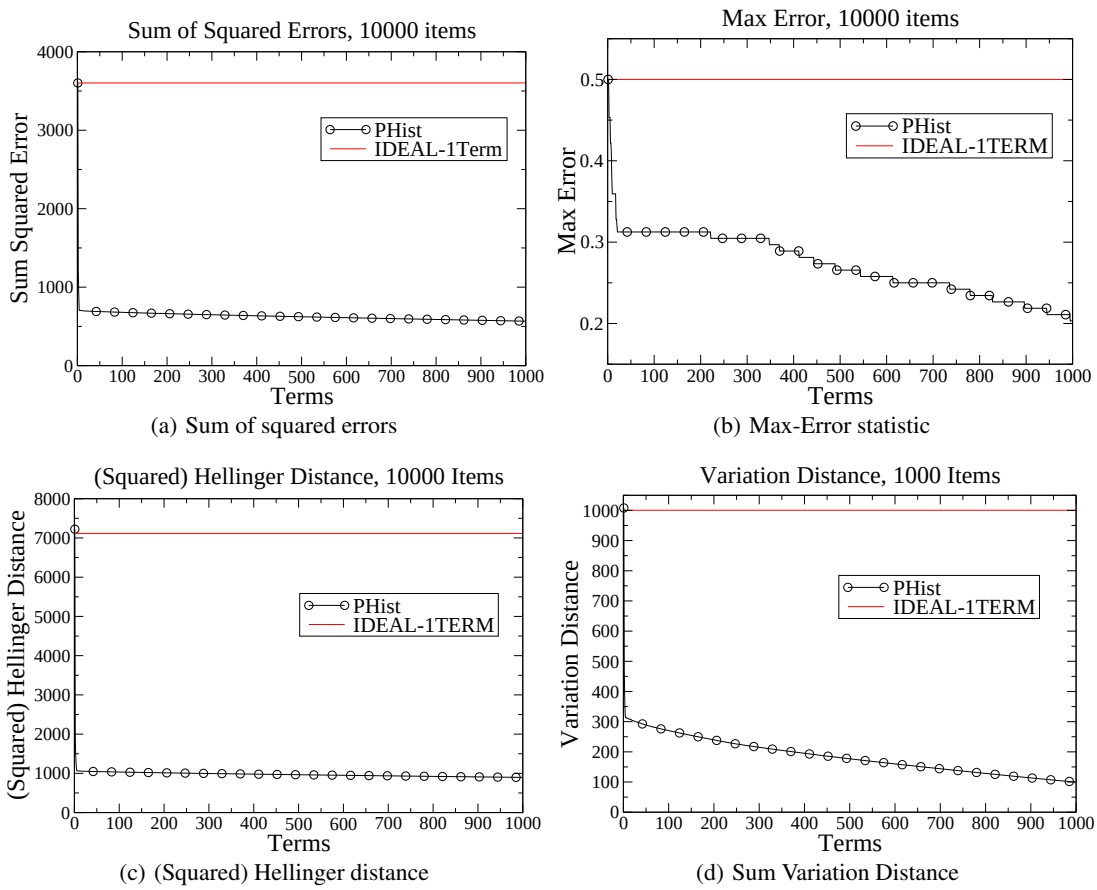
By the same style of argument as above, if buckets b_1 and b_2 are represented by t_1 and t_2 terms respectively, the resulting PDF can be represented by at most t_1+t_2-1 terms. Thus, probabilistic histograms are closed under join operations like these. From the resulting histograms, it is straightforward to extract expected values, tail bounds on distributions, and so on.

5. EXPERIMENTS

We implemented our algorithms for building probabilistic histograms (denoted PHIST) in C, and carried out a set of experiments to compare the quality and scalability of our results against techniques that assign a single term to each histogram bucket (described in more detail below). The experiments were performed on a server equipped with 4 Intel Xeon CPUs clocked at 1.6GHz, and 8GB of RAM. Each experiment was run on a single CPU.

Data Sets. We experimented using a mixture of real and synthetic data sets. The real data set came from the MystiQ project[1] which

[1] `http://www.cs.washington.edu/homes/suciu/`



(a) Sum of squared errors (b) Max-Error statistic

(c) (Squared) Hellinger distance (d) Sum Variation Distance

Figure 2: Results on Histogram Computation

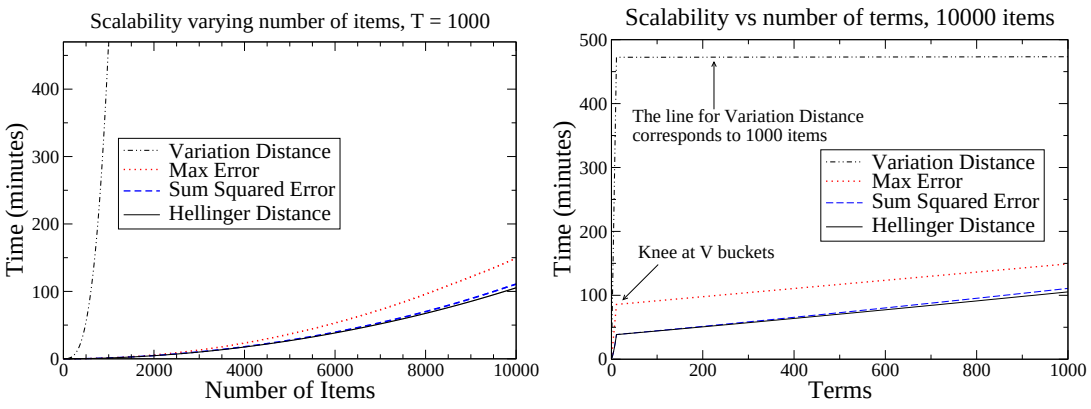
includes approximately 127,000 tuples describing 27,700 distinct items. These correspond to links between a movie database and an e-commerce inventory, so the tuples for each item define the distribution of the number of expected matches, formed by combining individual tuple linkage probabilities into PDFs. In this data set the maximum frequency of any item was 10, thus requiring us to estimate $V = 11$ frequency probabilities for each item (i.e., the probability that the frequency of each item is 0, 1,...,10). We also performed experiments on synthetic data generated using the MayBMS [3] extension to the TPC-H generator [2]. The results on this data were broadly similar to those on the real data set and so are omitted for space reasons.

Alternative Techniques Considered. We contrast our methods with a technique, termed as IDEAL-1TERM, that uses a distinct bucket to represent each item of the data, but limits the representation within each bucket to a single term. Essentially, this technique corresponds to the optimal that one may achieve when restricted to using a single term per bucket, if no additional space constraints are imposed. This shows a lower bound on the best that can be obtained by any algorithm which uses a single value to represent a bucket. In this experimental evaluation we do not explicitly compare our techniques to any algorithms which do assign a single value to a bucket, such as algorithms for deterministic data or the algorithms in [8] which were targeted at different error metrics. As we see empirically, the error for the IDEAL-1TERM algorithm, which represents a lower bound on what can be achieved by any such algorithm, is still significantly in excess of that obtained by our methods. Such representations provide no information on how the retained frequency can be translated into a PDF of the frequencies of each item. A naive solution is to substitute the expected frequency $E[f_i]$ of each item i with a PDF that contains a probability of 1 for the frequency value $E[f_i]$ (although in general $E[F_i] \notin \mathcal{V}$) and zero probabilities elsewhere. Any such representation results in very large errors for the metrics that we seek to minimize.

Result Quality. We use our methods described in Section 3 to build our PHIST histograms over N items using T terms, and compute the cost of the histograms under the relevant metric. The quality of our PHIST histograms is shown in Figure 2. In this figure, our techniques for minimizing the sum of squared errors, the (squared) Hellinger distance and the Max-Error metric are applied on the same $N = 10^4$ distinct data items. In the Variation Distance case, we limit to the first $N = 10^3$ distinct data items, as the computational cost for this algorithm is much higher. The general trend for all methods is the same: for the error metrics we consider, the probabilistic histogram approach of representing buckets PDFs with a representative PDF is more accurate than picking a single value, even if (as in the IDEAL-1TERM case) the algorithm is allowed to treat each PDF separately. The IDEAL-1TERM algorithm can never achieve zero error, even though it uses N buckets, since each bucket is limited to contain a single term. Comparing our PHIST histograms with the IDEAL-1TERM technique, we notice that the

project-mystiq.html

[2] www.cs.cornell.edu/database/maybms/



(a) Time as the number of items N varies (b) Time as T varies

Figure 3: Histogram Timing Costs

errors of our techniques are significantly lower (even when using just a few terms) than those of IDEAL-1TERM, even though the latter uses much more space (N buckets). This is due to the more intelligent partitioning of the domain space. The two have very similar errors when PHIST is restricted to a constant number of terms. This clearly demonstrates the need for using multiple terms in order to better approximate this probabilistic data set.

For sum squared error and the similar Hellinger distance (Figures 2(a) and 2(c) respectively), the cost decreases gradually as more terms are allowed. This suggests that there is relatively little benefit in using a large number of terms for this data set: around 100 terms seems to capture the behavior almost as well as 1000. (Note that with $N = 10000$ and $V = 11$, it would take $T = 110000$ terms to fully represent the input data with zero error). For the max-error case and the Variation Distance (Figures 2(b) and 2(d) respectively), there is a clearer benefit to adding more terms, with a more pronounced decrease of error.

Scalability. Figure 3 shows the time cost of our methods for the four metrics depicted in Figure 2. Figure 3(a) shows the time taken as the number of items (N) increases: it reveals a quadratic relationship between the running time and N for the sum of squared errors, the Hellinger Distance and the Max-Error cases. This is in line with the asymptotic costs for these methods given by our analysis, in which the leading term is N^2. The costs for the Sum Squared Error and the Hellinger Distance are quite close, mainly due to the similarity in the form of their solution. For Max-Error, the cost is slightly higher, and grows slightly faster. This is due in part to the fact that our implementation used a simpler version of the algorithm to determine the optimal bucket representative, which adds a logarithmic factor to the running time complexity of approximating each bucket. Thus, we expect that the running time of the PHIST algorithm for a more elaborate implementation would more closely match the running time for the Hellinger Distance and the sum of squared errors. From Figure 3(a) it is also clear that minimizing the variation distance results in high running times, which is approximately cubic with N in our implementation.

Figure 3(b) shows a clear linear trend as the number of terms increases, as predicted by the analysis, but there is also a sharp "knee" in the line for a small number of terms. This knee occurs precisely where $T = V$, and is explained by the fact that, up to this point, the algorithm has to explore increasingly many combinations of ways to choose $T < V$ terms to represent any bucket. But it makes no sense to assign more than 1 term for each of the V possible frequency values within a bucket. The error of a bucket using more than V terms is identical to the corresponding error when using exactly V terms. As mentioned in the discussion of the dynamic programming solution (Section 3.1), for values of $T > V$, and for any assignment of T terms to each possible bucket, the optimal T-term representation of the aforementioned bucket is guaranteed to have been computed in a prior iteration of the algorithm. From the analysis of the algorithms, for $T \leq V$, the cost on all the metrics considered grows in proportion to T^2 ; for $T > V$, all methods grow proportional to T, resulting in the linear growth pattern observed for larger values of T in Figure 3(b).

6. CONCLUDING REMARKS

Other PDF distance metrics. It is certainly possible to consider other error metrics, by analogy with metrics used for histogram construction over deterministic data. For instance, the sum squared error can be extended to sum squared *relative* error, $\sum_{v \in \mathcal{V}} (\Pr[X = v] - \Pr[Y = v] / \max(c, \Pr[X = v]))^2$ for a chosen constant c. It is possible to extend our methods to cover such metrics, but we do not consider them in detail, since this measure can place undue emphasis on small probability values. Other variations of the Earth Movers' Distance, such as EMD_2, will also be important to address.

Other probabilistic models of uncertain data. We have primarily considered models of probabilistic data which capture first-order correlations, but not higher correlations between items, or across relations. This is in line with the emerging probabilistic systems, since such higher order effects are relatively small, and are harder to measure and model. However, it remains of some interest to extend the models we consider to be able to model and summarize data with explicit correlations between items or across relations. In the meantime, ignoring such effects merely weakens the quality of the approximation in the intended applications; similar issues affect histograms for deterministic data, which typically do not capture complex correlations across tuples or relations.

A natural next target would be graphical models, which make explicit the dependence of some variables on other variables. In particular, *Bayesian networks* show how the probability of one event depends on a particular subset of other events (and is assumed independent of all others). To some extent, our work can already address this setting. As remarked earlier, the item PDF model which presents distributions X_i can be seen as giving the distribution of X

conditioned on i. More generally, a relation might give information about the distribution of X conditioned on other variables $Y, Z \ldots$. The probabilistic histogramming approach can certainly be applied to this setting after applying a linearization to the other variables to form a one-dimensional ordering (i.e. treating the linear ordering of X given $Y = y, Z = z$ for all values of y and z as a single relation to build a histogram of). However, a more faithful representation may be found by more explicitly taking the semantics of the distributions into account, for example by histogramming the two dimensional space given by the cross-product of Y and Z. This in turn brings new challenges, in particular how to scalable build probabilistic histograms of multidimensional data.

Faster computation. As observed in the experimental section, the algorithms proposed have potentially high cost. Although the histogram construction may only need to be done occasionally, when new uncertain relations are inserted into the PDBMS, it is still desirable to improve the speed. We observe that the majority of the cost is due to the dynamic programming approach, whose cost is polynomial in the parameters of the input. There are many potential savings: (1) It is not worthwhile to exhaustively consider every possible bucketing. Large buckets contribute large errors, and are unlikely to be used in any optimal histogram. Rather than compute the exact bucket cost, various lower bounds can be used. For example, when the distance d obeys the triangle inequality, we have $Err(b) = \sum_{i=s}^{e} d(\hat{X}, X_i) \geq \frac{1}{2} \sum_{i=s}^{e-1} d(X_i, X_{i+1})$. This bound can be computed in constant time with some precomputation, allowing large buckets to be ignored. (2) Various techniques are known for the dynamic programs that arise in histogram construction [14]. These also apply in our case, and so can be used to speed up the search. The result is an approximation to the optimal histogram; since the histograms are themselves used to approximate query answering, this approximation is usually acceptable.

7. REFERENCES

[1] C. Aggarwal, editor. *Managing and Mining Uncertain Data*. Springer, 2009.

[2] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *ACM Symposium on Theory of Computing*, 1996.

[3] L. Antova, T. Jansen, C. Koch, and D. Olteanu. Fast and simple relational processing of uncertain data. In *IEEE International Conference on Data Engineering*, 2008.

[4] B. Aronov, T. Asano, N. Katoh, K. Mehlhorn, and T. Tokuyama. Polyline fitting of planar points under min-sum criteria. *Int. J. Comput. Geometry Appl.*, 16(2-3):97–116, 2006.

[5] O. Benjelloun, A. D. Sarma, C. Hayworth, and J. Widomn. An introduction to ULDBs and the Trio system. *IEEE Data Engineering Bulletin*, 29(1):5–16, Mar. 2006.

[6] J. Boulos, N. Dalvi, B. Mandhani, S. Mathur, C. Re, and D. Suciu. Mystiq: A system for finding more answers by using probabilities. In *ACM SIGMOD International Conference on Management of Data*, 2005.

[7] G. Cormode and M. Garofalakis. Sketching probabilistic data streams. In *ACM SIGMOD International Conference on Management of Data*, 2007.

[8] G. Cormode and M. Garofalakis. Histograms and wavelets on probabilistic data. In *IEEE International Conference on Data Engineering*, 2009.

[9] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *International Conference on Very Large Data Bases*, 2004.

[10] N. Dalvi and D. Suciu. Management of probabilistic data: foundations and challenges. In *ACM Principles of Database Systems*, 2007.

[11] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, 2003.

[12] H. Fournier and A. Vigneron. Fitting a step function to a point set. In *European Symposium on Algorithms (ESA)*, 2008.

[13] H. Gabow, J. Bentley, and R. Tarjan. Scaling and related techniques for geometry problems. In *ACM Symposium on Theory of Computing*, 1984.

[14] S. Guha, N. Koudas, and K. Shim. Approximation and streaming algorithms for histogram construction problems. *ACM Transactions on Database Systems*, 31(1):396–438, 2006.

[15] S. Guha, K. Shim, and J. Woo. REHIST: Relative error histogram construction algorithms. In *International Conference on Very Large Data Bases*, 2004.

[16] Y. E. Ioannidis. The history of histograms (abridged). In *International Conference on Very Large Data Bases*, 2003.

[17] H. V. Jagadish, N. Koudas, and S. Muthukrishnan. Mining deviants in a time-series database. In *International Conference on Very Large Data Bases*, 1999.

[18] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. Sevcik, and T. Suel. Optimal histograms with quality guarantees. In *Int. Conf. on Very Large Data Bases*, 1998.

[19] R. Jampani, L. L. Perez, F. Xu, C. Jermaine, M. Wi, and P. Haas. MCDB: A monte carlo approach to managing uncertain data. In *ACM SIGMOD International Conference on Management of Data*, 2008.

[20] T. S. Jayram, S. Kale, and E. Vee. Efficient aggregation algorithms for probabilistic data. In *ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[21] T. S. Jayram, A. McGregor, S. Muthukrishnan, and E. Vee. Estimating statistical aggregates on probabilistic data streams. In *ACM Principles of Database Systems*, 2007.

[22] S. Muthukrishnan, V. Poosala, and T. Suel. On rectangular partitionings in two dimensions: Algorithms, complexity, and applications. In *International Conference on Database Theory*, 1999.

[23] V. Poosala and Y. E. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *International Conference on Very Large Data Bases*, 1997.

[24] F. P. Preparata and M. Shamos. *Computational Geometry : An Introduction*. Springer-Verlag, 2nd edition, 1985.

[25] A. D. Sarma, P. Agrawal, S. Nabar, and J. Widom. Towards special-purpose indexes and statistics for uncertain data. In *Proceedings of the Workshop on Management of Uncertain Data (MUD)*, 2008.

[26] S. Singh, C. Mayfield, R. Shah, S. Prabhakar, and S. Hambrusch. Query selectivity estimation for uncertain data. In *Statistical and Scientific Database Management (SSDBM)*, 2008.

[27] S. Singh, C. Mayfield, R. Shah, S. Prabhakar, S. Hambrusch, J. Neville, and R. Cheng. Database support for probabilistic attributes and tuples. In *IEEE International Conference on Data Engineering*, 2008.