December 1, 2003 TR CS-03-29

Linear Sketches for Approximate Aggregate Range Queries[1,2]

Vasundhara Puttagunta and Konstantinos Kalpakis

Computer Science and Electrical Engineering Department
University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, Maryland 21250

[1]Email: {vputta1,kalpakis}@csee.umbc.edu
[2]Supported in part by NASA under Cooperative Agreement NCC5–315.

Linear Sketches for Approximate Aggregate Range Queries[1,2]

Vasundhara Puttagunta and Konstantinos Kalpakis

Abstract

Answering aggregate queries approximately over multidimensional data is an important problem that arises naturally in many applications. An approach to the problem is to maintain a succinct (i.e. $O(k)$ space) representation, called *sketch*, of the frequency distribution h of the data, and use $\hat{h}$ for answering queries. Common sketches are constructed via linear mappings of h onto a k–dimensional space, e.g. map h to its top–k Fourier/Wavelet coefficients. We call such sketches linear sketches, since $\hat{h} = P^*h$ for some sketching matrix P. Linear sketches have the benefit that they can be easily maintained incrementally over data streams. Sketches are typically optimized for approximating the data distribution, but not the answers to queries.

In this paper, we are concerned with linear sketches that approximate well not only the data but also the answers to the aggregate queries. The quality of approximations is measured using the mean squared and relative errors (MSE and RLE). A query is represented by a column vector q such that its answer is $q^T h$. A given set of queries can be represented by an appropriate query matrix Q.

We show that the MSE for the queries is minimized when the sketching matrix used to construct a linear sketch of h has as columns the top-k eigenvectors of the query matrix Q. Further, if the query matrix Q corresponds to all range queries of a given extent, then Q has a succinct representation and a universal set of eigenvectors. For the 1–dimensional case, these eigenvectors are precisely the vectors in the Discrete Fourier Transform. Hence, these eigenvectors have a succinct representation. Generalizations to higher dimensions are also given. Because of this succinct representation it is particularly advantageous for maintaining sketches over streaming data.

Further, in many instances, there could already be a (linear) sketch of a distribution, maintained over the data for various applications. We show how to extend that sketch so that the MSE for a given set of queries is minimized. This provides a novel method to construct sketches that consider both the data as well as the queries. Using both synthetic and real data, we experimentally demonstrate that our approach delivers significantly smaller errors than various other standard approaches.

Keywords: Linear Sketching, Approximate query answering, Circulants, Fourier Vectors.

1 Introduction

The past decade has seen an explosive growth in the amount of data being generated, collected and communicated. Most data is of little use unless it is analyzed and understood. Therefore it is not surprising that current day applications are required to deal with massive amounts of data. Consider the Internet for example. It is estimated [13] that the Internet backbone traffic in the U.S was of the order of 10^{16} bits per day in the December of 2002. In order to deal with transmitting/transferring this kind of data, network management applications have to perform numerous tasks such as capacity planning, band–width allocation, congestion control, fault tolerance and the like. Such applications rely heavily on monitoring the network which includes (a) continuously collecting massive amounts of data about the network itself (for example,

[1]Supported in part by NASA under Cooperative Agreement NCC5–315.

[2]Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250. E-mail: {vputta1,kalpakis}@csee.umbc.edu. Phone: 410-455-3143. Fax: 410-455-3969. **Please send all correspondence to Prof. Kalpakis.**

number of packets received, number packets lost, packet delay, etc) and (b) analyzing this data. Such data is often multidimensional and due to the way it is continuously arriving, one can naturally view it as a data stream. The rates at which such data is generated coupled with the fact that data streams could be potentially unbounded means that it is impractical to store the data or even a fraction of the data for the purpose of data analysis or other computations. This opens up a challenging avenue of stream computations. (See Babu and Widom [3] for issues with data streams).

One class of queries that are very useful in data analysis and monitoring applications is aggregate range queries over multidimensional data. By a range query, we mean an aggregate query over a rectangular window (a hyper–rectangle specified by an interval along each dimension). It is usually not necessary to answer these queries exactly. However, it may be critical to answer them fast or almost instantly. We shall call such queries "approximate range queries". Answering approximate range queries effectively and efficiently over multi–dimensional data (streams) is an important problem and has been the focus of many studies in database research. It is also the point of interest in this paper. Maintaining the data distribution of the stream will enable us to answer range queries accurately. Therefore one approach to the problem is to be able to dynamically maintain a synopses data structure which is a succinct representation of the data stream distribution. One class of widely used synopses data structures for this purpose is that of Histograms. If we consider the data distribution to be a vector h, a linear sketch of h is a typically a vector much smaller in size, that is simply a linear transformation of the vector h using a matrix P. Linear sketches have the benefit that they can be easily maintained incrementally over data streams. We refer to P as the sketching matrix. Linear sketches have been found to be succinct representations of the data distribution. Different sketching matrices have been proposed and shown to be useful in extracting approximate histograms and answering range queries. Thaper et al [15] use linear sketches with random projection matrix as the sketching matrix and Guha et al [11] use linear sketches with vectors from the wavelet basis corresponding to the top–k wavelet coefficients of the data distribution as their sketching matrix.

Note that both the sketch based approaches mentioned above, have two parts. First they maintain a linear sketch of the data incrementally over the stream. In [11] there is also the issue of maintaining the top–k. Then there is a *reconstruction* algorithm to extract an approximate histogram which is then used to answer range queries. However, it is possible to answer approximate range queries directly from the linear sketches and avoid the reconstruction phase altogether. This has been explored by Gilbert et. al. [9]. Further, note that the sketches computed in both the above approaches depend only on the data distribution. They completely ignore the queries. Often, we have some information about the queries. For example, there may be a region of interest that is most often queried. There may be continuous queries which can be viewed as the same query being posed at all times. This is especially important when dealing with streaming data. Also statistics such as average query extent, etc. can be observed. It is beneficial to make use of such information if it means getting better results. It is also known from previous studies (see Bruno et al [4] for example) that it is be beneficial to take queries into consideration while constructing histograms and similar synopses data structures. Aboulnaga and Chaudhuri [2] in fact construct *histograms without looking at data*, by tuning them based on errors on queries. From these observations, a natural question that arises is *what sketching matrix should be used in order to directly answer (a set of) approximate range queries effectively and efficiently?*

Our Contributions

In this paper we present our study on linear sketches that approximate well the answers to aggregate range queries. The quality of approximations is measured using the mean of squared errors (MSE) as well as mean relative errors (RLE). A query is represented by a column vector q such that its answer is $q^T h$. A given set of queries can be represented by an appropriate query matrix Q. Our contributions include the following:

(a) For a given set of queries Q, we show that the MSE is minimized when the sketching matrix used to

construct a linear sketch of h has as columns the top-k eigenvectors of the matrix QQ^*. This requires $O(k)$ time and space to update the sketch, and $O(kN)$ space to maintain the sketching matrix P if it does not have a succinct representation and needs to be stored explicitly.

While the space requirement for storing the sketching matrix may be prohibitive for many applications it should be noted that storing P is just for the lookup. Computing the sketch or an update to the sketch takes only $O(k)$ time and space. Further, if the sketch has to be communicated, the communication cost is just $O(k)$. Therefore, in applications where communication cost is very expensive (for example in the wireless domain), this approach is useful.

(b) Further, if Q corresponds to all the range queries of a given extent, then Q has a succinct representation and a universal set of eigenvectors. For the 1–dimensional case, these eigenvectors are precisely the vectors in the Discrete Fourier Transform, and thus they do not need to be stored explicitly. This result generalizes to higher dimensions as well with the use of Kronecker products. Because of the succinct representation, such sketches for such queries are advantageous for streaming data and can be maintained using $O(k)$ space and time per update where k is the size of the sketch.

(c) We further show how to extend any linear sketch so that the MSE for a given set of queries is minimized. Most approaches for approximate range query answering either look only at the data while constructing the sketch or look only at the queries. Our method of extending linear sketches provides a novel approach for constructing sketches that make use of both the data as well as the queries. (Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT) Random Projections and Histograms of the data distributions are some classical linear sketches.)

(d) Even though the sketching matrix used to compute the linear sketch is optimized for a pre–defined set of queries, other queries can be represented as a linear combination of these pre–defined set of queries and can be answered approximately using the same sketch. Also, several applications (particularly monitoring and stream applications) have continuous queries. Unlike traditional queries where queries are answered once according to the snapshot of the data at the given time and forgotten, continuous queries require continuous estimates for the same query continuously over time. Therefore, optimizing the sketching matrix for a given set of queries in indeed important for such applications.

(e) We perform experiments with both synthetic and real datasets and compare our methods with standard techniques of DFT, DWT and Random Projections. We experimentally demonstrate that our approach indeed achieves significantly smaller errors (MSE and RLE) for counting range queries.

Organization

The rest of the paper is organized as follows. For the convinience of the reader we give overview of concepts from matrix analysis used in this paper in section 3. In section 4, we present our approach for constructing sketching matrices. In section 5, we discuss the use of circular range queries for constructing small foot–print sketching matrices. Results of the experimental evaluation are given in section 6, and conclusions are given in section 7.

2 Previous Work

The problem of answering range queries effectively and efficiently has been studied extensively by the database community for a long time. Traditional methods concentrate on giving exact answers. To this end, several indexing methods were proposed for multi–dimensional data (see Gaede and Gunther [7] for a survey on multidimensional access methods). Here, the idea is to be able to retrieve only data that falls

within the query–range quickly, and compute the aggregate over the data incrementally as they are retrieved using iterators (see the classical survey by Graefe [10] for query evaluation techniques).

On–line Analytical Processing (OLAP) applications demand very fast approximate aggregate query answering techniques. These applications fanned development of several techniques for multidimensional aggregate queries. For example, Vitter et al. [16] present a technique based on multi–dimensional wavelet transform of the data distribution to compute small space data–cubes that are effective in answering aggregate range queries approximately. Similarly, fast approximate query estimating techniques are used for query optimization in database systems. Such techniques work under the following premise: avoid having to deal with the real data as much as possible, by maintaining some *synopses* data structures. These data structures are constructed in one (or a few) passes over the data, and used for getting quick query estimates when required. Gibbons and Matias [8] present several synopses data structures that are useful for computing aggregate queries. One class of synopses data structures that has been studied extensively is the histogram. Poosala [14] presents several categories of histograms and approximate query estimation techniques using histograms in database systems.

With the advent of stream computations, these techniques come in very handy, particularly if the synopses data structures can be maintained in one pass (incrementally). This is because we are allowed to see the data only once as the data stream by and further, data streams can be potentially unbounded and streaming at a tremendous rate. In other words, such a data structure should have the following two properties: (a) it should have a small memory foot–print, and (b) it should have a small update time, typically $O(1)$ per data item, while using small memory. It should be updatable under insertions and possibly deletions and updates in the data stream.

For the purpose of this paper, we want to answer range queries approximately over streaming data. Maintaining the data distribution of the stream will enable us to answer range queries accurately. Therefore, the problem in this case is to be able to dynamically maintain a synopses data structure which is a succinct representation of the data distribution of the stream. Several recent papers on computations over streams have the following two–step approach (a) there is a *sketching* algorithm that maintains a small data–structure (called *sketch*) incrementally from the stream; and (b) there is a *reconstruction* algorithm that performs some computation over the sketch to produce the required output. Thaper et al [15] propose maintaining a small footprint *sketch* over the multidimensional stream that is essentially a random projection of the data distribution. They give algorithms to extract a *b–bucket Histogram* from the sketch on demand. Such a histogram could then be used to answer range queries. Guha et al [11] use the top–k wavelet coefficients of the frequency distribution of the data as their sketch and present efficient algorithms to maintain it. They present fast algorithms to extract *b–bucket Histogram* from the top–k wavelet coefficients. They give a deterministic algorithm that approximates the optimal histogram by a histogram whose norm is within a factor of ϵ using total space of $B(\log(N)\log\|h\|^2/\epsilon)^{O(1)}$ and time of $O((B\log(N)\log\|h\|^2/\epsilon)^{O(1)})$.

3 Overview of Matrix Analysis

We introduce various basic concepts and notations. Further, for brevity, we provide a quick overview of concepts and facts from matrix analysis that we will be using in subsequent sections. The interested reader is referred to an appropriate reference on matrix analysis, such as Lancaster and Tismenetsky [12]. The proofs of Propositions and Theorems in this section, that are stated without proofs, can be found in [12]. Unless stated otherwise, we are concerned with matrices and vectors over the field of complex numbers $\mathcal{C}$.

Definition 1 *Let A be a matrix over $\mathcal{C}$. The complex conjugate of A is denoted by $\overline{A}$. The conjugate transpose A^* of A is defined as $A^* = \overline{A}^T$. Matrix A is Hermitian if and only if $A^* = A$. Matrix A is idempotent if and only if $A^2 = A$. Matrix A is unitary if and only if $AA^* = A^*A = I$, i.e. $A^{-1} = A^*$.*

Definition 2 (Standard Inner Product)
For any two complex vectors $x, y \in \mathcal{C}^n$, the standard inner product is the scalar given by

$$\langle x, y \rangle = y^* x. \tag{1}$$

The norm of a vector $x \in \mathcal{C}^n$ is given by $\|x\|^2 = \langle x, x \rangle = x^* x$.

Proposition 1 *(Orthonomalization)*
Let $S = \{x_1, x_2, \ldots, x_m\}$ be a set of m vectors in $\mathcal{C}^n$. The span of S is the subspace of $\mathcal{C}^n$ spanned by the vectors in S. Let k be the dimension of the span of S. Given S, one can always find, via the Gram–Schmidt orthonormalization algorithm, a set with k orthonormal vectors with the same span as S. A set S of orthonormal vectors is called an orthonormal set of vectors.

Definition 3 (Bi–Orthonormal System)
Let S_1 and S_2 be two orthonormal sets of vectors. The sets S_1 and S_2 form a bi–orthonormal system if and only if $\langle x, y \rangle = \langle y, x \rangle = 0$ for all $x \in S_1$ and $y \in S_2$.

Definition 4 *(Column Subspace and Complementary Orthogonal Subspace)*
Let A be an $n \times m$ complex matrix. The column space $\mathcal{S}_{(A)}$ of A is the subspace of $\mathcal{C}^n$ spanned by the columns of A. The complementary orthogonal subspace of $\mathcal{S}_{(A)}$ is denoted by $\mathcal{S}_{(A)}^{\perp}$ [3]. Each vector in $\mathcal{S}_{(A)}^{\perp}$ is orthogonal to all vectors in $\mathcal{S}_{(A)}$, and any vector $x \in \mathcal{C}^n$ can be writtern as $x = x_1 + x_2$ where $x_1 \in \mathcal{S}_{(A)}$ and $x_2 \in \mathcal{S}_{(A)}^{\perp}$.

Proposition 2 (Complex Roots of Unity)
Let $\omega = e^{(\frac{2\pi i}{n})} = \cos(\frac{2\pi}{n}) + i \sin(\frac{2\pi}{n})$, where $i = \sqrt{-1}$ is the imaginary unit. The n distinct complex n^{th} roots of unity are $\{1, \omega, \omega^2, \ldots, \omega^{n-1}\}$. Furthermore, $\omega^n = 1$, $\omega\overline{\omega} = 1$, $\overline{\omega} = \omega^{-1}$, $\overline{\omega}^k = \omega^{-k} = \omega^{n-k}$, and $1 + \omega + \omega^2 + \ldots + \omega^{n-1} = 0$.

Proposition 3 (Fourier Matrix)
A Fourier matrix F of order n is a square matrix for which

$$F^*_{(j,k)} = \frac{1}{\sqrt{n}}\, \omega^{(j-1)(k-1)} \tag{2}$$

Hence,

$$F^* = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{n-1} & \omega^{n-2} & \cdots & \omega \end{bmatrix} \tag{3}$$

Every Fourier matrix F is unitary.

[3] Two subspaces $\mathcal{S}_2$ and $\mathcal{S}_2$ are said to be orthogonal if for any $x \in \mathcal{S}_1$ and $y \in \mathcal{S}_2$, $\langle x, y \rangle = \langle y, x \rangle = 0$, i.e. x and y are perpendicular to each other.

Definition 5 (Discrete Fourier Transform)
The Discrete Fourier Transform (DFT) of a vector $x \in \mathcal{C}^n$ is given by

$$\widehat{x} = Fx, \tag{4}$$

and the inverse DFT of $\widehat{x}$ is given by

$$F^*\widehat{x} = F^{-1}\widehat{x} = x. \tag{5}$$

The DFT of an n–dimensional vector is usually computed using the celebrated Fast Fourier Transform (FFT) algorithm in $O(n \lg n)$ time and not by the direct computation implied by Equation 4, which takes $O(n^2)$ time. However, if we need to maintain only certain k Fourier coefficients, this direct computation approach requires only $O(nk)$ time. Further, note that if we use F^* as the projection matrix P in equation 16, the sketch of a vector coincides with the DFT of that vector.

Proposition 4 *(*Shift Property of DFT*)*
Let x be a vector in $\mathcal{C}^n$, and let $\widehat{x}$ be its DFT. Let x_τ be the cyclic right shift version of x by τ positions. Then, the DFT of x_τ is given by $\langle \widehat{x}, \omega^\tau \rangle$.

Proposition 5 *(*Moore–Penrose Pseudoinverse*)*
Let A be an $m \times n$ complex matrix with full rank. The Moore–Penrose pseudoinverse of A is given by

$$A^+ = \begin{cases} A^*(AA^*)^{-1} & \text{if } m \leq n \\ (A^*A)^{-1}A^* & \text{if } m \geq n \end{cases} \tag{6}$$

Proposition 6 *Let A be an $n \times m$ complex matrix with orthonormal columns. Then,*

1. *$A^*A = I_m$, where I_m is the $m \times m$ identity matrix.*
2. *if A has more rows than columns, $A^+ = (A^*A)^{-1}A^* = A^*$ and $A^\perp = I - AA^+ = I - AA^*$.*

Proposition 7 *(*Projector Matrices*)*
A complex matrix P is a projector matrix if and only if it is Hermitian and idempotent. A projector matrix P projects each vector on its column space. Projector matrices commute. If A is an $n \times m$ complex matrix, then

1. *AA^+ is a projector matrix that projects any vector in $\mathcal{C}^n$ onto $\mathcal{S}_{(A)}$, and*
2. *$A^\perp = I - AA^+$ is a projector matrix that projects any vector in $\mathcal{C}^n$ onto $\mathcal{S}^\perp_{(A)}$.*

Definition 6 (Matrix Eigenvalues)
Let A be an $n \times n$ complex matrix. A scalar λ is an eigenvalue of A if and only if there exists a non–trivial vector $x \in \mathcal{C}^n$, called its corresponding eigenvector, such that $Ax = \lambda x$. Let $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote the eigenvalues of A with maximum and minimum norm respectively. Matrix A has n eigenvalues, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ with corresponding (orthonormal) eigenvectors $x_1, x_2, \ldots, x_n$. An eigenspace of A is the subspace of $\mathcal{C}^n$ spanned by a subset of A's eigenvectors. The set of $1 \leq k \leq n$ eigenvalues of A with the largest norm, is called the set of top–k eigenvalues of A, and the set of their corresponding eigenvectors is called a set of top–k eigenvectors of A.

Definition 7 *(*Rayleigh Quotient)
Let A be an $n \times n$ Hermitian matrix. The Rayleigh quotient of A is the function

$$R_A(x) = \frac{\langle Ax, x\rangle}{\langle x, x\rangle} = \frac{x^* Ax}{x^* x}, \tag{7}$$

over all non–zero vectors $x \in \mathcal{C}^n$.

Theorem 1 (Min–Max Rayleigh Quotient of Eigenspaces)
Let A be an $n \times n$ Hermitian matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$, and corresponding orthonormal eigenvectors $x_1, x_2, \ldots, x_n$. Let $\mathcal{C}_p$ be the eigenspace generated by the eigenvectors $x_p, x_{p+1}, \ldots, x_n$, for $p = 1, 2, \ldots, n$. Then,

$$\lambda_p = \min_{0 \neq x \in \mathcal{C}_p} R_A(x), \tag{8}$$

with the minimum attained at the eigenvector x_p, and

$$\lambda_p = \max_{0 \neq x \in \mathcal{C}_{n-p+1}} R_A(x) \tag{9}$$

with the maximum attained at the eigenvector x_p. Moreover,

$$\lambda_1 = \min_{x \neq 0} R_A(x) \quad \text{and} \quad \lambda_n = \max_{x \neq 0} R_A(x), \tag{10}$$

with the minimum and maximum attained at eigenvectors x_1 and x_n respectively.

Theorem 2 *(*Courant–Fischer)
Let A be an $n \times n$ Hermitian matrix with eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$, and corresponding orthonormal eigenvectors $x_1, x_2, \ldots, x_n$. Let $\mathcal{S}_j$ be an arbitrary $(n-j+1)$–dimensional subspace of $\mathcal{C}^n$, $1 \leq j \leq n$. Then,

$$\lambda_j = \max_{\mathcal{S}_j} \min_{0 \neq x \in \mathcal{S}_j} R_A(x) \tag{11}$$

and

$$\lambda_{n-j+1} = \min_{\mathcal{S}_j} \max_{0 \neq x \in \mathcal{S}_j} R_A(x) \tag{12}$$

The extrema are attained when S_j coincides with the eigenspace generated by $x_j, x_{j+1}, \ldots, x_n$.

Definition 8 (Kronecker Product)
If A and B are $m \times m$ and $n \times n$ complex matrices respectively and $A = [a_{ij}]_{i,j=1}^m$ and $B = [b_{ij}]_{i,j=1}^n$, then the right Kronecker product of A and B, $A \otimes B$ is a $mn \times mn$ matrix given by,

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mm}B \end{bmatrix} = [a_{ij}B]_{i,j=1}^m \tag{13}$$

The Kronecker product can be extended in a straightforward way to non–square matrices as well.

We have the following theorem about eigenvalues and eigenvectors of Kronecker products of matrices.

Theorem 3 (Eigenvectors of Kronecker Product)
Let A and B be $m \times m$ and $n \times n$ matrices, and let $C = A \otimes B$. Let $\lambda_1, \lambda_2, \ldots, \lambda_m$ be the eigenvalues of A, and let $x_1, x_2, \ldots, x_m$ be its corresponding eigenvectors. Let $\mu_1, \mu_2, \ldots, \mu_n$ be the eigenvalues of B and $y_1, y_2, \ldots, y_n$ be its corresponding eigenvectors. Then, $\lambda_r \mu_s$ are the eigenvalues of C, and and $x_r \otimes y_s$ are its corresponding eigenvectors, where $r = 1, 2, \ldots, m$ and $s = 1, 2, \ldots, n$ In other words, if $A = U^* \Lambda_A U$ and $B = V^* \Lambda_B V$ then $C = (U^* \otimes V^*)(\Lambda_A \otimes \Lambda_B)(U \otimes V)$, where Λ_A and Λ_B are diagonal matrices with the eigenvalues of A and B respectively.

4 Linear Sketches for Aggregate Range Queries

Consider a discrete multi–dimensional dataset $\mathcal{D}$ of items. Without loss of generality, suppose that each item has l attributes, and each attribute takes values from the set $\{1, 2, \ldots, n\}$. Hence, each item $s = (s_1, s_2, \ldots, s_l)$, $1 \leq s_i \leq n$, can be thought of as an l–tuple (point) in $\{1, 2, \ldots, n\}^l$. The *frequency distribution* of a dataset $\mathcal{D}$ is a function $H(s)$ that describes the number of times each point s in $\{1, 2, \ldots, n\}^l$ occurs in the dataset. Thus, H can be thought of as an l–dimensional array (matrix) $n \times n \times n \times \ldots \times n$, whose each entry s gives us the frequency of point s in the dataset $\mathcal{D}$.

We are interested in aggregate range queries of the form: count the number of points in a dataset that are within a user–specified range in the l–dimensional space. Similar to frequency distributions, a range query q can be thought of as an l–dimensional array $n \times n \times n \times \ldots \times n$ with all its entries equal to 0, except those that fall within the query's range that are equal to 1. The *extent* of a range query is equal to the number of 1's it contains.

It is well known that any multi–dimensional array can be represented as a vector in a straightforward way, using a column–major, row–major, or any other space–filling curve–based traversal of the array. For the purposes of this paper, an l–dimensional $n \times n \times n \times \ldots \times n$ array is represented as an N–dimensional vector, $N = n^l$, using a column–major traversal of the array. Hence, we talk about the vector representation of multi–dimensional frequency distributions and aggregate counting range queries.

Consider now the dataset $\mathcal{D}$, whose frequency distribution is represented by the N–dimensional vector h, and an aggregate (counting) range query represented by the N–dimensional vector q. Clearly, the exact answer to this query is the standard inner product of the vectors h and q.

$$a = \langle h, q \rangle. \tag{14}$$

Example 1. Consider a two dimensional dataset, with just 3 possible values along each dimension. Let the frequency distribution H be as in Figure 1(a). Then vector representation of H is given by $h = [1, 2, 3, 4, 5, 6, 7, 8, 9]^T$.

Consider the query:

select sum(*)
from distribution
where $(1 \leq x \leq 2)$ and $(2 \leq y \leq 3)$.

Visually, the query looks like Figure 1(b). Its vector representation is $q = [1, 1, 0, 1, 1, 0, 0, 0, 0]^T$. The answer to this query is $a = \langle h, q \rangle = q^* h = [1, 1, 0, 1, 1, 0, 0, 0, 0][1, 2, 3, 4, 5, 6, 7, 8, 9]^T = 1 + 2 + 0 + 4 + 5 + 0 + 0 + 0 + 0 = 12$, where q^* is the conjugate transpose of q. ■

1	4	7
2	5	8
3	6	9

(a)

1	1	0
1	1	0
0	0	0

(b)

Figure 1: Example 1. (a) Frequency distribution H, and (b) query.

In general, a set $\{q_1, q_2, \ldots, q_m\}$ of m queries can be represented by an $N \times m$ matrix Q, where each query corresponds to a column of Q. Then, the exact answers to all these queries are given by the corresponding elements of the vector a in the following equation

$$a = Q^* h \tag{15}$$

Clearly, one can maintain the complete histogram incrementally in a trivial manner and answer the queries exactly. Unfortunately, this approach has a major drawback. The size of the complete histogram, N can be very large, making the task of computing and maintaining the frequency distribution of the dataset very expensive. An obvious trick is to maintain exact answers to only a small number of queries so that any other query can be answered approximately using results from the answers to these queries. So, the problem is to find a good set of queries or equivalently a good query matrix. In the remaining part of this section, we will define what a linear sketch is and tackle this problem algebraically.

Definition 9 (Linear Sketch)
Let P be an $N \times k$ complex projection matrix, whose columns are $P = [p_1 p_2 \cdots p_k]$. We call P a sketching matrix. Let x be a vector in $\mathcal{C}^N$. The projection of x onto p_i is given by $\langle x, p_i \rangle$, $i = 1, 2, \ldots, k$. The (linear) sketch or projection of x with respect to the sketching matrix P is the k–dimensional vector given by $\widehat{x} = [\langle x, p_1 \rangle \langle x, p_2 \rangle \cdots \langle x, p_k \rangle]^T$. Equivalently,

$$\widehat{x} = P^* x \tag{16}$$

The sketch of an $N \times m$ matrix A with respect to the sketching matrix P is a matrix whose columns are equal to the sketch of the corresponding column of A, i.e.

$$\widehat{A} = P^* A. \tag{17}$$

Clearly, the sketch of a vector x is nothing more than a linear transformation/projection of x onto a lower–dimensional space. The size of a sketch of an N–dimensional vector computed using an $N \times k$ sketching matrix is $O(k)$.

Trivial Sketching Matrix

Notice the similarity between Equations 14 and 16. The query matrix Q, can be used as a sketching matrix for maintaining the sketch $\widehat{h}$ of the frequency distribution h of a dataset. ($P = Q$ in equation 16 gives equation 15). Then the sketch $\widehat{h}$, is simply the exact answers to all the queries in Q. Maintenance of such a sketch simply corresponds to using the following strategy: when a new data point is inserted (or deleted) in the stream, find all the queries that contain this data point and update the answer to the corresponding queries.

If N' is the number of non–zero elements of Q, then maintaining $\widehat{h}$ can be done in $O(N')$ time and memory. Given that there is at least a single non–zero entry in every query, $O(N')$ is at least as large as the number of queries, which can be prohibitively large. If we consider all range queries of a fixed extent in the 1–dimensional case, the number of queries is as large as $O(N^2)$. Therefore the time and space requirement for maintaining such a sketch is too big. So the problem here is to find a good sketching matrix that has a small space representation. In the light of the earlier discussion on having a small set of queries, the sketching matrix is nothing but the query matrix with a set of special queries.

Some Classical Sketching Matrices

Fourier matrix is a classical sketching matrix (see Section [**?**]–Definition 5). When we use the Fourier matrix as the sketching matrix, the sketch $\widehat{h}$, of the data distribution h, is simply the Discrete Fourier Transform (DFT) of h. The DFT of a vector of length N is usually computed using the celebrated Fast Fourier Transform (FFT) algorithm in $O(N \lg N)$ time and not by the direct computation as implied by Equation 4, which takes $O(N^2)$ time. However, if we need to maintain only certain k Fourier coefficients, this direct computation approach requires only $O(Nk)$ time. Further it can be maintained incrementally as we shall discuss in the next section. In order to maintain the top–k Fourier coefficients of the distribution, we simply use only the corresponding Fourier vectors for the sketching matrix. Note that in the case of streaming data, we do not know in advance which of the Fourier vectors correspond to the top–k coefficients.

Similarly any Wavelet matrix can be used as the sketching matrix to get the the corresponding Discrete Wavelet transform. For example the Haar matrix will give the Haar transform.

Linear sketches using random projections have been recently used for streaming data applications. In this case, the sketching matrix, simply contains random vectors. Due to the Johnson–Lindenstrauss lemma [5], the linear sketch of a vector using random projections has a nice property that the norm of the vector can be estimated well from the norm of the sketch.

4.1 Incremental Sketch Maintanence

The sketch $\widehat{h}$ of a frequency distribution h, is a linear transformation of h, by a sketching matrix P. Therefore it can be computed very easily (as a matrix multiplication) in $O(kN)$ time.

Whenever a point s is inserted to (or deleted from) the dataset $\mathcal{D}$, the sketch of the frequency distribution of $\mathcal{D}$ can be updated by adding (or subtracting) the sketch of s. This is true because sketching is a linear operation. Say u is the update vector to the distribution. Therefore, the h is updated to $h + u$. The updated sketch will be $P^*(h + u) = P^*h + P^*u = \widehat{h} + \widehat{u}$. Therefore the time to update the sketch includes time to compute the sketch $\widehat{u}$, of the update vector, and the time to add the two sketches. The time to add the two sketches is proportional to the size of the two sketches or $O(k)$. The time to compute $\widehat{u}$ depends on the number of non–zero components in u. If we consider update after every single insert or delete, there is just a single non–zero component. If we consider a constant number of non–zero components in the update vector, computing the sketch of the update takes just $O(k)$ time.

Note that memory requirements of maintaining the sketch of a frequency distribution is $O(k)$ plus the amount of memory needed for the sketching matrix P. If P is stored explicitly, then the memory requirement is $O(kN)$. However, if the sketching matrix P is sparse, it can stored using $O(N')$ space, where N' is the number of non–zero elements of P. Further, in several instances, P does not have to be explicitly stored. If P has a succinct representation, any element of P can be generated using a program with a small footprint in $O(1)$ time. Then only the rows corresponding to the non–zero elements in the update vector u can be generated on the fly and used to compute $\widehat{u}$. Therefore incremental update of the linear sketch of the data

distribution can be done in $O(k)$ space and time.

Having seen the computation of the sketch of the frequency distribution of the data, in the following sections, we present how approximate answers can be computed using sketches and prove some bounds on approximation errors.

4.2 Approximate Answers to Queries using Linear Sketches

First, we consider the problem of computing approximate answers to a set of queries Q when we are given a linear sketch of the frequency distribution h of a dataset.

Let P be the $N \times k$ sketching matrix used to compute the sketch $\widehat{h}$. Observe that if the column space of P has dimension N, then P is an invertible matrix, h can be reconstructed from $\widehat{h}$ exactly, and all the queries can be answered exactly. Unfortunately, since we desire $k \ll N$, the sketching matrix P is not invertible, and thus h can not be recovered from its sketch exactly, and thus the queries can not be answered exactly in general.

We consider two methods of answering the queries Q using the sketch $\widehat{h}$.

Approach I: Using the sketching matrix P and the sketch $\widehat{h}$, find a frequency distribution $\widetilde{h}$ that has the smallest norm among all those with sketch $\widehat{h}$ (e.g. it is closest to the true h in the Euclidean norm). This can be done by effectively, using least–squares to recover h from $\widehat{h}$. Thus,

$$\widetilde{h} = (P^*)^+ \widehat{h} \tag{18}$$

Here A^+ is the Moore–Penrose Pseudoinverse of matrix A. Then, using the estimated frequency distribution, estimate the answers to the queries using the standard inner product,

$$\widetilde{a}_1 = Q^* \widetilde{h} \tag{19}$$

Approach II: Compute the sketch of the queries with respect to the same sketching matrix P. Then, estimate the answers to the queries using the standard inner product of the query sketches and the frequency sketch.

$$\widetilde{a}_2 = (\widehat{Q})^* \widehat{h} \tag{20}$$

Lemma 1 *If the columns of the sketching matrix P are orthonormal, then approaches I and II above yield the same answers to all the queries Q.*

Proof

$$\begin{aligned}
\widetilde{a}_1 &= Q^* \widetilde{h} && \text{from equation 19} \\
&= Q^* (P^*)^+ \widehat{h} && \text{substituting } \widetilde{h} \text{ from equation 18} \\
&= Q^* (P^*)^* (P^* (P^*)^*)^{-1} \widehat{h} && \text{from Proposition 6} \\
&= Q^* P (P^* P)^{-1} \widehat{h} && (21) \\
&= Q^* P \widehat{h} && \text{because } P^* P = I_N \text{ from Proposition 6} \\
&= (P^* Q)^* \widehat{h} \\
&= (\widehat{Q})^* \widehat{h} = \widetilde{a}_2 && \text{from equation 20}
\end{aligned}$$

■

Lemma 1 above shows that the two approaches for computing approximate answers to queries from sketches are equivalent when the sketching matrix has orthonormal columns. Observe that all the information that can be captured about any vector when using a sketching matrix P with linearly dependent columns, is available in the sketch of that vector computed using a sketching matrix consisting of a linearly independent subset of the columns of P. Moreover, it follows from Proposition 1, that an orthonormal subset of the columns of P can always be computed using the Gram–Schmidt orthogonalization algorithm. Henceforth, without loss of generality, and unless stated otherwise, we only consider sketching matrices with orthonormal columns and use Approach II to estimate the answers to queries.

4.3 Bounds on the Approximation Error

We compute upper bounds on the error in computing approximate answers to queries from a sketch of the frequency distribution of a dataset. Recall, that we assume sketching matrices with orthonormal columns, and that the approximate answer to a set of queries is computed with Approach II, i.e. the inner product of the sketch of a query and the sketch of the distribution. We assess the quality of approximation of query answers using the sum of squares of the errors (SSE) of the approximate answers.

Let P be an $N \times k$ sketching matrix, $\widehat{h}$ be the sketch of a frequency distribution h, and Q be an $N \times m$ query matrix. The error in approximating the answers to the m queries is the vector e below

$$\begin{aligned} e &= a - \widetilde{a} \\ &= Q^*h - (\widehat{Q})^*\widehat{h} \\ &= Q^*h - (P^*Q)^*P^*h \\ &= Q^*h - Q^*PP^*h \\ &= Q^*(I - PP^*)h \\ &= Q^*P^{\perp}h \end{aligned} \tag{22}$$

Here, $A^{\perp} = I - AA^{+}$ is a projector matrix that projects any vector in $\mathcal{C}^n$ onto the complementary orthogonal subspace of the column space of A. The sum of the squared errors (SSE) of the estimated answers is given by the norm of e,

$$\|e\|^2 = e^*e = (Q^*P^{\perp}h)^*Q^*P^{\perp}h = h^*(P^{\perp})^*QQ^*P^{\perp}h, \tag{23}$$

and since the matrix $P^{\perp}$ is Hermitian, it follows that

$$\|e\|^2 = h^*P^{\perp}QQ^*P^{\perp}h \tag{24}$$

Since $h = 0$ implies $e = 0$, hereafter w.l.o.g we assume non–zero frequency distributions h.

Lemma 2 *Let h be a non–zero frequency distribution and Q a query matrix. Let P be a sketching matrix with orthonormal columns. Then the norm of the error in approximating the answers to Q using sketches with respect to P is*

$$\|e\|^2 \leq \lambda_{\max}(QQ^*)\ (\|h\|^2 - \|\widehat{h}\|^2) \tag{25}$$

Proof
From Equation 24, the norm of the error is $\|e\|^2 = h^*P^{\perp}QQ^*P^{\perp}h$. Since $P^{\perp}$ is Hermitian, it follows that

$$\|e\|^2 = (P^{\perp}h)^*QQ^*P^{\perp}h \tag{26}$$

Note that the matrix QQ^* is Hermitian because $(QQ^*)^* = QQ^*$. Then, from equation 26, and the definition of the Rayleigh quotient for the matrix QQ^* evaluated at $P^\perp h$, we have that

$$R_{QQ^*}(P^\perp h) = \frac{\|e\|^2}{(P^\perp h)^* P^\perp h} \tag{27}$$

Therefore,

$$\begin{array}{rcl} \|e\|^2 & = & R_{QQ^*}(P^\perp h)\ ((P^\perp h)^* P^\perp h) \\ & = & R_{QQ^*}(P^\perp h)\ (h^*(P^\perp)^* P^\perp h) \end{array} \tag{28}$$

Since $P^\perp$ is Hermitian and idempotent, $(P^\perp)^* P^\perp = P^\perp$. Thus,

$$\|e\|^2 = R_{QQ^*}(P^\perp h)\ (h^* P^\perp h) \tag{29}$$

Since the sketching matrix P has orthonormal columns, from Proposition 6 we have $P^\perp = I - PP^*$. Then,

$$\begin{array}{rcl} h^* P^\perp h & = & h^*(I - PP^*)h \\ & = & h^* h - h^* PP^* h \\ & = & h^* h - (P^* h)^* P^* h \\ & = & h^* h - (\widehat{h})^* \widehat{h} \\ & = & \|h\|^2 - \|\widehat{h}\|^2 \end{array} \tag{30}$$

Therefore,

$$\|e\|^2 = R_{QQ^*}(P^\perp h)\ (\|h\|^2 - \|\widehat{h}\|^2) \tag{31}$$

Moreover, since $h \neq 0$, from Theorem 1 follows that

$$R_{QQ^*}(P^\perp h) \leq \max_{x \neq 0}\ R_{QQ^*}(x) \leq \lambda_{\max}(QQ^*). \tag{32}$$

Therefore,

$$\|e\|^2 \leq \lambda_{\max}(QQ^*)\ (\|h\|^2 - \|\widehat{h}\|^2) \tag{33}$$

■

Lemma 2 demonstrates that one way to reduce the norm of the error is to choose a sketching matrix P so that the norm of the sketch of the distribution is as close to the actual norm of the distribution as possible. For example, P may consist of the Fourier vectors corresponding to the highest–norm Fourier coefficients of h. Likewise, P may consist of the Haar–wavelet vectors corresponding to the largest wavelet coefficients of h, etc. This is the basic idea in the work by Guha et al [11]. In the work by Thaper et al [15], P is chosen to be a large enough random matrix, since then by the Johnson–Lindenstrauss lemma [5] the norm of the sketch of h is close to the norm of h.

An alternate bound on the norm of the approximation error is given by the following lemma.

Lemma 3 *Let h be a non–zero frequency distribution and Q a query matrix. Let P be a sketching matrix with orthonormal columns. Then the norm of the error in approximating the answers to Q using sketches with respect to P is*

$$\|e\|^2 = R_{P^\perp QQ^* P^\perp}(h)\ \|h\|^2 \tag{34}$$

Here $R_A(x)$ is the Raleigh quotient of matrix A.

Proof
The norm of the error, from equation 24, is

$$\|e\|^2 = h^* P^\perp QQ^* P^\perp h \tag{35}$$

Observe that the matrix $P^\perp QQ^* P^\perp$ is Hermitian because $(P^\perp QQ^* P^\perp)^* = (P^\perp)^* QQ^* (P^\perp)^* = P^\perp QQ^* P^\perp$. Therefore, the Rayleigh quotient of the matrix $P^\perp QQ^* P^\perp$ at $x = h$ is

$$R_{P^\perp QQ^* P^\perp}(h) = \frac{\|e\|^2}{h^* h} = \frac{\|e\|^2}{\|h\|^2} \tag{36}$$

or, equivalently

$$\|e\|^2 = R_{P^\perp QQ^* P^\perp}(h)\ \|h\|^2 \tag{37}$$

■

Lemma 3 demonstrates that an alternate strategy for reducing the approximation error would be to choose a sketching matrix P that minimizes the maximum value of the Rayleigh quotient of the matrix $P^\perp QQ^* P^\perp$. The following Theorem shows what sketching matrix achieves just that.

Theorem 4 *Let Q be an $N \times m$ query matrix, h a non–zero frequency distribution, and P be an $N \times k$ sketching matrix with orthonormal columns. Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ be the eigenvalues of the matrix QQ^*, and let $x_1, x_2, \ldots, x_N$ be its corresponding orthonormal eigenvectors. The maximum value of the Rayleigh quotient $R_{P^\perp QQ^* P^\perp}(x)$ is minimized when P has as columns the top–k eigenvectors of the matrix QQ^*, i.e. when $P = [x_{N-k+1} x_{N-k+2} \ldots x_N]$. Furthermore, in that case, the norm of the approximation error is*

$$\|e\|^2 \leq \lambda_{N-k} \|h\|^2 \tag{38}$$

Proof:
By definition, for any non-zero vector x, we have that

$$R_{P^\perp QQ^* P^\perp}(x) = \frac{x^* P^\perp QQ^* P^\perp x}{x^* x} \tag{39}$$

Then, by dividing and multiplying the right–hand side of the equation above with $(P^\perp x)^* P^\perp x$, it follows that

$$R_{P^\perp QQ^* P^\perp}(x) = \frac{x^* P^\perp QQ^* P^\perp x}{(P^\perp x)^* P^\perp x}\ \frac{(P^\perp x)^* P^\perp x}{x^* x} \tag{40}$$

Further, since both QQ^* and $P^\perp$ are Hermitian matrices, by definition of the Rayleigh quotient follows that

$$R_{P^\perp QQ^* P^\perp}(x) = R_{QQ^*}(P^\perp x)\ R_{P^\perp}(x) \tag{41}$$

From Theorem 1, we have that $R_{P^\perp}(x) \leq \lambda_{\max}(P^\perp)$. However, since $P^\perp$ is idempotent, $(P^\perp)^2 = P^\perp$, and therefore the eigenvectors of $P^\perp$ are precisely the columns of $P^\perp$, and all the eigenvalues of $P^\perp$ are equal to 1. Hence, $R_{P^\perp}(x) \leq \lambda_{\max}(P^\perp) = 1$. Therefore,

$$R_{P^\perp QQ^* P^\perp}(x) \leq R_{QQ^*}(P^\perp x)\ (\max_{z \neq 0}\ R_{P^\perp}(z)) = R_{QQ^*}(P^\perp x)\ (1) = R_{QQ^*}(P^\perp x). \tag{42}$$

Since for any vector x, $P^\perp x \in \mathcal{S}^\perp_{(P)}$, from the equation above follows that

$$R_{P^\perp QQ^* P^\perp}(x) \leq R_{QQ^*}(P^\perp x) \leq \max_{y \in \mathcal{S}^\perp_{(P)}} R_{QQ^*}(y). \tag{43}$$

Thus, We want to find P so that the maximum value of $R_{QQ^*}(y)$ for any $y \in \mathcal{S}^\perp_{(P)}$ is minimized.

The rank of the sketching matrix P is equal to k, since its columns are linearly independent. Thus, P's column space and the complementary orthonormal space of its column space are of dimension k and $N-k$ respectively. From the Courant–Fischer Theorem 2 we know that

$$\min_S \max_{y \in S} R_{QQ^*}(y) = \lambda_{N-k}, \tag{44}$$

where the minimization is over all subspaces of $S \subseteq \mathcal{C}^N$ of dimension $N-k$, and that the minimum is achieved for the eigenspace spanned by the eigenvectors $x_1, x_2, \ldots, x_{N-k}$. Note that in that case, the complementary orthonormal subspace of S is the eigenspace spanned by the eigenvectors $x_{N-k+1}, \ldots, x_N$. Therefore, we want $\mathcal{S}^\perp_{(P)}$ to be equal to the eigenspace spanned by the eigenvectors $x_1, x_2, \ldots, x_{N-k}$, and thus the space $\mathcal{S}_{(P)}$ to equal to the eigenspace spanned by the eigenvectors $x_{N-k+1}, \ldots, x_N$.

Therefore, when $P = [x_{N-k+1} x_{N-k+2} \ldots x_N]$, we have that

$$\|e\|^2 \leq \max_{x \neq 0}\ R_{P^\perp QQ^* P^\perp}(x)\ \|h\|^2 \leq \lambda_{N-k}\ \|h\|^2 \tag{45}$$

■

4.4 Extending a Linear Sketch

We have see how sketching matrices can be constructed using the queries. In many instances, it is often the case that a particular sketch of the frequency distribution is already (required to be) maintained for various other applications. Let P_1 be the sketching matrix of order $N \times k_1$ for such a sketch. So, the natural question is how should a sketching matrix P_1 be extended by k_2 columns, so that a the norm of the approximation error for the queries Q be minimized? Let P_2 be the matrix of the additional k_2 columns used in extending the $N \times k_1$ sketching matrix P_1 to an $N \times k$ sketching matrix P, $k = k_1 + k_2$, i.e. $P = [P_1 P_2]$. Note that approximation error when using P is always smaller than the approximation error when using either P_1 or P_2 alone.

The first thing that comes to mind is take P_2 to have as columns the top–k_2 eigenvectors of the matrix QQ^*. However, the columns of P_2 may not be linearly independent from those of P_1, leading to wasted effort

and sub–optimal performance. So, we can assume, without loss of generality, that P_1 and P_2 should form a bi–orthonormal set, i.e. that the columns of P_1 and P_2 are all orthonormal.

Then we have the following lemma.

Lemma 4 *Let P_1 and P_2 be two matrices of dimensions $N \times k_1$ and $N \times k_2$, that form a bi–orthonormal system. Let $P = [P_1 P_2]$. Then,*

$$P^{\perp} = {P_1}^{\perp} {P_2}^{\perp} = {P_2}^{\perp} {P_1}^{\perp}. \tag{46}$$

Proof
Since P has orthonormal columns, by Proposition 6 we have

$$\begin{aligned} P^{\perp} &= I - PP^* \\ &= I - [P_1 P_2][{P_1}^* {P_2}^*]^T \\ &= I - P_1 {P_1}^* - P_2 {P_2}^* \end{aligned} \tag{47}$$

Furthermore, since both P_1 and P_2 have orthonormal columns, it follows that

$$\begin{aligned} {P_1}^{\perp} {P_2}^{\perp} &= (I - P_1 {P_1}^*)(I - P_2 {P_2}^*) \\ &= (I - P_1 {P_1}^* - P_2 {P_2}^*) + P_1 ({P_1}^* P_2) {P_2}^* \end{aligned} \tag{48}$$

Since P_1 and P_2 form a bi–orthonormal system, ${P_2}^* P_1 = 0$. Therefore,

$${P_1}^{\perp} {P_2}^{\perp} = P^{\perp} + P_1 0 {P_2}^* = P^{\perp}. \tag{49}$$

Similarly, it can be shown that ${P_2}^{\perp} {P_1}^{\perp} = P^{\perp}$. ■

Using Lemma 4, we obtain the following corollary to Theorem 4.

Corollary 1 *Let P_1 be a sketching matrix with orthonormal columns. Let Q be a query matrix and h a non–zero frequency distribution. Let P be a k_2 extension of P_1, such that $P = [P_1 P_2]$ and P_1 and P_2 form a bi–orthonormal system. Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ be the eigenvalues of the matrix ${P_1}^{\perp} Q Q^* {P_1}^{\perp}$, and $x_1, x_2, \ldots, x_N$ be the corresponding eigenvectors. The maximum value of the Rayleigh quotient of the matrix $P^{\perp} Q Q^* P^{\perp}$ is minimized when P_2 has as columns the top–k_2 eigenvectors of the matrix ${P_1}^{\perp} Q Q^* {P_1}^{\perp}$, and the maximum value attained is equal to λ_{N-k_2}. Further, the norm of the approximation error in answering the queries Q when using P is*

$$\|e\|^2 \leq \lambda_{N-k_2} \|h\|^2. \tag{50}$$

Proof
From Lemma 4, $P^{\perp} = {P_1}^{\perp} {P_2}^{\perp} = {P_2}^{\perp} {P_1}^{\perp}$. Therefore,

$$\begin{aligned} P^{\perp} Q Q^* P^{\perp} &= {P_2}^{\perp} {P_1}^{\perp} Q Q^* {P_1}^{\perp} {P_2}^{\perp} \\ &= {P_2}^{\perp} ({P_1}^{\perp} Q) ({P_1}^{\perp} Q)^* {P_2}^{\perp} \end{aligned} \tag{51}$$

The corollary follows by applying Theorem 4 for the "query" matrix ${P_1}^{\perp} Q$ and the sketching matrix P_2. Hence, P_2 should consist of the top–k_2 eigenvectors of the matrix ${P_1}^{\perp} Q Q^* {P_1}^{\perp}$. Observe that each eigenvector

of the matrix ${P_1}^{\perp} Q Q^* {P_1}^{\perp}$ is orthogonal to the columns of P_1, and thus the matrices P_1 and P_2 form a bi–orthonormal system. ■

This technique of extending linear sketches is of particular significance because, one can take into consideration both the structure in the data as well as the queries while constructing sketches. For example, traditional sketches such as the top–k Fourier or Haar coefficients and most other linear sketching techniques do not make use of the queries at all. (Histograms are also linear sketches) Similarly the linear sketches using eigenvectors of the query matrices described in previous sections are constructed completely independent of the actual data. This approach of extending sketches gives us a neat and unique technique for making use of both the data as well as the queries to construct sketches for better approximate query answering by doing the following. Let P_1 be the sketching matrix that is based on the data: for e.g. the Fourier vectors corresponding to the top–k_1 Fourier coefficients of the data distribution. P_1 should have orthonormal vectors, if not one can always find, via the Gram–Schmidt orthonormalization algorithm, a set with (possibly fewer) orthonormal vectors with the same span as P_1 as use them instead. Then we can use the above technique of extending linear sketches by using the query matrix to find P_2 that has k_2 vectors.

5 Circular Range Queries and Circulant Matrices

Whenever there is a new point added to a dataset $\mathcal{D}$, the sketch of the frequency distribution needs to be updated. For this, we require, the corresponding row of the $N \times k$ sketch matrix P. If we store the matrix P, then the space requirement for the sketch maintenance would be $O(kN)$. Since N can be very large this may be prohibitive.

For answering a query q using Approach II, we need to compute the sketch of the query. This takes $O(sk)$ where s is the extent of the query q. In this section, we shall discuss properties of certain classes of queries that have a succint representation for the eigenvectors of their query matrices, leading to efficient maintenance of sketches and storage of sketch matrix. We consider the set of all right circular shifted versions of range queries with a given extent w. We seek succint representations for suitable sketching matrices in order to approximately answer all such queries with small sum–of–squares error. Based on the discussion in section 4, in a nutshell, the sketching matrix P should consist of the top–k eigenvectors of the query matrix. The objective of this section is to show that for circular queries such eigenvectors can be found within small amount of time and memory.

5.1 Circular Queries in 1–D

Consider a 1–dimensional dataset. A range query of a extent w will have w consecutive 1s in its query vector. There are $N - w + 1$ possible range queries of extent w. All of these queries are contained in the set of N queries resulting from the N circular right shifts of a single range query with extent w. We call this set of queries the set of circular range queries with extent w. There is a natural way to order the set of circular queries so that the query matrix Q representing this set is a special kind of matrix, a so called circulant matrix.

Example.
Consider a 1–dimensional dataset, with the domain of the single attribute having 4 ordered values, i.e. $N = 4$. Consider the range query $q = [1, 1, 0, 0]^T$ whose extent is 2. The set of circular range queries with extent 2 can be derived from q by considering all the circular shifts of q. This set of

circular queries is represented by the matrix Q

$$Q = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{52}$$

The i^{th} column of Q is equal to the right circular shift of q by $i-1$ positions, for $i = 1, 2, \ldots, N$. We call the 1st column the *mother* query of Q.

We review certain properties of circulants matrices that we will use later on. Interested readers are referred to Davis [6] for further details.

Definition 10 *(*Circulant Matrix)
A circulant (matrix) of order n, is a square matrix of the form

$$C = circ(c_0, c_1, \ldots, c_{n-1}) = \begin{bmatrix} c_0 & c_1 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & \cdots & c_{n-2} \\ \vdots & \vdots & \vdots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{bmatrix} \tag{53}$$

Each row of C is a circular right shift of its preceding row.

Proposition 8 (See Davis [6]) *A matrix C is circulant if and only if C^* is a circulant. If both A and B are circulant matrices then AB and $A+B$ are also circulant.*

Theorem 5 (See Davis [6]) *(Universal Eigenvectors of Circulants)*
Let C be a circulant matrix of order n. Matrix C is diagonalized by the Fourier matrix F of order n,

$$C = F^* \Lambda F, \tag{54}$$

where Λ is a diagonal matrix with the eigenvalues of C as its diagonal elements. Thus, each column of F^* is a (right) eigenvector of every circulant matrix, i.e. the columns of F^* is a universal set of eigenvectors for for all the circulant matrices. Moreover, for each root of unity $\rho \in \{1, \omega, \omega^2, \ldots, \omega^{n-1}\}$,

$$\lambda(\rho) = \sum_{k=0}^{n-1} c_k \rho^k \tag{55}$$

is an eigenvalue of C, and its corresponding eigenvector is

$$\frac{1}{\sqrt{n}}[1 \quad \rho \quad \rho^2 \quad \cdots \quad \rho^{n-1} \quad]^T \tag{56}$$

Further, any matrix that is diagonalized by the Fourier matrix is a circulant matrix.

Let Q be the query matrix formed by all the circular range queries with extent w. Since Q is circulant, from Theorem 5 it follows that $Q = F^*\Lambda F$. Further,

$$\begin{aligned} QQ^* &= F^*\Lambda F(F^*\Lambda F)^* \\ &= F^*\Lambda(FF^*)\Lambda^* F, \end{aligned} \tag{57}$$

and since F is unitary, it follows that $FF^* = I$ and

$$QQ^* = F^*\Lambda(I)\Lambda^* F = F^*\Lambda\Lambda^* F. \tag{58}$$

Therefore, QQ^* is also circulant and, for every root of unity ρ, its corresponding eigenvalue equals the norm of Q's eigenvalue corresponding to that ρ. Therefore, the top–k eigenvectors of QQ^* are the same as the top–k eigenvectors of Q, and vice versa.

Recall that from Theorem 4, a good choice for an $N \times k$ sketch matrix P is to use a matrix with k columns that are the top–k (Fourier) eigenvectors of Q. Then, the sketch $\widehat{h} = P^*h$ of a frequency distribution h consist simply of the corresponding coefficients of the DFT of h. Thus, the sketch matrix P does not need to be stored at all. Moreover, since we use approach II to estimate the answer to each query q in Q, we need the sketch $\widehat{q}$ of each query q. However, each query q is a right circular shift of the mother query q_1 of Q (the first column of Q), and hence the sketch of q can be computed form the sketch of q_1 in $O(k)$ time using the shift property of the DFT (Property 4).

The following Lemma considers a a circular set of queries with more than one extent.

Lemma 5 *Let Q_1 and Q_2 be the query matrices corresponding to the set of all circular queries with extent w_1 and w_2, respectively. Let $Q = [Q_1 Q_2]$, i.e. Q represents the set of all circular queries with extent w_1 or w_2. The matrix QQ^* is circulant, and for each root of unity ρ, the corresponding eigenvalue of QQ^* is equal to the sum of the corresponding eigenvalues of $Q_1{Q_1}^*$ and $Q_2{Q_2}^*$.*

Proof:
Let Λ_1 and Λ_2 be diagonal matrices with the eigenvalues of $Q_1{Q_1}^*$ and $Q_1{Q_1}^*$ corresponding to the universal set of eigenvectors F^*, respectively. Since $Q = [Q_1 Q_2]$, we have that

$$\begin{aligned} QQ^* &= [Q_1 Q_2][Q_1 Q_2]^* \\ &= [Q_1 Q_2][{Q_1}^* {Q_2}^*]^T \\ &= Q_1{Q_1}^* + Q_2{Q_2}^* \\ &= F^*\Lambda_1 F + F^*\Lambda_2 F \\ &= F^*(\Lambda_1 + \Lambda_2)F \\ &= F^*\Lambda F, \end{aligned} \tag{59}$$

where $\Lambda = \Lambda_1 + \Lambda_2$. Thus, QQ^* is circulant with eigenvalues the elements of the diagonal matrix Λ. ■

Lemma 5 enables us to find the top–k eigenvectors for a set of circular queries with multiple extents using the basic machinery we already discussed for a circular query set with queries of a single extent only.

5.2 Multi–Dimensional Circular Queries

We consider circular queries for multi–dimensional datasets. We show that a succinct representation of a suitable sketching matrix exists in this case as well. Intuitively, the idea is based on the fact that an l–dimensional regular grid can be represented as the Kronecker product of l 1–dimensional regular grids.

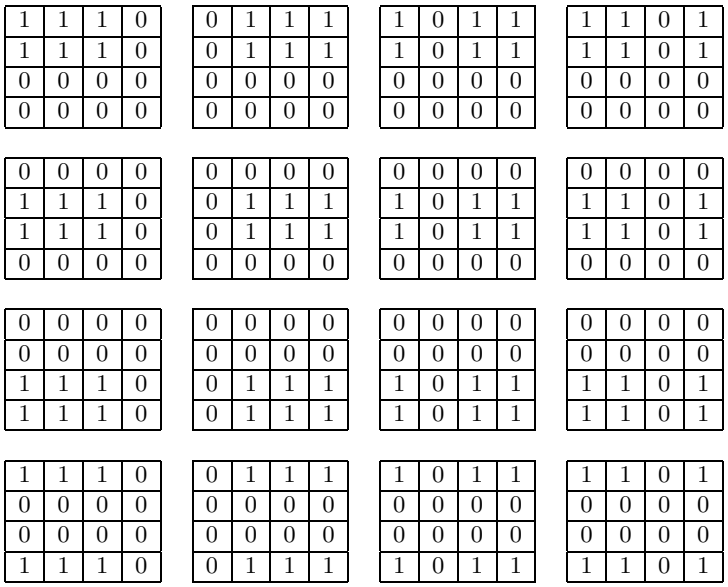


Figure 2: The set of the 16 right circular queries with extent 2×3.

Example

Consider a 2-dimensional dataset, with 4 possible values along each dimension. In this case, $l = 2$, $n = 4$, and $N = n^l = 16$. Consider all queries 2–D range queries with extent 2×3. The queries would look as in Figure 2. Then the query matrix would be the following. The columns are the vector representations of the above queries taken in a column–major order. The queries are visited in a row–major way, w.r.t. Figure 2.

$$Q = \left[\begin{array}{cccc|cccc|cccc|cccc} 1&0&0&1&0&0&0&0&1&0&0&1&1&0&0&1 \\ 1&1&0&0&0&0&0&0&1&1&0&0&1&1&0&0 \\ 0&1&1&0&0&0&0&0&0&1&1&0&0&1&1&0 \\ 0&0&1&1&0&0&0&0&0&0&1&1&0&0&1&1 \\ \hline 1&0&0&1&1&0&0&1&0&0&0&0&1&0&0&1 \\ 1&1&0&0&1&1&0&0&0&0&0&0&1&1&0&0 \\ 0&1&1&0&0&1&1&0&0&0&0&0&0&1&1&0 \\ 0&0&1&1&0&0&1&1&0&0&0&0&0&0&1&1 \\ \hline 1&0&0&1&1&0&0&1&1&0&0&1&0&0&0&0 \\ 1&1&0&0&1&1&0&0&1&1&0&0&0&0&0&0 \\ 0&1&1&0&0&1&1&0&0&1&1&0&0&0&0&0 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&0&0 \\ \hline 0&0&0&0&1&0&0&1&1&0&0&1&1&0&0&1 \\ 0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&0 \\ 0&0&0&0&0&1&1&0&0&1&1&0&0&1&1&0 \\ 0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1 \end{array}\right] \tag{60}$$

The query matrix Q is not circulant in general. Notice that it contains blocks that are circulants. Let Q_1 be the query matrix for the set of 1-D circular queries with extent 2. Clearly, Q_1 and $Q_1 Q_1{}^*$ are both circulant. Similarly, Q_2 be the query matrix for the set of 1-D circular queries with extent 3. Clearly, Q_2 and $Q_2 Q_2{}^*$ are both circulant. Therefore,

$$Q_1 = \begin{bmatrix} 1&0&0&1 \\ 1&1&0&0 \\ 0&1&1&0 \\ 0&0&1&1 \end{bmatrix} \qquad Q_2 = \begin{bmatrix} 1&0&1&1 \\ 1&1&0&1 \\ 1&1&1&0 \\ 0&1&1&1 \end{bmatrix} \tag{61}$$

Moreover, observe that

$$Q = \begin{bmatrix} Q_1 & 0 & Q_1 & Q_1 \\ Q_1 & Q_1 & 0 & Q_1 \\ Q_1 & Q_1 & Q_1 & 0 \\ 0 & Q_1 & Q_1 & Q_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \otimes Q_1 = Q_2 \otimes Q_1 \tag{62}$$

In general, for l–dimensional datasets, the query matrix Q for the set of all circular range queries with extent $w_1 \times w_2 \times \ldots \times w_l$, is given by $Q = Q_l \otimes Q_{l-1} \otimes \ldots \otimes Q_1$, where each Q_i is the query matrix for the set of all 1–dimensional circular queries with extent w_i, $i = 1, 2, \ldots, l$. Therefore, the eigenvectors of Q are the columns of $F^* \otimes F^* \otimes \ldots \otimes F^*$ (l times) and the eigenvalues of Q are the products of eigenvalues of $Q_1, Q_2, \ldots, Q_l$. Moreover, if we are seeking a sketching matrix of size $N \times k$, then we need to consider only the top–$(k^{1/l})$ eigenvalues/eigenvectors of each Q_i.

6 Experimental Evaluation

6.1 Set Up

Consider an l–dimensional dataset $\mathcal{D}$ with $N = n^l$, and an $N \times k$ sketching matrix P. Let h the frequency distribution of $\mathcal{D}$, and $\widehat{h}$ its sketch with respect to P. Consider also an $N \times m$ query matrix Q with queries (columns) q_i, $i = 1, 2, \ldots, m$, where a_i and $\widehat{a_i}$ are the exact and approximate answers computed for query q_i using Approach II for the sketching matrix P.

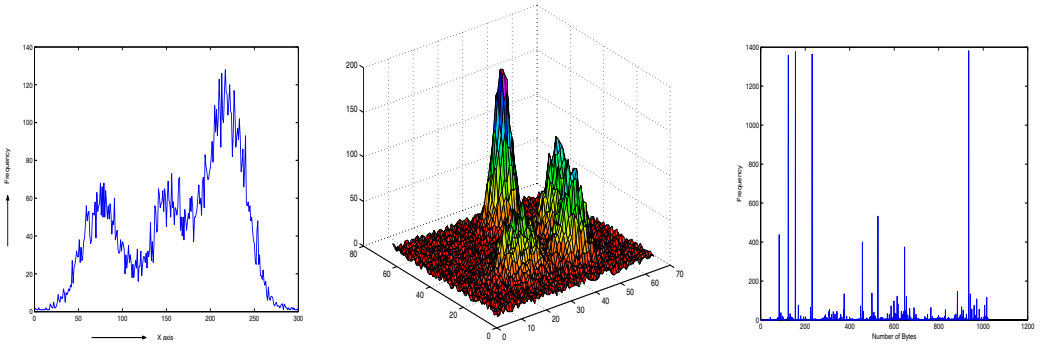
6.1.1 Datasets

We perform experiments over both synthetic and real datasets that are described below.

- $\mathcal{D}_1$: is a synthetic 1-dimensional dataset with $n = 300$, and $N = n^l = 300$. It consists of $12,000$ points in $[0, 1)$, and randomly generated using a mixture of 4 Gaussians. Each Gaussians has mean uniformly chosen in $[0, 1)$, and variance equal to 0.01. Each Gaussian is chosen with equal probability. The frequency distribution of this dataset is shown in Figure 3(a).
- $\mathcal{D}_2$: is a synthetic 2–dimensional dataset with $n = 64$, and hence $N = n^l = n^2 = 4096$. It consists of $45,000$ points, of which $5,000$ are generated from the uniform distribution over $[1, 64]^2$, and $40,000$ are generated from a mixture of 4 Gaussian distributions. Each of these 4 Gaussians has mean uniformly distributed in $[1, 64]^2$, and variance $9, 16, 16$ and 25 respectively. The frequency distribution of the data is shown in Figure 3(b).
- $\mathcal{D}_3$: is a real 1-dimensional dataset obtained from [1]. The trace consists of a day's worth of all HTTP requests to the EPA WWW server located at Research Triangle Park, NC. For the experiment, we consider number of bytes in reply to each request. We consider short requests that have less than 1000KBytes for the experiment. It has 17,687 entries. The frequency distribution of this dataset is shown in Figure 3(c).

6.1.2 Queries

For our experiments, we consider two kinds of queries: Random Queries and Fixed Extent Queries.



(a) Synthetic 1D dataset, $\mathcal{D}_1$ (c) Synthetic 2D dataset, $\mathcal{D}_2$ (b) Internet Traffic dataset, $\mathcal{D}_3$

Figure 3: Frequency Distributions

Random Queries: For the dataset under consideration, we generate a set of $1,000$ random queries as follows. Each query is generated by choosing its center and its extent. To ensure that queries have some correlation, the query centers are chosen as follows. Let R_1 and R_2 be two $(0.3n)^l$ rectangles (squares) with centers uniformly distributed over $[1,n]^l$, and let $R_3 = [1,n]^l$, for $l = 1, 2$. To specify the center of each query, we first choose a rectangle among the R_1, R_2, or R_3 with probability 0.25, 0.25, and 0.50 respectively, and then generate the query center from the uniform distribution over the chosen rectangle. For the 1D datasets, the extents are chosen from a Gaussian with mean equal to 10% of N and variance 4. For the 2D dataset, the extents on each dimension for the queries is chosen from a Gaussian with mean equal to $n/8 = 8$ and variance 4. We refer to the random queries set for the datasets $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ as RANDOM–1, RANDOM–2 and RANDOM–3 respectively.

Fixed Extent Queries: For a given extent w, and for each dataset, let FXEXT–w be the set of all the queries with extent w. For the case of the 1–dimensional dataset $\mathcal{D}_1$, $w \in [1,n]$, while for the 2–D dataset $\mathcal{D}_2$, $w \in [1,n]^2$. Similarly, let CIRCFXEXT–w the set of all the circular queries with extent w.

6.1.3 Methods

For each dataset we consider several different approaches for comparing results of the proposed approaches.

DWT : The sketching matrix consists of the classic Haar–wavelet vectors corresponding to the top–k wavelet coefficients of h. Note that, in the case of streaming data we do not know before hand, which DWT coefficients are the top–k coefficients. Any algorithm that maintains the top–k DWT coefficients over streams attempts to get approximate results compared to what one would get using this method.

DFT : The sketching matrix consists of the popularly used Fourier vectors corresponding to the top–k Fourier coefficients of h.

EIG : The sketching matrix consists of the top–k eigenvectors of the matrix QQ^*, where Q is the query matrix for the relevant query set.

CIRC : We use the top–k eigenvectors of the circulant matrix corresponding to the CIRCFXEXT–w queries as the sketching matrix. It is computed using Theorems 5 and 3 as discussed in section 5. In the case of fixed extent queries w is chosen to be simply the extent of the queries. In this case, the results we get would be same as when we use EIG . In the case of random queries, we chose w to be the average extent of the queries. Note that this method is advocated because of its succinct sketching matrix which has Fourier vectors.

RP : The sketching matrix consists of pseudo–random 0–1 vectors with a fixed seed as used by Thaper et al [15]. It is important to consider this approach for evaluation because, unless the proposed methods give significantly better results than results using this method, it means that the proposed sketching matrices are just as good or as bad as any random sketching matrix.

6.1.4 Performance Measures

We report the following performance quantities:

REN: the relative error in the norm of the estimate of the frequency distribution, defined as REN $= |\|h\|^2 - \|\widehat{h}\|^2|/\|h\|^2$. The REN measures the amount of "energy" of h captured by $\widehat{h}$, and it is an indicator of how well h can be reconstructed by its sketch; a value of 0 means exact reconstruction. Minimizing this is the measure is the underlying principle for most approximate query answering techniques.

MSE: the mean squared error for the queries, defined as MSE $= \frac{1}{m}\sum_{i=1}^{m}(a_i - \widehat{a_i})^2$.

RLE: the average relative error for all the queries, defined as RLE $= \frac{1}{m}\sum_{i=1}^{m}(|a_i - \widehat{a_i}|/max\{a_i, 1\})$. This is a standard measure for approximate queries.

SEL: the average query selectivity, defined as SEL $= \frac{1}{m}\sum_{i=1}^{m} a_i/|\mathcal{D}|$.

6.2 Experiments

First, we consider random queries described earlier for evaluating different methods. The performance results for the datasets $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ over their corresponding random query sets, are shown in Tables 1, 2 and 3 respectively. The tables present the REN, MSE and RLE measures as the size of the sketch k is increased. We first observe that for all the methods, increasing k improves the REN (expect RP) as well as the MSE. The RLE also generally reduces with larger sketches. All the other methods achieve dramatically better results than the results using RP. We note that in general the DFT and DWT approaches do a better job at preserving the energy of h than the the EIG and CIRC approaches. However, this does not always translate to smaller MSE or RLE errors in answering queries.

For $\mathcal{D}_1$, the EIG and CIRC methods give significantly better MSE and RLE when sketches sizes at least 10% of N. In that case: the EIG gives 37%–84% improvement in MSE and 26%–77% improvement in RLE with respect to DFT. With respect to DWT, for sketches at least 3%, EIG gives 60%–95% improvement in MSE with the exception only when DWT captured at least 99.9% of the norm. It showed 40%–79% improvement in RLE over DWT. So it is important to note that it is not enough to capture the norm of the data distribution in the sketch. Another important observation is that CIRC almost comparable MSEs as EIG and significantly better MSEs than DWT and DFT. Note that CIRC uses just Fourier vectors as the sketching matrix based on CircFxExt–w where w is the mean extent (30 in this case). It is also important to note that while DWT and DFT methods do give better results in some cases, they rely on finding the top–k coefficients of the data distribution, which in the streaming scenario, can be only approximately maintained. In fact when we used incremental approaches to maintain DWT and DFT coefficients using the technique employed by Guha et al [11], we found that they always gave much worse results that EIG and CIRC .

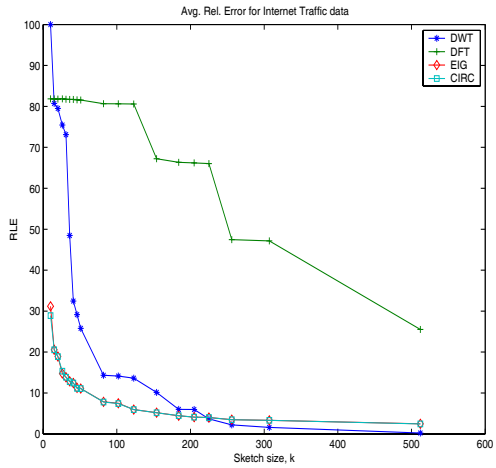
In the case of the real dataset $\mathcal{D}_3$, EIG and CIRC clearly outperform DFT and RP by at least 80% in the case of MSE and 50% on RLE. In the case of DWT, only after DWT captures more than 90% of the norm does it get better results in terms of MSE and RLE. Plots of RLE and REN are shown in Figures 4.

For the 2D synthetic dataset $\mathcal{D}_2$, we report just the results from DWT, DFT and CIRC methods (RP performs poorly and we have seen that CIRC gives comparable results as EIG). For sketch sizes at least 0.5%

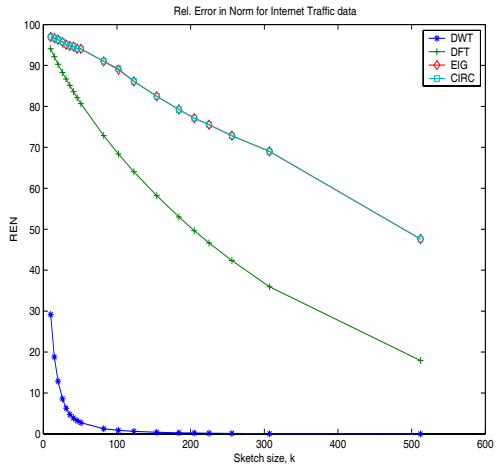
of the N, the CIRC approach gives average RLE/MSE errors that are at least 30% smaller than the errors achieved by either of DFT or DWT, except when DFT and DWT capture at least 95% of the energy of h, in which case the MSE/RLE errors are comparable. Thus, the CIRC approach does quite well at answering approximately random aggregate range queries.

Table 1: Results for the random query set RANDOM–1 over the 1D real dataset $\mathcal{D}_1$ as we increase the sketch size k. $N = 300$. Mean extent of queries = 10%. over the The average selectivity of the queries, SEL is 12.14%. MSE for RP is reported in 10^3.

k/N%	k	RP			DWT			DFT			EIG			CIRC		
		REN%	MSE	RLE%	REN%	MSE	RLE%	REN%	MSE	RLE%	REN%	MSE	RLE%	REN%	MSE	RLE%
1.0	3	92.40	14300	649.77	16.58	242293	128.16	20.07	312611	111.55	67.04	842882	73.22	20.07	312611	111.55
2.0	6	75.78	6796	558.95	8.75	91579	70.00	4.43	35490	24.94	60.77	706040	46.83	6.27	55713	28.57
3.0	9	79.81	5787	775.48	4.57	13320	17.21	1.92	2378	10.11	2.08	1728	8.07	1.92	2378	10.11
4.0	12	45.87	2859	633.95	3.06	5836	11.30	1.63	156	2.32	1.86	315	2.83	1.65	143	1.67
5.0	15	86.87	37646	766.81	2.48	3329	10.15	1.57	156	2.46	1.81	162	2.11	1.65	124	1.43
10.0	30	52.79	1602	465.74	1.50	904	7.07	1.31	125	2.00	1.71	78	1.47	1.59	82	1.39
15.0	45	32.88	592	242.62	1.05	552	4.99	1.10	120	2.01	1.63	57	1.18	1.47	55	1.46
20.0	60	36.02	423	105.48	0.76	182	4.52	0.92	115	2.03	1.49	38	0.80	1.42	50	1.38
25.0	75	20.46	283	107.84	0.56	160	4.47	0.77	107	2.07	1.42	32	0.62	1.29	44	1.64
30.0	90	13.60	223	77.79	0.40	74	2.06	0.63	86	2.05	1.34	26	0.63	1.19	37	1.48
50.0	150	1.14	166	58.28	0.08	9	0.90	0.29	65	2.20	0.91	10	0.50	0.90	25	1.22



(a) Sketch Size Vs Ave. Rel. Error (RLE) for $\mathcal{D}_3$



(b) Sketch Size Vs Rel. Err. in Norm (REN) for $\mathcal{D}_3$

Figure 4: Performance Results for Random Queries over Internet Traffic Data.

Next, we evaluate fixed extent queries over the datasets $\mathcal{D}_1$ and $\mathcal{D}_2$. We report results for DWT, DFT and CIRC methods (RP performs poorly and EIG gives almost same results as CIRC . It would have given exact same results as CIRC when we use CIRCFXEXT–w instead of FXEXT–w). The performance results for $\mathcal{D}_1$ are shown in Tables 4. Here, we kept the sketch size fixed to 30, and varied the extent of the queries. This sketch size is sufficient for the DFT/DWT approaches to capture over 99% of the energy of h; the CIRC is also capturing over 99% of h's energy. Note that by increasing the extent of the queries, their selectivity

Table 2: Results for the random query set RANDOM–3 over the 1D real dataset $\mathcal{D}_3$ as we increase the sketch size k. $N = 1024$. Mean extent of queries = 10%. over the The average selectivity of the queries, SEL is 9.96%. All MSEs are reported in 10^3.

k/N %	k	RP			DWT			DFT			EIG			CIRC		
		REN %	MSE	RLE %	REN %	MSE	RLE %	REN %	MSE	RLE %	REN %	MSE	RLE %	REN %	MSE	RLE %
1.0	10	16.98	62872	671.27	29.16	2577	100.74	94.07	798	81.85	96.95	141	31.17	96.97	147	28.90
2.0	20	4.00	22921	450.28	12.88	749	79.49	90.29	797	81.78	96.28	67	18.92	96.31	68	18.90
3.0	31	9.44	17487	445.71	6.29	644	73.10	86.68	795	81.75	95.08	39	13.86	95.08	39	13.86
4.0	41	17.29	27899	515.10	3.86	132	32.46	83.62	794	81.69	94.62	34	12.44	94.62	34	12.44
5.0	51	8.51	10894	338.97	2.75	76	25.79	80.68	792	81.55	94.05	29	11.08	94.05	29	11.08
8.0	82	5.12	10831	285.20	1.27	20	14.33	72.87	781	80.66	91.00	19	7.85	91.04	19	7.80
10.0	102	8.35	8565	279.83	0.88	19	14.14	68.38	781	80.64	88.95	15	7.49	89.21	16	7.43
12.0	123	11.23	7034	257.23	0.63	18	13.62	64.04	780	80.58	86.13	12	5.95	86.13	12	5.95
15.0	154	9.02	6001	254.96	0.41	10	10.15	58.19	582	67.21	82.47	9	5.20	82.47	9	5.18
20.0	205	7.64	5119	233.38	0.22	4	5.98	49.64	568	66.20	77.14	6	4.10	77.14	6	4.10
25.0	256	9.24	3791	193.34	0.12	1	2.21	42.35	379	47.46	72.84	5	3.48	72.83	5	3.47
30.0	307	5.43	2448	151.26	0.07	0	1.59	35.95	377	47.14	69.02	4	3.33	69.02	4	3.33
50.0	512	8.66	2886	159.74	0.01	0	0.23	17.92	72	25.52	47.59	2	2.47	47.59	2	2.46

is generally increasing as well. Despite the fact that the DFT/DWT captured the energy of h very well, for queries with selectivity less than 10%, their average RLE is typically twice as big as the RLE achieved by the CIRC . The MSE achieved by the DFT/DWT approaches is 30–50% more than that of the EIG approach.

For the 2–D dataset, the performance results are given in Table 5. In this case, the EIG approach gives comparable results with those of DWT/DFT inspite of not capturing as much energy of h as them. The CIRC attempts to use sketching matrices that approximate the queries well, while the DFT/DWT attempts to use sketching matrices that capture the energy of a frequency distribution h well. The above experiments indicate that one can approximately answer range queries with reasonable accuracy without capturing almost all of the energy of h. It does matter how the energy of h is captured.

Finally, we conduct experiments to evaluate the idea of extending a DFT linear sketching matrix by an CIRC sketching matrix according to the method discussed in section 4.4. We present results for the RANDOM–1 queries over the 1D dataset $\mathcal{D}_1$, with the following sketching matrices: the $N \times k_1$ sketching

Table 3: Results for the RANDOM–2D queries over the 2–D dataset $\mathcal{D}_2$ ($N = 64^2 = 4096$) for a sketch size of k. The average selectivity of the queries is 1.93%.

k/N %	k	DWT			DFT			CIRC		
		REN %	MSE	RLE %	REN %	MSE	RLE %	REN %	MSE	RLE %
0.10	4	34.38	1116.23	53.81	35.06	754.84	36.20	93.97	704.74	36.10
0.20	8	27.18	753.25	37.37	30.15	800.57	40.91	64.59	612.35	30.95
0.39	16	18.24	443.94	22.42	23.13	675.12	37.04	42.59	410.48	18.90
0.78	32	9.73	338.13	18.13	13.42	350.30	22.50	31.46	269.41	12.95
1.56	64	5.51	111.07	7.12	4.69	117.99	8.19	9.49	121.77	5.44
3.13	128	3.19	38.06	2.26	2.25	30.63	2.16	6.35	60.70	3.06
6.25	256	1.86	15.68	1.04	1.63	14.79	0.98	4.53	47.15	2.58
12.50	512	1.03	5.02	0.38	1.26	10.59	0.71	3.61	19.14	1.21

Table 4: Results for FxExt–w queries, for various w, over the 1–D dataset $\mathcal{D}_1$, with fixed sketch size $k = 30$. The REN for DWT, DFT, and EIG is 0.43%, 0.31%, and 0.55%.

FxExt–w queries			DWT		DFT		CIRC	
w/N%	w	SEL %	MSE	RLE %	MSE	RLE %	MSE	RLE %
1	3	1.01	8.04	17.13	7.62	19.03	7.23	9.75
2	6	2.03	11.42	13.35	9.76	11.48	8.13	5.88
5	15	5.24	16.39	7.62	11.81	5.58	6.27	1.74
10	30	11.03	20.86	2.25	9.64	1.45	7.25	1.01
20	60	24.18	28.93	1.06	11.03	0.42	7.55	0.29
	$2 \leq w \leq 3$	0.84	7.29	18.62	6.95	21.38	6.75	11.09
	$2 \leq w \leq 5$	1.35	9.17	16.09	8.30	16.67	7.48	8.57
	$2 \leq w \leq 15$	2.90	12.45	12.21	10.09	10.71	7.27	5.15
	$2 \leq w \leq 30$	5.60	15.62	8.15	10.48	6.86	7.40	3.18

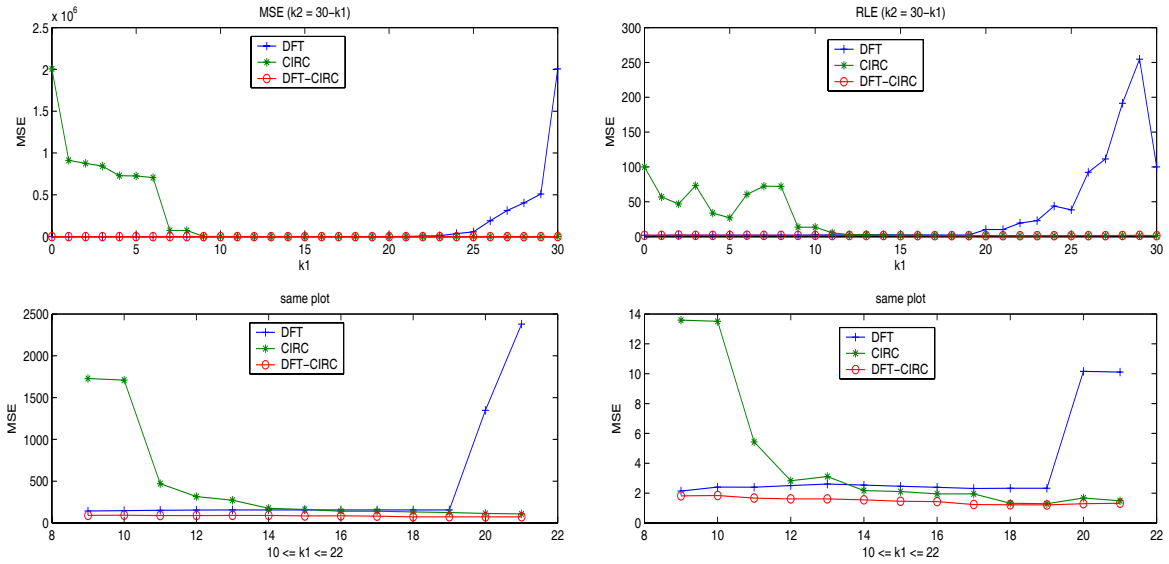
Table 5: Results for FxExt–w queries, for various extents w, over 2–D dataset $\mathcal{D}_2$, with sketch size $k = 65$.

FxExt–w queries		DWT			DFT			CIRC		
w	SEL %	REN %	MSE	RLE %	REN %	MSE	RLE %	REN %	MSE	RLE %
9x9	2.21	5.45	127.93	5.61	4.57	102.31	5.24	5.43	149.07	7.98
14x14	5.67	5.45	213.44	3.61	4.57	79.98	1.49	10.14	194.45	3.85
20x20	11.97	5.45	265.95	2.26	4.57	178.10	1.56	15.94	223.46	2.04
29x29	24.98	5.45	350.75	1.34	4.57	193.20	0.81	13.14	236.52	1.01
9x9 and 14x14	3.66	5.45	168.64	4.69	4.57	97.32	3.38	5.43	135.66	5.02

matrix P_1 of the DFT method (DFT–k_1), the $N \times k_2$ sketching matrix P_e of the CIRC (CIRC –k_2), and $N \times (k_1 + k_2)$ sketching matrix $P = [P_1 P_2]$ (DFT–CIRC –$[k_1, k_2]$), where P_2 is computed by the CIRC for the ${P_1}^{\perp} Q$ matrix. The results of this experiment are shown in Table 6 and Figures 5(a) and (b). Using the extended sketch DFT–CIRC always obtains lesser MSE and RLE than DFT and CIRC methods. We already proved that this must be the case with MSE. The DFT–CIRC method reduces the MSE obtained when using just DFT by as much as 40% when 5 DFT coefficients and 25 coefficients of the extended sketch (DFT–CIRC –$[5, 25]$) is used and RLE by 34%. Thus constructing a sketch by using both the frequency distribution and the queries together, makes it possible to get much better approximate answers to queries than using just the frequency distribution or just the queries. This experiment successfully demonstrates that the novel approach that we proposed of extending linear sketches is beneficial and encourages exploration of this approach.

Table 6: Results for the RANDOM–1 queries over the 1D dataset $\mathcal{D}_1$, with the following sketching matrices: the $N \times k_1$ sketching matrix P_1 of the DFT method (DFT–k_1), the $N \times k_2$ sketching matrix P_e of the CIRC (CIRC –k_2), and $N \times (k_1 + k_2)$ sketching matrix $P = [P_1 P_2]$ (DFT–CIRC –$[k_1, k_2]$), where P_2 is computed by the CIRC for the ${P_1}^{\perp} Q$ matrix, with Q the query matrix for RANDOM–.

Sketch sizes		DFT–k_1		EIG –k_2		DFT–EIG –$[k_1, k_2]$	
k_1	k_2	MSE	RLE %	MSE	RLE %	MSE	RLE %
30	0	125.62	2.01	2005109.25	100.00	125.62	2.01
25	5	135.33	2.05	56147.14	38.38	101.86	2.01
20	10	148.31	2.41	1347.70	10.16	93.15	1.84
15	15	156.01	2.46	124.62	1.43	85.40	1.46
10	20	1347.70	10.16	109.97	1.79	73.61	1.29
5	25	56147.14	38.38	89.60	1.41	71.43	1.29
0	30	2005109.25	100.00	82.92	1.40	78.07	1.47



(a) MSE (c) RLE

Figure 5: Performance of DFT, CIRC and DFT-CIRC with random queries on $\mathcal{D}_1$.

7 Conclusions

We consider the problem of answer aggregate range queries approximately. We presented an approach for constructing linear sketches for frequency distributions of data that take into account the set of (counting) range queries to be answered. We show how to extend a linear sketch to also take into consideration such queries. Furthermore, we show how to construct sketching matrices with small foot–print that consider range queries of various extents. Experimental results demonstrate that the proposed approaches provides significant improvements on the approximation errors (MSE and RLE) for such queries, with respect to DFT/DWT approaches of the frequency distributions.

References

[1] The internet traffic archive: Epa-http. http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html.

[2] Ashraf Aboulnaga and Surajit Chaudhuri. Self-tuning histograms: building histograms without looking at data. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 181–192. ACM Press, 1999.

[3] Shivnath Babu and Jennifer Widom. Continuous queries over data streams. Technical report, Stanford University, 2001.

[4] Nicolas Bruno, Surajit Chaudhuri, and Luis Gravano. Stholes: a multidimensional workload-aware histogram. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 211–222. ACM Press, 2001.

[5] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. Technical Report TR-99-006, Berkeley, CA, 1999.

[6] Philip J Davis. *Circulant Matrices*. John Wiley & Sons, Inc., 1979.

[7] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[8] Phillip B. Gibbons and Yossi Matias. Synopsis data structures for massive data sets. *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science: Special Issue on External Memory Algorithms and Visualization*, A, 1999.

[9] Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin Strauss. Quicksand: Quick summary and analysis of network data. Technical report, DIMACS, 2001.

[10] Goetz Graefe. Query evaluation techniques for large databases. *ACM Computing Surveys*, 25(2):73–170, 1993.

[11] Sudipto Guha, Piotr Indyk, S. Muthukrishnan, and Martin Strauss. Histogramming data streams with fast per-item processing. In *ICALP 2002*, pages 681–692.

[12] Peter Lancaster and Miron Tismenetsky. *The Theory of Matrices with Applications*, volume Computer Science and Applied Mathematics of *A Series of Monographs and Textbooks*. Academic Press, Orlando, FL, 2^{nd} edition, 1985.

[13] Andrew. M. Odlyzko. Internet traffic growth: Sources and implications. In *ITCOM 2003, SPIE, 2003*, 2003.

[14] Viswanath Poosala. *Histogram-Based Estimation Techniques in Database Systems*. PhD thesis, University of Wisconsin, Madison, Wisconsin, U.S.A., 1997.

[15] Nitin Thaper, Sudipto Guha, Piotr Indyk, and Nick Koudas. Dynamic multidimensional histograms. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 428–439. ACM Press, 2002.

[16] Jeffrey Scott Vitter, Min Wang, and Bala Iyer. Data cube approximation and histograms via wavelets. In Georges Gardarin, James C. French, Niki Pissinou, Kia Makki, and Luc Bouganim, editors, *Proceedings of the 7th ACM International Conferences on Information and Knowledge Management*, pages 96–104, New York, U.S.A., 1998. Association for Computer Machinery.

Contents