Recovering Information from Summary Data

*Christos Faloutsos**
Dept. of Computer Science and
Inst. of Systems Research
Univ. of Maryland
College Park MD 20742
christos@cs.umd.edu

H. V. Jagadish
AT&T Laboratories
Florham Park, NJ 07932
jag@research.att.com

N. D. Sidiropoulos†
Dept. of Electrical Eng.
Univ. of Virginia
Charlottesville, VA 22903

Abstract

Data is often stored in summarized form, as a histogram of aggregates (COUNTs, SUMs, or AVeraGes) over specified ranges. We study how to estimate the original detail data from the stored summary.

We formulate this task as an *inverse problem*, specifying a well-defined cost function that has to be optimized under constraints. We show that our formulation includes the uniformity and independence assumptions as a special case, and that it can achieve better reconstruction results if we maximize the smoothness as opposed to the uniformity. In our experiments on real and synthetic datasets, the proposed method almost consistently outperforms its competitor, improving the root-mean-square error by up to 20 per cent for stock price data, and up to 90 per cent for smoother data sets.

Finally, we show how to apply this theory to a variety of database problems that involve partial information, such as OLAP, data warehousing and histograms in query optimization.

*This research was partially funded by the National Science Foundation under Grants No. EEC-94-02384, IRI-9205273 and IRI-9625428.

†Part of this work was performed while with the Inst. for Systems Research, Univ. of Maryland at College Park.

**Proceedings of the 23rd VLDB Conference
Athens, Greece, 1997**

1 Introduction

Consider the problem of an unknown set of numbers x_i $(i = 1, \ldots N)$, for which we are given some partial information. For example, x_i could represent the total sales for the i-th day, and we could be given only the monthly total sales. Suppose that we also have some additional, a-priori information, for example, that the sales patterns are "smooth", without abrupt jumps (*i.e.*, $x_i \approx x_{i+1}$). The goal is to recover the unknown values as best as we can[1].

In a multi-dimensional setting, this problem becomes even more interesting. Suppose that the unknown numbers are the counts $c_{i,j}$ of employees of a company, for each age-bracket i and for each salary-bracket j; suppose that we are only given the age- and salary-histograms, that is the counts $c_{i,*}$ for the i-th age-bracket and the counts $c_{*,j}$ for the j-th salary-bracket. The goal is to estimate the unknown $c_{i,j}$ counts.

This sort of problem arises in a host of different situations. Data is summarized over discrete ranges to create a database of manageable size for storage, manipulation, and display. Often, there is a need to respond to queries that can be answered accurately only from the base data, but that must be answered quickly from the summarized data. The task then is

[1]The research work described in this paper was motivated by exactly this problem in AT&T. There was interest in estimating daily totals for some data, which historically had been stored aggregated over months. The base data, while available, was several orders of magnitude more voluminous and therefore impractically expensive and time-consuming to handle. If reasonable guesses could quickly be made with respect to the daily totals, these were much preferred. The error could be estimated by computing over the full base data for selected sample aggregates.

to reconstruct as good an estimate of the original base data as possible. Applications of such a generic reconstruction method abound:

- Query optimization: DBMSs typically maintain histograms [15] reporting the number of tuples for selected attribute-value ranges. Queries may select only specific values, or select ranges that only partially overlap with the value ranges used in the histogram. Cost estimation for such queries will benefit from an accurate reconstruction of attribute-value occurrences for the queried value(-range). Similarly, range queries on multiple attributes will benefit from an accurate synthesis and extrapolation from the histograms of value distributions for individual attributes.

- Data warehousing [28]: The idea is that the central site will have meta-data, and condensed information (*e.g.*, summary data) from each participating site, which has detailed information. Accessing the remote site might be slow and/or expensive; a cheap, accurate estimate of the missing information is attractive.

- Transaction recording systems: A large enterprise (company, hospital *etc.*) has huge numbers of detailed records (sales transaction records, patient records *etc.*), which cannot be stored on-line. Thus, older records are either stored in tertiary storage, or discarded altogether. Saving summary data on-line, and providing a reconstruction algorithm, is an attractive alternative. This sort of technique is at the heart of the proposal in [17]. Managing such data well is a necessary prerequisite for effective data mining and decision support.

- Statistical databases [19], particularly in conjunction with the DataCube operator [10, 13]: For example, consider Census data with income levels, given as summary tables (=histograms), with one histogram for each of several attributes (age, years in school, years in present job, geographic location *etc.*). Again, the problem is to recover the detail information, or at least enough of it, so that we can answer combined queries on multiple attributes.

- Scientific databases: For example, consider LANDSAT images with vegetation data over time. Clouds sometimes obscure the view and hide relevant information. The problem is to recover the missing data, exploiting a-priori knowledge (*e.g.*, that vegetation data vary smoothly over space and time).

- Data integration: Two different databases often use different choices of attribute value ranges even for shared attributes. Merging such data requires that values be determined for the intersections of the respective ranges. This information is not directly available in either database and has to be reconstructed. For example, one state may store census data regarding income distribution over ranges 10000-20000, 20000-30000, 30000-40000, and so on. Another state may use a different set of ranges: 15000-25000, 25000-35000, and so on. A company targeting a promotion at some income section of the population may find it convenient to have a single union relation over the two states. Since data has been aggregated over incompatible ranges in the two base relations, such a union cannot easily be created.

In this paper, we show how to attack this reconstruction problem formally. We formulate this as an *inverse problem* (*cf.* [7]) so that we can draw upon the vast array of literature on this topic in the field of signal processing.

The paper is organized as follows. In Section 2 we present related work on query optimization and statistical databases. The mathematical problem formulation is given in Section 3. In Section 4 we present a brief introduction to the theory of inverse problems and some proposed solutions for database settings. In particular, our central Theorem regarding information recovery from aggregate data is established. In Section 5 we apply the proposed methods on real and realistic (synthetic) data, and report the improvements of our method over naive reconstructions. In Section 6 we present extensions of the basic technique to some additional scenarios. Our conclusions and future research directions are discussed in Section 7.

2 Survey

There is a large body of related work on query optimization, where the problem is to "guess" the attribute value distribution, to make selectivity estimates for specified queries. Early query optimizers used the uniformity assumption [23], which provably leads to pessimistic results [4]. Modern query optimizers typically use histograms [15]. The histogram of an attribute gives the count of records that fall into each pre-determined sub-range ("bucket") of the attribute range. DeWitt and Muralikrishna [20] examined combined histograms for multiple attributes. Ioannidis and Poosala [15] studied the trade-off between high prediction accuracy and ease of maintenance. Their recommendation was that histograms should maintain perfect information about selected attribute values, and assume the uniform distribution for the rest. A

recent, adaptive method, has been suggested by Chen and Roussopoulos [3]. The idea is to approximate the unknown value distribution with a polynomial, and to use query feedback to adjust the coefficients of the polynomial.

Similar approaches have been used for spatial databases: Theodoridis and Sellis [25] suggest a coarse discretization of the address space; for each grid cell, they use the average data density, and, making the uniformity assumption for each individual grid-cell, they estimate the performance of an R-tree.

Related work appeared in statistical databases: Malvestuto [19] examined the case of multiple summary tables, and developed algorithms to determine whether a given query can be evaluated to a single number, a range, or not at all. Ng and Ravishankar [21] also consider multiple summary tables, and propose a matrix-algebra criterion to choose the best combination of summary tables to answer a query.

Incomplete information has been studied extensively. For example, see [14] or [9]. The use of class structure, and other aggregation mechanisms, to store partial information has been presented in [16], and to respond to queries has been studied in [24]. All of these efforts have focussed on the logical nature of partial or missing information. In our paper, there is little qualitative reasoning; the emphasis is on effective numerical estimation.

Finally, there is much work on views with aggregates. For instance, [5] and [11] consider how to answer queries using aggregate views, and [12] shows how to maintain such views incrementally. Work along these lines hints at the importance of the problem we consider in this paper, but is not directly relevant to our concerns here.

3 Problem Formulation

The general problem is as follows: Consider a d-dimensional address space, discretized, and consider a function $\mathbf{x}$ on it: $\mathbf{x}[i_1, i_2, \ldots, i_d]$.

The question is: given some partial information about the values of $\mathbf{x}$ and general *a priori* information about the nature of distribution of $\mathbf{x}$ values, what is our best estimate for its value at each point.

Formally, the problem is as follows:

Problem 1 (General under-specified) *Estimate*

$$\mathbf{x}[i_1, i_2, \ldots, i_d] \quad i_j = 1, 2, \ldots \quad j = 1, 2, \ldots d \qquad (1)$$

under the constraints

$$C_k(\mathbf{x}) = 0 \quad k = 1, \ldots, n \qquad (2)$$

The problem is (typically) under-specified, with n being much smaller than the number of variables. We cannot obtain a unique solution unless we are willing to inject some additional knowledge. This additional knowledge comes in the form of *a priori* information regarding the nature of distribution of $\mathbf{x}$ values, and an error metric for the estimated solution. The problem to be solved then is to minimize this error metric, subject to the given constraints.

Nature of Constraints

The specific constraints can take many different forms, the solutions for most of which are fairly similar.

The simplest constraint is a summation constraint, where we require that the sum of specified $\mathbf{x}$ values be equal to some number. Most "rolled-up" data has this property, for instance, weekly sales totals are obtained as a summation of daily sales totals. Many histograms present counts, which are simple summations, such as the number of times a value within the specific range occurred. For example, the number of employees whose age is between 40 and 44 (inclusive) is the sum of the number of employees aged 40, 41, 42, 43 and 44 respectively.

The other commonly used constraint is an average. Thus, we may have the average temperature recorded for a week, obtained as the average of the average temperatures for each day in the week. Given the total number of $\mathbf{x}$ values being averaged over, converting a summation constraint to an average constraint simply involves a division by a constant.

Averages can sometimes be weighted. We may have average income for a region defined as the average of the average income for the constituent counties, weighted by their respective populations.

When a dimension is projected out, typically a summation (and sometimes an average) is performed on the dimension projected out. Thus, we could have a histogram for the number of employees in each age bracket and a separate histogram for the number of employees in each salary bracket. Each item in either marginal histogram represents a sum of the number of employees with that age (salary) and with each possible salary level (age).

Since all of the constraints described above are fundamentally similar in nature, and most can be transformed from one form to the other in a relatively straightforward manner, we choose to focus on a single well-defined problem for the bulk of this paper.

Also, for simplicity, we concentrate on the 1-d case. Issues with higher dimensions are considered in section 6.1. The matrix $\mathbf{x}$ becomes a vector $\vec{x}$, and the problem becomes:

Problem 2 (1-d under-specified) *Estimate the vector*

$$\vec{x} = [x_i] \quad i = 1, \ldots, N \qquad (3)$$

subject to the constraints

$$C_k(\vec{x}) = 0 \quad k = 1, \ldots, n \tag{4}$$

As a point of reference, consider a (time) sequence $\vec{x} = [x_i]$ $i = 1, \ldots, N$ (*e.g.*, count of occurrences of attribute value i or dollars spent on day i). Assume that it is hidden from us; instead, we are given the partial sums (*e.g.*, attribute value histograms or weekly sums) S_k, $k = 1, \ldots, n$, over contiguous and non-overlapping "batches". To further simplify the notation, in several places we will assume that the sequence is divided into "batches" of equal duration b (*e.g.*, weeks, with $b = 7$).[2]

The available information leads to the following problem formulation:

Problem 3 (Partial sums) *Estimate $\vec{x}$, given*

$$C_k(\vec{x}) = S_k - \sum_{i=B_{k-1}+1}^{B_k} x_i = 0 \qquad k = 1, \ldots, n \tag{5}$$

where B_k is the largest value of i included in the k^{th} batch. If all batches are of equal size b, then $B_k = b*k$.

The question is: given the above partial sums S_k ($k = 1, \cdots, n$), what is our best estimate for the ("daily") values x_i ($i = 1, \ldots, N$)?

4 Solution Technique

The aim is to minimize a suitable error metric between the estimate and the original vector. While the specific error metric used is not likely to be critical, for the sake of specificity we focus on the root-mean-squared error.

The theory of *inverse problems* [22] is applicable to the question at hand. Our specific case is typically under-constrained and thus ill-posed. Since the original vector is not known, we cannot use the root-mean-squared error as the objective function. We can force a unique solution by requiring minimization (or maximization) of some criterion ("functional") $\mathcal{F}(\vec{x})$, such as the entropy of the vector $\vec{x}$. Then, the problem is well defined:

Problem 4 (1-d Regularized) *Estimate $\vec{x}$ to minimize (maximize)*

$$\mathcal{F}(\vec{x})$$

under the constraints

$$C_k(\vec{x}) = 0 \quad k = 1, \ldots, n$$

[2]The up-coming "Linear Regularization" method applies even to *non-contiguous and/or overlapping and/or variable duration batches*. However, contiguous, non-overlapping, and equal duration batches appear most often in practice, and we have chosen to restrict ourselves to this case for the bulk of the paper, both to simplify the mathematical notation and to assist the reader in developing an intuition about the problem.

Under appropriate convexity and continuity conditions, the textbook method for solving both the minimization and the maximization version of such problems is the method of *Lagrange multipliers* [18]. The details are in a technical report [8]. The main question is how to choose the functional $\mathcal{F}()$. The objective is to minimize the expected value of the root-mean-squared error, given what we know a-priori about the distribution of values in the vector.

In the following subsections we describe two popular criteria, namely, *Maximum Entropy* and *Linear Regularization.*

4.1 Maximum Entropy (ME)

Maximum Entropy (*e.g.*, [22, sec. 18.7]) will introduce no additional constraints on the nature of the signal to be estimated. Recall that the entropy of a discrete probability distribution $\vec{p} = [p_1, \ldots, p_n]$ is given by

$$H(\vec{p}) = -\sum_i p_i \log p_i$$

The principle of Maximum Entropy suggests that, for an under-constrained problem, we could make it well-defined by requiring maximization of the entropy. If we know the grand total (sum) of the x_i's, we may assume that the x_i's are non-negative and normalized, so that they add to 1. Then, we have

Problem 5 (Partial sums with ME) *Maximize*

$$\mathcal{F}(\vec{x}) = -\sum_i x_i \log x_i$$

subject to the constraints

$$C_k(\vec{x}) \equiv (S_k - \sum_{i=B_{k-1}+1}^{B_k} x_i) = 0 \qquad k = 1, \ldots, n$$

We can show that the piece-wise constant curve, with $x_p = x_q$ for all p, q in the same "batch", is the solution to problem 5:

Lemma 4.1 *For Problem 5, the Maximum Entropy solution $\vec{x}$ is the piece-wise constant curve.*

Proof: Omitted, for brevity (see [8]). **QED**

4.2 Linear Regularization

In many situations, it is expected that there will only be a small difference between successive elements of the vector. Most population distributions, for large enough populations, would follow this principle. Thus, for instance, the distribution of employees across age may follow a "bell-shaped" curve with few very old or very young employees, and a relatively continuous

Symbol	Definition
N	total number of entries in the vector $\vec{x}$
n	number of constraints
b	batch size
Δx_i	$\equiv x_{i+1} - x_i$: forward difference operator
$\mathcal{F}(\vec{x})$	a functional of the vector $\vec{x}$
$\mathcal{H}(\vec{x})$	entropy function of the given vector ($=-\sum x_i \log x_i$)
$C_k(\vec{x})$	the k-th constraint on the vector $\vec{x}$
$\mathcal{Z}$	the set of signed integers (..., -1, 0, 1, ...)
ω_0	the highest frequency of a signal (in rads per second)

Table 1: Symbols and definitions

plateau in the middle. We would be surprised if some large company had many 34 year old and 36 year old employees, but very few 35 year old employees, for example.

In such situations, one can require that the solution $\vec{x} = [x_i]$ be smooth by minimizing the functional

$$\mathcal{F}(\vec{x}) = \sum_{i=1}^{N-1} (x_i - x_{i+1})^2 \qquad (6)$$

Intuitively, the above functional expresses our belief that the unknown solution $\vec{x}$ is rather smooth; thus, the functional penalizes large squared values for the forward differences $\Delta x_i = x_{i+1} - x_i$. Therefore, the problem becomes: Minimize Eq. 6, subject to the conditions of Eq. 5. The functional of Eq. 6 results in an instance of so-called Linear Regularization (or '*Phillips-Twomey method*', or '*constrained linear inversion method*' or '*Tikhonov-Miller regularization*' [22]). In the full paper [8] we show that this minimization problem leads to a matrix algebra problem, using Lagrange multipliers.

4.2.1 Computational Effort

The matrix inversion problem mentioned above involves a square matrix, with side $M = N + n$. In general, matrix inversion has complexity $O(M^3)$. This may be prohibitive, particularly since N may often be large.

However, in our case, the matrix is of a special form: it is singly-bordered tri-diagonal, and the border itself is block-diagonal. In this case, matrix inversion has complexity $O(M)$ [22, p. 72], that is *linear* on the matrix side. Since the length of the unknown distribution, N, is significantly greater than the number of batches/constraints n, for all practical purposes the inversion effort is $O(N)$. This is optimal: no method can achieve less than $O(N)$ complexity, since it needs $O(N)$ steps just to print the solution vector.

4.2.2 Full recovery for smooth signals

A major result in this paper is that we can achieve *full recovery* of information from the summarized data, if the original data is "sufficiently smooth". More specifically, we have the following theorem (stated informally at first):

> Consider a "slowly varying" discrete-time signal that consists solely of sinusoidal components of period greater than or equal to some T_0. This signal can be reconstructed perfectly from sole knowledge of its contiguous non-overlapping partial sums taken over $T_0/2$ samples at a time (or shorter). Thus, this signal can be recovered fully from an appropriately coarse histogram.

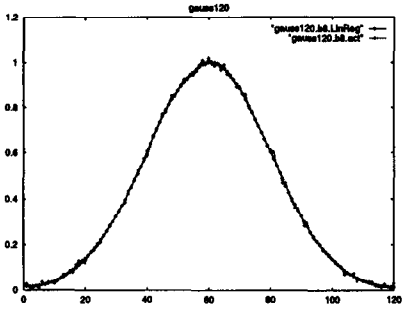
Formally, we have the following theorem, where ω_0 denotes the frequency that corresponds to the period T_0, $X(e^{j\omega})$ denotes the Discrete-Time Fourier Transform (DTFT) of the signal $\vec{x}$ and $\mathcal{Z}$ is the set of (signed) integers:

Theorem 4.2 (Band limited reconstruction from contiguous non-overlapping partial sums) *Consider a discrete-time signal* $\{x(i)\}_{i\in\mathcal{Z}}$. *Assume that its Discrete-Time Fourier Transform (DTFT)* $X(e^{j\omega})$ *converges, and* $X(e^{j\omega}) = 0$, $\frac{\pi}{b} \leq |\omega| \leq \pi$. *This signal can be recovered from its contiguous non-overlapping partial sums* $\{S_k\}_{k\in\mathcal{Z}}$, $S_k = \sum_{i=b(k-1)+1}^{kb} x(i)$, $\forall k \in \mathcal{Z}$

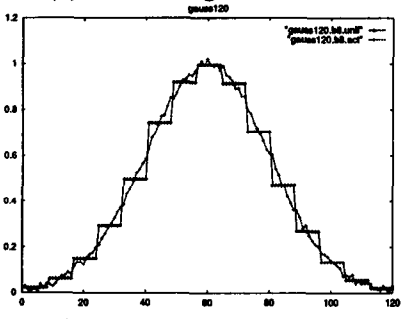
Proof: Omitted for brevity (see [8]). **QED**

Our Theorem guarantees full recovery when its conditions are met, and its proof is *constructive*, *i.e.*, it specifies a filter that achieves full recovery. However, this means of recovery might impractical, or the conditions of the Theorem might not be exactly satisfied. In this case, Linear Regularization is the next best approach, as we discuss in [8]. In addition to achieving near-optimal reconstruction, Linear Regularization has a number of important advantages: (a) it works

even for overlapping/variable size batches and/or missing summaries, and (b) it has low computational complexity, namely, linear on the number of unknowns N, as discussed in subsection 4.2.1. Notice that interpolation methods (like polynomial and spline interpolation) are *not* applicable in our case: They expect a "decimated" signal (e.g., x_b, x_{2b} ...) as their input, as opposed to the partial sums that we currently have. Notice that Linear Regularization is very general, and it can work even in the case of "decimated" signals, without even needing any user-determined constants (like the degree of the interpolating polynomial). See Section 6 for a more detailed discussion of additional applications of Linear Regularization.



(a) Linear Regularization



(b) Maximum Entropy

Figure 1: Reconstruction of a Gaussian distribution: with Linear Regularization, we obtain almost error-free reconstruction. Detailed base data: dashed line with "+". Reconstruction: solid line with "diamonds". Batch size b=8.

Thus, for smooth curves, Linear Regularization indeed creates an essentially error-free reconstruction. Fig. 1 (a)-(b) shows Linear Regularization and Maximum Entropy, respectively, applied to an approximately Gaussian distribution (more details on this and other datasets are provided in the experiments section). The batch size is $b = 8$. This is a very smooth dataset. Linear Regularization provides a visibly better reconstruction than Maximum Entropy.

5 Experiments

We ran several experiments to evaluate our approach. We used both the Maximum Entropy method and the Linear Regularization method. The measure of success was the normalized root-mean-square error (RMS), which is a typical measure for forecasting in time series [27]. Specifically, we define:

$$RMS = \left(1/N \sum_{i=1}^{N} (x_i - x_{actual,i})^2 \right)^{1/2} \quad (7)$$

where x_i is the reconstructed value and $x_{actual,i}$ is the actual value at time i.

We ran our experiments on a number of real and synthetic datasets, namely:

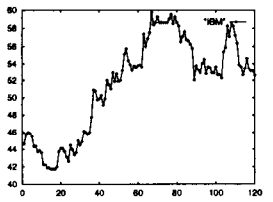
- **'GAUSS'** dataset (synthetic): this dataset has been estimated by drawing samples from a Gaussian distribution and counting the number of samples falling within a given histogram bin. We used N=120 bins. Attribute values, *e.g.*, patient height, patient weight *etc.*, are often distributed as a Gaussian, or some close variant thereof. To the extent that histograms are the typical means of storing attribute value data, this case is typical of the sort of situation in which one can expect the work in this paper to be of value. To make our experiment more realistic, rather than use a perfect Gaussian, we created an "approximate" Gaussian, of the sort one would expect from 20,000 items distributed according to a Gaussian distribution. Thus, the number of values in each bin is a little off from the ideal theoretical value. Furthermore, we normalized the data set to lie between 0 and 1 by dividing throughout with the peak value. This is the example used in the previous section; see Figure 1.

- **'SINE'** dataset (synthetic): a sinusoid, with N=120 samples: $x_i = \sin(2\pi i/60)$ $i = 0, \ldots, 119$. This is a very smooth dataset.

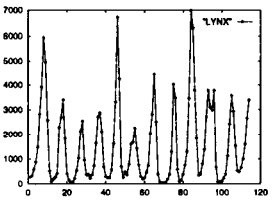
- **'IBM'** dataset (real): IBM closing prices, from `http://www.ai.mit.edu/ stocks.html`. The dataset starts from Aug. 30, 1993, and spans 120 working days. See Figure 2(a).

- **'LYNX'** dataset (real): Canadian lynx trappings data per year, 1821-1934, for a total of N=114 samples. This is a well known dataset in population biology - it can be found in any time-sequence book (*e.g.*, [2]), as well as on-line through the "S" statistical package [1]. Notice that it has a periodicity of 9-10 years. However, it is not very smooth: it has abrupt population explosions, with

significantly different peak values each time. See Figure 2(b).



(a) "IBM"



(b) "LYNX"

Figure 2: Plots of the two real datasets.

The experiments were designed to answer the following questions:

1. How good is the reconstruction when we use Linear Regularization and Maximum Entropy, and how does the "smoothness" constraint of Linear Regularization perform against the uniformity assumption (Maximum Entropy), for smooth and "rugged" data?

2. How does the accuracy of reconstruction depend on the length of the "batch"?

5.1 Accuracy

We try several values of the batch size b, and we let each method recover the original dataset. Table 2 shows the RMS error for the competing methods, for all the datasets.

Linear Regularization consistently outperforms the "uniform" method, as long as the batch size b obeys Theorem 4.2 (that is, it is less than the half-period $T_0/2$ of the shortest cycle of the signal). It is a pleasant surprise that the Linear Regularization does better even for real datasets, like the '**IBM**' dataset, which is not as smooth as the two synthetic ones.

The relative gains increase with the smoothness of the target sequence, as intuitively expected: the two synthetic, smooth datasets enjoy the best savings (up to 89%), followed by the '**LYNX**' dataset (up to 35% savings - notice that the dataset is somehow periodic), followed by the '**IBM**' dataset, the most 'rugged' of all (savings: up to 21%).

5.2 Dependency on batch size

Figures 3(a)-(b) and 4(a)-(b) plot the RMS error as a function of the batch size b, for the synthetic and real datasets, respectively. This is the same information as in Table 2, in pictorial form. Notice that Linear Regularization does consistently better than the uniformity/ME assumption, as long as the batch size b obeys our Theorem. The cross-over point for the '**LYNX**' dataset is at $b = 5$, as expected, since the half-period of the major oscillation is 9.5/2=4.25. Notice that

method	Lin. Reg.		ME
dataset	RMS	(rel. % sav. over ME)	RMS
'**SINE**' b= 2	0.004	89%	0.037
'**SINE**' b= 4	0.012	84%	0.082
'**SINE**' b= 6	0.024	80%	0.125
'**GAUSS**' b= 2	0.007	38%	0.012
'**GAUSS**' b= 4	0.009	59%	0.023
'**GAUSS**' b= 6	0.009	71%	0.034
'**GAUSS**' b= 8	0.010	77%	0.045
'**LYNX**' b= 2	387	35%	599
'**LYNX**' b= 3	676	26%	926
'**LYNX**' b= 4	927	19%	1148
'**LYNX**' b= 5	1229	0%	1239
'**LYNX**' b= 6	1442	-8%	1325
'**IBM**' b= 2	0.464	8%	0.507
'**IBM**' b= 4	0.664	13%	0.771
'**IBM**' b= 6	0.908	14%	1.059
'**IBM**' b= 8	0.891	17%	1.076
'**IBM**' b= 10	1.093	21%	1.396

Table 2: RMS errors for each method, and relative savings with respect to the 'uniform=ME'. Batch size b, as specified.

for the '**IBM**', '**GAUSS**', and '**SINE**' datasets, Linear Regularization is the consistent winner for a wide range of the b values, because these signals have most of their energy concentrated in low frequencies.

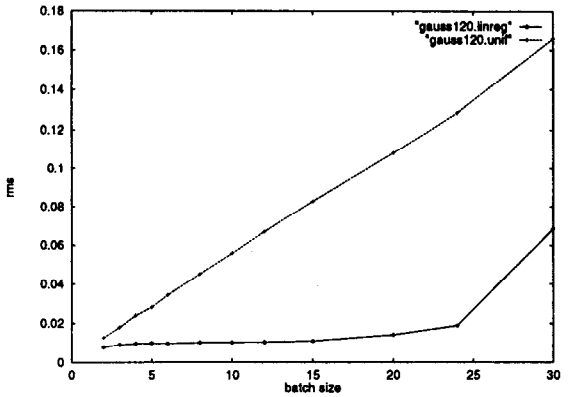
6 Extensions - Discussion

We have presented the theory of inverse problems, and we have shown how its special case, the Linear Regularization, can give better reconstruction from (one-dimensional) summary data. In this section we list some additional database applications of our approach.

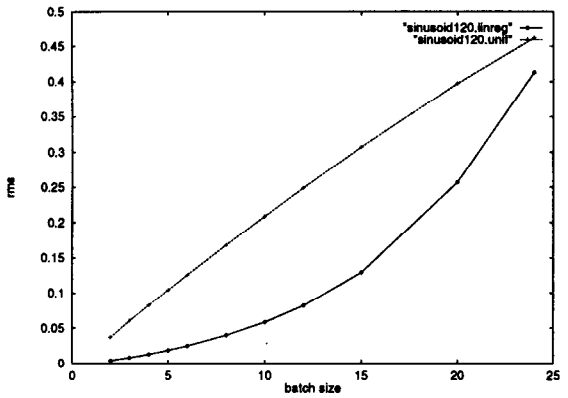
6.1 Merging Histograms and OLAP.

The theory of inverse problems can handle 1-d, 2-d and even higher dimensionality address spaces. We have focused mainly on 1-d signals (= time sequences), for two reasons: (a) they lead to a more clear description of the approach and (b) they are very interesting in their own right (sales patterns, stock prices, *etc.*) [2, 27]. However, the reduction in data size becomes particularly emphatic as the number of dimensions is increased, and the techniques presented in this paper become even more important.

Here we show how we could handle higher dimensionalities. Consider the case of a relation with two attributes, such as, *e.g.*, `employees`, with `age` and `salary`. Suppose that we are given one histogram

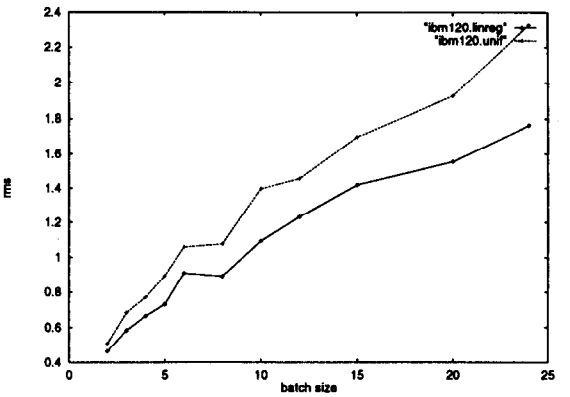


(a) **'GAUSS'**

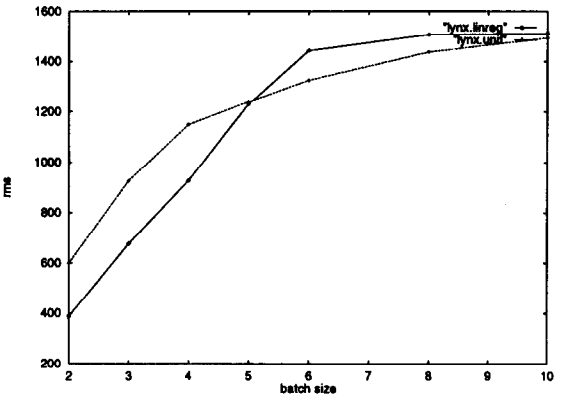


(b) **'SINE'**

Figure 3: RMS error vs batch size b, for each of our synthetic datasets. Maximum Entropy: dashed line with "+". Linear Regularization: solid line with "diamonds"



(a) **'IBM'**



(b) **'LYNX'**

Figure 4: RMS error vs batch size b, for each of our real datasets. Maximum Entropy: dashed line with "+". Linear Regularization: solid line with "diamonds"

for **age** and another for **salary**, each divided into 3 buckets (ranges). Table 3 illustrates the situation

$S_{3,*}$	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$
$S_{2,*}$	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$
$S_{1,*}$	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
.	$S_{*,1}$	$S_{*,2}$	$S_{*,3}$

Table 3: Illustration of the 2-d case: Recovery of detail data from two 1-d histograms.

Thus, we are given the histograms $S_{*,j}$ and $S_{i,*}$ (which correspond to the marginal distributions) and we want to recover the "hidden" values of $x_{i,j}$. The problem is formulated as follows: Given

$$S_{i,*} = \sum_j x_{i,j} \quad i = 1,2,3 \tag{8}$$

$$S_{*,j} = \sum_i x_{i,j} \quad j = 1,2,3 \tag{9}$$

minimize the functional $\mathcal{F}()$ of choice. Once again, Maximum Entropy corresponds to the independence assumption, that is, if N_{emp} is the total count of employee records, ME leads to

$$x_{i,j} = S_{i,*} * S_{*,j}/N_{emp} \quad \forall i,j \tag{10}$$

More formally:

Lemma 6.1 *In the 2-d problem above, the Maximum Entropy solution leads to the independence assumption*

Proof: Using the Lagrange multipliers and solving for $x_{i,j}$. See [8] for details. **QED**

However, if the 2-d joint distribution is smooth, we should do better with Linear Regularization. Specifically, we require that the sum of squares of forward differences (both horizontally and vertically) be minimized:

$$\mathcal{F}(\vec{x}) = \sum_{i,j}(x_{i,j} - x_{i+1,j})^2 + \sum_{i,j}(x_{i,j} - x_{i,j+1})^2 \tag{11}$$

The Lagrange equations will be linear, and the resulting system can be solved exactly, with a matrix inversion package.

Aside from the database context, 2-d estimation is of use in image processing applications as well. For instance, it is well-known that if an image is "zoomed", say each pixel is replaced by four pixels, that the resulting higher resolution image will be "grainy". In image processing, a local smoothing function is typically used to improve the zoomed image. Linear Regularization can be used to obtain exactly the same effect [6, 26].

6.2 Data Warehousing

It should be clear that the summation constraints $C_k(\vec{x})$ may be arbitrary. Our proposed approach can also handle *overlapping* intervals, as well as intervals of *variable length*. This is especially suitable in the case that we have to merge information from several sources, as in multi-databases and data warehousing [28]. For example, suppose that one source provides weekly (non-overlapping) sums, a second source provides the exact values for some selected days, and a third source provides monthly sums (where month boundaries do not usually coincide with week boundaries). The question is to find the best estimates for the daily values x_i. The problem can easily be formulated:

$$\min_{\vec{x}}(\mathcal{F}(\vec{x})) \tag{12}$$

under the constraints

$$C_{weekly,k}(\vec{x}) = 0 \quad k = 1, \ldots \tag{13}$$

$$C_{daily,j}(\vec{x}) = 0 \quad j = 1, \ldots \tag{14}$$

$$C_{monthly,m}(\vec{x}) = 0 \quad m = 1, \ldots \tag{15}$$

where $\mathcal{F}(\vec{x})$ is a suitable functional, *e.g.*, the Linear Regularization functional.

6.3 Reconstruction of missing values

Also notice that the proposed Linear Regularization approach can handle not only variable length and/or overlapping intervals, but also *missing sums and/or values*, even when the grand total is unknown. Linear Regularization will use the known sums (and/or values), and it will fill-in the missing values, to furnish a smooth curve. Notice that, without a given grand total, Maximum Entropy can not be used *at all* in this case.

7 Conclusions

The main contribution of this work is a formal approach to the recovery of information from summary data, and, more generally, arbitrary, partial data in the form of constraints. The idea is to use the machinery of the well-developed "inverse problem theory", to inject a-priori knowledge about the domain, eventually transforming the problem into a constrained optimization problem.
Additional contributions are

- Theorem 4.2, which shows that for "smooth" enough distributions, it is possible to have full recovery of information, given partial sums.
- Lemmas 4.1, 6.1, showing that the theory of inverse problems includes the traditional uniformity and independence assumptions as special cases, when the Entropy is used as the cost function.
- A conceptual basis for selecting Linear Regularization as the technique of choice to obtain a smooth reconstruction. Further, the use of an existing numerical analysis technique that gives the solution for Linear Regularization in linear time $O(N)$.
- Experiments showing that, under the conditions of Theorem 4.2, Linear Regularization consistently outperforms the Maximum Entropy/uniformity assumption, not only for smooth data, but for "fractal", real data as well (IBM stock price movements, and the lynx trappings dataset).

Future work could examine further ties with the well developed field of inverse problems and image restoration. The interaction between two types of summaries, marginal summations and batching summations, is important for multi-dimensional reconstruction (OLAP), histogram maintenance in query optimization, compression of real distributions, and numerous other database applications.

Acknowledgments

We thank John Goutsias for his help with Lemma 4.1, and I. S. Mumick for providing feedback on a draft of this paper.

References

[1] Richard A. Becker, John M. Chambers, and Allan R. Wilks. *The New S Language.* Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA, 1988.

[2] George E.P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control.* Prentice Hall, Englewood Cliffs, NJ, 1994. 3rd Edition.

[3] Chungmin M. Chen and Nick Roussopoulos. Adaptive selectivity estimation using query feedback. *Proc. of the ACM-SIGMOD*, pages 161–172, May 1994.

[4] S. Christodoulakis. Implication of certain assumptions in data base performance evaluation. *ACM TODS*, June 1984.

[5] Shaul Dar, H.V. Jagadish, Alon Y. Levy, and Divesh Srivastava. Answering SQL queries with aggregation using views. Technical report, AT&T, 1995.

[6] D. E. Dudgeon and R. M. Mersereau. *Multi-Dimensional Digital Signal Processing.* Prentice-Hall, 1984.

[7] Heinz W. Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems.* Kluwer, Dordrecht, 1996.

[8] Christos Faloutsos, H.V. Jagadish, and Nikolaos Sidiropoulos. Information recovery from partial data. Technical Report ISR-TR-97-7, Inst. for Systems Research, Univ. of Maryland, College Park, MD, 1997. Also available at ftp://olympos.cs.umd.edu/ pub/TechReports/ vldb97.ps.

[9] Georg Gottlob and Roberto Zicari. Closed world database opened through null values. In *Proc. 14th Int'l Conf. on Very Large Databases*, pages 50–61, 1988.

[10] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: a relational aggregation operator generalizing group-by, cross-tab, and subtotals. Technical Report No. MSR-TR-95-22, Microsoft, 1995.

[11] Ashish Gupta, Venkatesh Harinarayan, and Dallan Quass. Generalized projections: A powerful approach to aggregation. In *Proc. 21st International Conference on VLDB*, Zurich, Switzerland, September 1995.

[12] Ashish Gupta, Inderpal Singh Mumick, and V. S. Subrahmanian. Maintaining views incrementally. In *Proc. of ACM SIGMOD*, Washington, D.C., May 1993.

[13] Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman. Implementing data cubes efficiently. In *Proc. ACM SIGMOD*, pages 205–216, Montreal, Canada, May 1996.

[14] T. Imielinski and W. Lipski. Incomplete information in relational databases. *JACM*, 31(4), October 1984.

[15] Yannis E. Ioannidis and Viswanath Poosala. Balancing histogram optimality and practicality for query result size estimation. *ACM SIGMOD*, pages 233–244, June 1995.

[16] H. V. Jagadish. The incinerate data model. *ACM TODS*, 20(1):71–110, March 1995.

[17] H. V. Jagadish, Inderpal Singh Mumick, and Avi Silberschatz. View maintenance issues in the chronicle data model. In *Proc. ACM PODS*, pages 113–124, 1995.

[18] D. G. Luenberger. *Introduction to Linear and Nonlinear Programming.* Addison-Wesley, Reading, Massachusetts, 1973.

[19] Francesco M. Malvestuto. A universal-scheme approach to statistical databases containing homgeneous summary tables. *ACM TODS*, 18(4):678–708, December 1993.

[20] M. Muralikrishna and David J. DeWitt. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries. *Proc. ACM SIGMOD*, pages 28–36, June 1988.

[21] Wee-Keong Ng and Chinya V. Ravishankar. Information synthesis in statistical databases. *Proc. CIKM*, pages 355–361, November 1995.

[22] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992. 2nd Edition.

[23] P.G. Selinger, D.D. Astrahan, R.A. Chamberlain, R.A. Lorie, and T.G. Price. Access path selection in a relational database management system. *Proc. ACM-SIGMOD*, pages 23–34, 1979.

[24] Chung-Dak Shum and Richard Muntz. An information-theoretic study on aggregate responses. *Proc. of VLDB*, pages 479–490, August 1988.

[25] Yannis Theodoridis and Timos Sellis. A model for the prediction of r-tree performance. *Proc. of ACM PODS*, 1996.

[26] P. P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice-Hall, 1993.

[27] Andreas S. Weigend and Neil A. Gerschenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison Wesley, 1994.

[28] Jennifer Widom. Research problems in data warehousing. *CIKM*, November 1995. Invited paper.