Data Extraction via Histogram and Arithmetic Mean Queries: Fundamental Limits and Algorithms

I-Hsiang Wang, Shao-Lun Huang, Kuan-Yun Lee, and Kwang-Cheng Chen
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
Email: {ihwang,huangntu,b01901024,ckc}@ntu.edu.tw

***Abstract*—The problems of extracting information from a data set via histogram queries or arithmetic mean queries are considered. We first show that the fundamental limit on the number of histogram queries, m, so that the entire data set of size n can be extracted losslessly, is $m = \Theta(n/\log n)$, sub-linear in the size of the data set. For proving the lower bound (converse), we use standard arguments based on simple counting. For proving the upper bound (achievability), we proposed two query mechanisms. The first mechanism is random sampling, where in each query, the items to be included in the queried subset are uniformly randomly selected. With random sampling, it is shown that the entire data set can be extracted with vanishing error probability using $\Omega(n/\log n)$ queries. The second one is a non-adaptive deterministic algorithm. With this algorithm, it is shown that the entire data set can be extracted exactly (no error) using $\Omega(n/\log n)$ queries. We then extend the results to arithmetic mean queries, and show that for data sets taking values in a real-valued finite arithmetic progression, the fundamental limit on the number of arithmetic mean queries to extract the entire data set is also $\Theta(n/\log n)$.**

I. Introduction

Efficiently and effectively acquiring information from large-scale data sets is an important step in analyzing a huge amount of data. In general, a common process for data analysts to acquire data from a data set can be stated as follows:

- A data analyst sends queries to a data curator, who is in charge of releasing data.
- The data curator responds to the queries with answers based on the data in the data set as well as the queries.

In this process, the type of queries that the data analysts can send typically depends on the applications. For the sake of privacy protection, the data in many systems are released in the form of certain statistics such as *histograms* or *averages*. For example, in a medical system, we can often only acquire the statistics of patients about certain diseases, but not the information about individual patients.

In this paper, our goal is to understand how many and what kinds of queries are required to extract all entries in the data set. Such information is particularly useful for data analysts, since the total cost of data extraction often grows with the number of queries sent to the data curator. It is also useful for the data curator to protect the data set from query-aggregation attacks. Therefore, the fundamental limit on the necessary number of queries and the corresponding querying algorithms are important to both data analysts and data curators.

Specifically, we investigate the fundamental limit on the number of queries, m, required to extract the entire n-item data set "losslessly", when the data curator responds *honestly* (without noise). The data set consists of n items labeled from 1 to n, and each item takes the *value* in a finite alphabet $\mathcal{A}$ of size d, a collection of abstract symbols or real numbers. For simplicity, in this conference paper we assume d is fixed and does not scale with n. Two kinds of queries are considered:

- *Histogram* Queries: the analyst queries a subset of items in the data set, and the data curator releases the histogram of the values of these queried items.
- *Arithmetic Mean* Queries: the analyst queries a subset of items in the (real-valued) data set, and the data curator releases the arithmetic mean of the values of these items.

Note that a trivial upper bound on m is n, since one can query each item one by one and extract the entire data set. The question is, can we leverage histogram/arithmetic mean queries of larger subsets to significantly reduce the necessary number of queries? The answer turns out to be yes. We prove that for both histogram queries with arbitrary alphabet $\mathcal{A}$ and arithmetic mean queries with $\mathcal{A}$ being a real-valued finite arithmetic progression, the fundamental limit is $m = \Theta(n/\log n)$, sub-linear in the size of the data set.

For the impossibility part (lower bound on query complexity m), we use simple counting arguments to show that if $m = o(n/\log n)$, lossless extraction is impossible. For the achievability part, two kinds of mechanisms to choose the queried subsets are considered:

- *Deterministic Sampling*: The queried subsets are determined beforehand. For deterministic sampling, it is required to extract the entire data set exactly no matter what the values of the data sets are.
- *Random Sampling*: In each query, the items to be included in the queried subset are randomly chosen. The extraction criterion is to have vanishing error probability as $n \to \infty$, no matter what the values of the data sets are.

Note that both mechanisms are *non-adaptive*: a queried subset does not depend on the responses to the previous queries.

For deterministic sampling, we first propose an explicit non-adaptive algorithm and show that it extracts the entire data set exactly (no error) using $\Omega(n/\log n)$ histogram queries. Then for arithmetic mean queries, we focus on the setting where the alphabet $\mathcal{A}$ is a real-valued finite arithmetic progression. Using the proposed deterministic data extraction algorithm for histogram queries as the building block, another non-adaptive extraction algorithm for arithmetic mean queries is proposed,

978-1-5090-1806-2/16/$31.00 ©2016 IEEE

and it is shown that with $\Omega(n/\log n)$ queries, the entire data set can be extracted without any error. For random sampling, we focus on histogram queries and analyze the probability of error under uniform i.i.d. random sampling. It is shown that the entire data set can be extracted with vanishing error probability using $\Omega(n/\log n)$ queries.

II. Problem Formulation

We shall cast the data extraction problem with n items and m queries as a linear inverse problem:

Solving n unknowns with m linear equations.

Notations: Let $[N_1 : N_2] \triangleq \{N_1, N_1+1, \ldots, N_2\}$ for integers $N_1 < N_2$, and $[N] \triangleq \{1, 2, \ldots, N\}$ for $N \in \mathbb{N}$. Let $(\cdot)^\intercal$ denote the transpose operation.

A. Data Set

Consider a *data set* with n items, labeled from 1 to n. Each item has a piece of data taking values in a finite *alphabet* $\mathcal{A} = \{a_1, a_2, \ldots, a_d\}$, and $|\mathcal{A}| = d$. We model the data set as a matrix $\mathbf{X} \triangleq \begin{bmatrix}\mathbf{x}_1^\intercal & \mathbf{x}_2^\intercal & \ldots & \mathbf{x}_n^\intercal\end{bmatrix}^\intercal$, with n rows $\mathbf{x}_1, \ldots, \mathbf{x}_n$. For different query models, the values of row vectors $\mathbf{x}_i$'s are defined differently.

1) Histogram Query: For the histogram-query problem, for all $i \in [n]$, $\mathbf{x}_i = \mathbf{e}_l$ if and only if the i-th item in the data set takes value at a_l. Here $\mathbf{e}_l$ denotes the l-th unit row vector of the d-dimensional Euclidean space, $l = 1, \ldots, d$. In other words, the value of $\mathbf{x}_i$ indicates which symbol $a_l \in \mathcal{A}$, $l \in [d]$, the i-th item takes. The data set is viewed as an $n \times d$ matrix $\mathbf{X} \in \{0,1\}^{n\times d}$, where in each column there is only one non-zero entry. Since the rows are limited to the standard basis $\mathcal{B}_d \triangleq \{\mathbf{e}_l \,|\, l \in [d]\}$, we denote the range of $\mathbf{X}$ by $\mathcal{B}_d^{n\times 1}$.

2) Arithmetic Mean Query: For the arithmetic-mean-query problem, we assume that $\mathcal{A} \subset \mathbb{R}$ and the d (ordered) elements $a_1 \le a_2 \le \ldots \le a_d$ form an arithmetic progression. In this case, $\mathbf{x}_i$ is simply the value of the i-th item of the data set and hence the range of the data-set matrix $\mathbf{X}$ is $\mathcal{A}^{n\times 1}$.

B. Queries and Responses

Consider m queries and each query is a subset of labels in $[n]$. Let $\mathcal{S}_i$ denote the queried subset in the i-th query. We shall use an $m \times n$ query matrix $\mathbf{Q} \in \{0,1\}^{m\times n}$ to collectively represent the m queries. In particular, $(\mathbf{Q})_{i,j} = 1$ if and only if the j-th item is included in the i-th queried subset. In other words, $(\mathbf{Q})_{i,j} = \bullet\ \{j \in \mathcal{S}_i\}$. Hence, the i-th row $\mathbf{q}_i$ represents the queried subset in the i-th query.

The response to the queries in both scenarios can then be represented as the multiplication of query matrix and the data-set matrix. Let us denote the responses to the m queries by m row vectors $\mathbf{y}_1, \ldots, \mathbf{y}_m$, and the corresponding m-row matrix by $\mathbf{Y} \triangleq \begin{bmatrix}\mathbf{y}_1^\intercal & \ldots & \mathbf{y}_m^\intercal\end{bmatrix}^\intercal$.

1) Histogram Query: For the i-th histogram query,

$$\mathbf{y}_i = \left[\binom{\#\text{ of } a_1}{\text{in } \mathcal{S}_j} \quad \binom{\#\text{ of } a_2}{\text{in } \mathcal{S}_j} \quad \cdots \quad \binom{\#\text{ of } a_d}{\text{in } \mathcal{S}_j}\right].$$

It is not hard to see that $\mathbf{y}_i = \sum_{j\in\mathcal{S}_i} \mathbf{x}_j$ and hence

$$\mathbf{Y} = \mathbf{Q}\mathbf{X},\ \mathbf{Y} \in [0:n]^{m\times d}.$$

2) Arithmetic Mean Query: For arithmetic mean query, since the data analyst who sends out the query knows the queried subset and hence the number of items, the response is equivalent to the arithmetic sum. Hence, if $\mathbf{y}_i \in \mathbb{R}$ denotes the arithmetic sum of the i-th queried subset $\mathcal{S}_i$, it is straightforward to see that

$$\mathbf{Y} = \mathbf{Q}\mathbf{X},\ \mathbf{Y} \in \mathbb{R}^{m\times 1}.$$

C. Data Extraction as a Linear Inverse Problem

Let us summarize the above formulation as follows.

1) Data-set matrix: $\mathbf{X} \in \mathcal{B}_d^{n\times 1}$ in the histogram-query case, and $\mathbf{X} \in \mathcal{A}^{n\times 1}$ in the arithmetic-mean-query case.
2) Query matrix: $\mathbf{Q} \in \{0,1\}^{m\times n}$, $(\mathbf{Q})_{i,j} = \bullet\ \{j \in \mathcal{S}_i\}$, where $\mathcal{S}_i$ denotes the queried subset in the i-th query.
3) Response matrix: $\mathbf{Y} = \mathbf{Q}\mathbf{X}$, where $\mathbf{Y} \in [0:n]^{m\times d}$ in the histogram-query case and $\mathbf{Y} \in \mathbb{R}^{m\times 1}$ in the arithmetic-mean-query case.

Now, the data extraction problem for both cases is to reconstruct the data-set matrix $\mathbf{X}$ based on the response matrix $\mathbf{Y}$ and the query matrix $\mathbf{Q}$. In other words, it is equivalent to solving n unknowns with m linear equations, i.e., a linear inverse problem. In general, it requires $m = n$ linear equations to solve n unknowns. However, as we will show in our main results, by making use of the structure of the data-set matrix $\mathbf{X}$, even though the query matrix $\mathbf{Q}$ is limited, we are able to solve n unknowns with sub-linear-in-n linear equations.

D. Querying Mechanisms and Criteria of Data Extraction

The goal is to recover the data-set matrix $\mathbf{X}$ *losslessly*. Two kinds of mechanisms and recovery criteria are considered.

1) Deterministic Sampling: The query matrix $\mathbf{Q}$ is deterministic and set beforehand. The criterion of lossless data extraction is to recover $\mathbf{X}$ exactly. Hence, it is required that

$$\tilde{\mathbf{X}} \neq \mathbf{X} \implies \mathbf{Q}\tilde{\mathbf{X}} \neq \mathbf{Q}\mathbf{X}. \tag{1}$$

Definition 2.1 (Recoverability): Suppose a query matrix $\mathbf{Q} \in \{0,1\}^{m\times n}$ satisfies (1) for all data-set matrix $\mathbf{X}$, then it is called (m,n)*-recoverable*.

2) Random Sampling: Here the query matrix $\mathbf{Q}$ is random. To specify the criterion of lossless data extraction under random sampling, let us define *probability of error* associated with the randomly generated $\mathbf{Q}$ as follows.

Definition 2.2 (Probability of Error): For a data-set matrix $\mathbf{X}$, its probability of error under query matrix $\mathbf{Q}$ is defined as

$$\mathsf{P_e}(\mathbf{X};\mathbf{Q}) \triangleq \mathbb{P}_{\mathbf{Q}}\left\{\exists\, \tilde{\mathbf{X}} \neq \mathbf{X} \text{ such that } \mathbf{Q}\tilde{\mathbf{X}} = \mathbf{Q}\mathbf{X}\right\}.$$

The subscript "$(\cdot)_{\mathbf{Q}}$" is to emphasize that the probability is induced by the randomly generated query matrix $\mathbf{Q}$.

Given a sequence of randomly generated query matrices $\{\mathbf{Q}^{m,n}\}$ (m grows with n), lossless data extraction is to ensure vanishing probability of error as $n \to \infty$:

$$\lim_{n\to\infty} \max_{\mathbf{X}\in\mathcal{X}^{n\times 1}} \mathsf{P_e}\left(\mathbf{X};\mathbf{Q}^{(m,n)}\right) = 0. \tag{2}$$

Here $\mathcal{X} = \mathcal{B}_d$ and $\mathcal{A}$ for histogram-query and arithmetic-mean-query settings respectively.

III. Main Results

First we give a lemma on the impossibility part.

Lemma 3.1 (Impossibility Results): For any sequence of query matrices $\{\mathbf{Q}^{(m,n)}\}$, if $m = o(n/\log n)$, then lossless extraction is impossible for the histogram-query setting and the arithmetic-mean-query setting.

Proof: Note that for both deterministic sampling and random sampling, the respective lossless extraction criteria (1) and (2) will fail as long as the total number of possible $\mathbf{Y}$ is strictly smaller than the total number of possible $\mathbf{X}$.

For histogram queries, the total number of possible $\mathbf{X}$ is d^n, while the total number of possible $\mathbf{Y}$ is at most $(n+1)^{(d-1)m}$. For arithmetic mean queries, the total number of possible $\mathbf{X}$ is d^n, while the total number of possible $\mathbf{Y}$ is at most $(n(d-1)+1)^m$. Hence,

$$(n+1)^{(d-1)m} < d^n \iff m < \left(\tfrac{\log d}{d-1}\right)\tfrac{n}{\log(n+1)};$$
$$(n(d-1)+1)^m < d^n \iff m < \tfrac{n\log d}{\log(n(d-1)+1)}.$$

Since d is a constant with respect to n, proof complete. ■

Next, we summarize the achievability results in the following two theorems.

Theorem 3.1 (Achievability of Deterministic Sampling): We propose an explicit construction of query matrix $\mathbf{Q}$ such that the exact data extraction criterion (1) is satisfied with $m = \Theta(n/\log n)$ histogram queries. An extension of this construction attains exact data extraction with $m = \Theta(n/\log n)$ arithmetic mean queries if the alphabet $\mathcal{A}$ forms a real-valued arithmetic progression.

Proof: See Section IV for the construction. The idea is that, by leveraging the structure of the domain of solution, one can reduce the number of linear equations needed. ■

Theorem 3.2 (Achievability of Random Sampling): In the histogram-query setting, if one generates the query matrix $\mathbf{Q}^{(m,n)}$ according to the following distribution:

$$\left(\mathbf{Q}^{(m,n)}\right)_{i,j} \overset{\text{i.i.d.}}{\sim} \text{Ber}(1/2),\ \forall i \in [m],\ \forall j \in [n].$$

Then, $m = \Omega(n/\log n)$ implies that the lossless extraction criterion (2) is satisfied.

Proof: The proof involves detailed analysis on the probability of error. We find upper bounds on $\mathsf{P_e}\left(\mathbf{X}; \mathbf{Q}^{(m,n)}\right)$ from first principles. Details are given in Section V. ■

IV. Deterministic Sampling

Let us first illustrate the basic idea in our construction with a simple example in the histogram-query setting.

Example 4.1 (Extract 4 Items with 3 Histogram Queries): Consider the following query matrix

$$\mathbf{Q}^{(3,4)} \triangleq \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

The key is to apply the following row operation on $\mathbf{Q}^{(3,4)}$:

$$\text{Row } 1 - \text{Row } 2 + \text{Row } 3 = \begin{bmatrix} 0 & 2 & 0 & 1 \end{bmatrix}.$$

which can be translated to the same row operation on the response matrix $\mathbf{Y}$, and get a row $\mathbf{y}' = 2\mathbf{x}_2 + \mathbf{x}_4$. Since each row of $\mathbf{X}$ is a unit vector, we can uniquely determine $\mathbf{x}_2$ and $\mathbf{x}_4$ from $\tilde{\mathbf{y}}$. Then we can determine $\mathbf{x}_1$ and $\mathbf{x}_3$ from $\mathbf{y}_1$ and $\mathbf{y}_2$ (the first and the second rows of $\mathbf{Y}$) successively.

In the above example, even though the construction of the query matrix $\mathbf{Q}$ is quite limited (entries of $\mathbf{Q}$ are constrained in $\{0,1\}$), we are able to save 1 equation in solving 4 unknowns, by harnessing the structure of the domain of solution (each row of $\mathbf{X}$ is a unit vector).

A. Proof of Theorem 3.1 for Histogram Queries

In the following, let us focus on the histogram-query setting and propose a recursive way to construct larger query matrices when n grows. Suppose we have a construction $\mathbf{Q}^{(m,n)}$ that is (m,n)-recoverable, and consider the construction:

$$\mathbf{Q}^{(3m,3n+m)} \triangleq \begin{bmatrix} \mathbf{Q}^{(m,n)} & \mathbf{Q}^{(m,n)} & \mathbf{0}_{m\times n} & \mathbf{0}_m \\ \mathbf{Q}^{(m,n)} & \mathbf{0}_{m\times n} & \mathbf{Q}^{(m,n)} & \mathbf{0}_m \\ \mathbf{0}_{m\times n} & \mathbf{Q}^{(m,n)} & \mathbf{Q}^{(m,n)} & \mathbf{I}_m \end{bmatrix}.$$

Lemma 4.1: If $\mathbf{Q}^{(m,n)}$ is (m,n)-recoverable, then the constructed $\mathbf{Q}^{(3m,3n+m)}$ is $(3m,3n+m)$-recoverable.

Proof: Consider the following elementary row operations on $\mathbf{Q}^{(3m,3n+m)}$ (drop the superscript in the following for simplicity):

$$\begin{aligned} &\mathbf{Q}_{[1\downarrow m]} - \mathbf{Q}_{[(m+1)\downarrow 2m]} + \mathbf{Q}_{[(2m+1)\downarrow 3m]} \\ &= \begin{bmatrix} \mathbf{0}_{m\times n} & 2\mathbf{Q}^{(m,n)} & \mathbf{0}_{m\times n} & \mathbf{I}_m \end{bmatrix} \end{aligned} \tag{3}$$

Here for a matrix $\mathbf{M}$, we use the notation $\mathbf{M}_{[i\downarrow j]}$ to denote the sub-matrix formed by the i-th to the j-th rows of $\mathbf{M}$.

Applying the same row operations on the corresponding response matrix $\mathbf{Y}$, we get

$$\mathbf{Y}' = \begin{bmatrix} \mathbf{0}_{m\times n} & 2\mathbf{Q}^{(m,n)} & \mathbf{0}_{m\times n} & \mathbf{I}_m \end{bmatrix} \mathbf{X}.$$

Hence we can recover $\mathbf{X}_{[(3n+1)\downarrow(3n+m)]}$ and

$$\mathbf{Q}^{(m,n)}\mathbf{X}_{[1\downarrow n]},\ \mathbf{Q}^{(m,n)}\mathbf{X}_{[(n+1)\downarrow 2n]},\ \mathbf{Q}^{(m,n)}\mathbf{X}_{[(2n+1)\downarrow 3n]}.$$

By the original assumption that $\mathbf{Q}^{(m,n)}$ is (m,n)-recoverable, we can also recover $\mathbf{X}_{[1\downarrow n]}$, $\mathbf{X}_{[(n+1)\downarrow 2n]}$, and $\mathbf{X}_{[(2n+1)\downarrow 3n]}$. ■

We are now ready to prove Theorem 3.1 for the histogram-query setting. Using Lemma 4.1, if we start at $(m,n) = (0,1)$ being the 0-th iteration, at the t-th iteration we shall have $m_t = 3^t$ and $n_t = 3n_{t-1} + 3^{t-1} = 9n_{t-2} + 3^{t-1} + 3^{t-1} = \cdots = 3^t + t3^{t-1} = (t+3)3^{t-1}$. Hence, we have

$$\lim_{t\to\infty} \frac{m\log_3 n}{n} = \lim_{t\to\infty} \frac{3^t(\log_3(t+3)+t-1)}{(t+3)3^{t-1}} = 3.$$

We have shown that for $n = (t+3)3^{t-1}$, $t \in \mathbb{N}$, the construction yields $m = \Theta(n/\log n)$. For $n \neq (t+3)3^{t-1}$ for all $t \in \mathbb{N}$, pick $t \in \mathbb{N}$ such that $n_t < n < n_{t+1}$ where $n_t = (t+3)3^{t-1}$. Since $n < n_{t+1}$, we are able to extract the data set by using at most $m_{t+1} = 3m_t$ histogram queries, by first inserting "dummy" items to enlarge the original n-item data set to a n_{t+1}-element data set, and then recovering them using $\mathbf{Q}^{(m_{t+1},n_{t+1})}$.

Note that $\frac{3m_t \log_3 n_{t+1}}{n_{t+1}} \le \frac{3m_t \log_3 n}{n} \le \frac{3m_t \log_3 n_t}{n_t}$ and

$$\lim_{t\to\infty} \frac{3m_t \log_3 n_{t+1}}{n_{t+1}} = \lim_{t\to\infty} \frac{3\times 3^t(\log_3(t+4)+t)}{(t+4)3^t} = 3,$$
$$\lim_{t\to\infty} \frac{3m_t \log_3 n_t}{n_t} = \lim_{t\to\infty} \frac{3\times 3^t(\log_3(t+3)+t-1)}{(t+3)3^{t-1}} = 9.$$

Therefore, our construction yields $m = \Theta(n/\log n)$.

B. Proof of Theorem 3.1 for Arithmetic Mean Queries

Recall that the alphabet $\mathcal{A} = \{a_1, \ldots, a_d\}$ forms a real-valued arithmetic progression $a_1 < \cdots < a_d$. Since the response to a queried subset is equivalent to the arithmetic sum, we can assume without loss of generality that $\mathcal{A} = [0 : d-1]$.

Note that for $d = 2$, the data extraction problems for both the histogram-query and the arithmetic-mean-query settings are identical. Hence, we can use the scheme in Section IV-A directly. In the following, we first prove a lemma about the achievability for $d = 2^h$, $h \in \mathbb{N}$. Then, for general d, we consider an enlarged alphabet $\overline{\mathcal{A}} \triangleq \left[0 : 2^h - 1\right]$, $h = \lceil \log_2 d \rceil$, and use the scheme developed in the lemma for $d = 2^h$.

Lemma 4.2: Let the alphabet $\mathcal{A} = \left[0 : 2^h - 1\right]$ for a fixed $h \in \mathbb{N}$, not scaling with n. If $m = \Theta(n/\log n)$, then the data-set matrix $\mathbf{X}$ can be extracted exactly (no error).

Proof: The proof is based on induction. First, for $h = 1$, by the result in Section IV-A, the claim is true.

Suppose for all $h \le \eta \in \mathbb{N}$ the claim is true. Now, for $h = \eta + 1$, we make a first pass to reduce the original data extraction problem to some sub-problems with a alphabet of smaller size ($h = \eta$). The procedure is described as follows:

1) Pick $t \in \mathbb{N}$ such that $n_t < n \le n_{t+1}$, $n_t = (t+3)3^{t-1}$.
2) Append zeros to $\mathbf{X}$ to generate $\overline{\mathbf{X}} \in \mathcal{A}^{n_t \times 1}$.
3) Calculate $\overline{\mathbf{Y}} \triangleq \mathbf{Q}^{(m_{t+1}, n_{t+1})}\overline{\mathbf{X}}$.

We then apply the row operation in (3) on $\overline{\mathbf{Y}}$ to get

$$\overline{\mathbf{Y}}' = \begin{bmatrix}\mathbf{0}_{m_t \times n_t} & 2\mathbf{Q}^{(m_t, n_t)} & \mathbf{0}_{m_t \times n_t} & \mathbf{I}_{m_t}\end{bmatrix}\overline{\mathbf{X}}.$$

From $\overline{\mathbf{Y}}'$, we are able to learn the *parity* of the last m_t entries of $\overline{\mathbf{X}}$, that is, $\overline{\mathbf{X}}_{[(3n_t+1)\downarrow(3n_t+m_t)]}$. Hence after the first pass, the range of each element in the column vector $\overline{\mathbf{X}}_{[(3n_t+1)\downarrow(3n_t+m_t)]}$ is reduced from a size-$2^{\eta+1}$ arithmetic progression to a size-2^η arithmetic progression.

The second pass is to extract $\overline{\mathbf{X}}_{[(3n_t+1)\downarrow(3n_t+m_t)]}$ exactly by using the method for a size-2^η arithmetic progression. By the induction hypothesis, this can be done with $\Theta(m_t/\log m_t)$ queries. After the second pass, we can obtain $\mathbf{Q}^{(m_t,n_t)}\overline{\mathbf{X}}_{[1\downarrow n_t]}$, $\mathbf{Q}^{(m_t,n_t)}\overline{\mathbf{X}}_{[(n_t+1)\downarrow 2n_t]}$, and $\mathbf{Q}^{(m_t,n_t)}\overline{\mathbf{X}}_{[(2n_t+1)\downarrow 3n_t]}$. Again, apply the row operation in (3) on them, and similar to the above discussion, we can then figure out the parity of the last m_{t-1} entries of $\overline{\mathbf{X}}_{[1\downarrow n_t]}$, $\overline{\mathbf{X}}_{[(n_t+1)\downarrow 2n_t]}$, and $\overline{\mathbf{X}}_{[(2n_t+1)\downarrow 3n_t]}$ respectively.

Then, again by the induction hypothesis, we can extract these $3m_{t-1} = m_t = 3^t$ entries with $\Theta(m_t/\log m_t)$ queries. Continuing with the procedure repeatedly t times, we can extract the entire $\overline{\mathbf{X}}$ with

$$m = m_{t+1} + t\Theta(m_t/\log m_t) = 3^{t+1} + t\Theta\left(3^t/t\right) = \Theta\left(3^t\right)$$

queries. Note that $n = \Theta(t3^t)$, and hence similar to the arguments in Section IV-A, $m = \Theta(n/\log n)$ for $h = \eta + 1$.

The proof is complete by induction. ∎

V. Random Sampling

Let us first provide a simple upper bound on the probability of error when the query matrix $\mathbf{Q}$ is generated according to

$$(\mathbf{Q})_{i,j} \overset{\text{i.i.d.}}{\sim} \text{Ber}(q), \ \forall i \in [m], \ \forall j \in [n].$$

With a slight abuse of notation, let $\mathbf{x}$ denote the *first column* of $\mathbf{X}$. Then, $\mathsf{P_e}(\mathbf{X}; \mathbf{Q})$ is upper bounded by

$$\begin{aligned}&\mathbb{P}\left\{\exists\, \tilde{\mathbf{x}} \in \{0,1\}^{n\times 1}, \tilde{\mathbf{x}} \neq \mathbf{x} \text{ such that } \mathbf{Q}\tilde{\mathbf{x}} = \mathbf{Q}\mathbf{x}\right\}\\ &= \mathbb{P}\left\{\bigcup_{\tilde{\mathbf{x}}\neq\mathbf{x}} \{\mathbf{Q}\tilde{\mathbf{x}} = \mathbf{Q}\mathbf{x}\}\right\} \le \sum_{\tilde{\mathbf{x}}\neq\mathbf{x}} \mathbb{P}\{\mathbf{Q}\tilde{\mathbf{x}} = \mathbf{Q}\mathbf{x}\} \quad (4)\\ &= \sum_{\tilde{\mathbf{x}}\neq\mathbf{x}} \left(\mathbb{P}\{\mathbf{q}\tilde{\mathbf{x}} = \mathbf{q}\mathbf{x}\}\right)^m. \quad (5)\end{aligned}$$

Here (4) is due to union bound, and (5) is due to the fact the rows of $\mathbf{Q}$, $\{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$, are i.i.d. distributed as $\mathbf{q}$.

To get a handle on (5), we introduce couples of notations below. Let k_1 and k_2 denote the number of 1's in $\mathbf{x}$ (the first column of $\mathbf{X}$) and $\tilde{\mathbf{x}}$ respectively. Furthermore, let $\mathcal{T}_1 \triangleq \left\{j \,\middle|\, (\tilde{\mathbf{x}})_j = 0, (\mathbf{x})_j = 1\right\}$, and $\mathcal{T}_2 \triangleq \left\{j \,\middle|\, (\tilde{\mathbf{x}})_j = 1, (\mathbf{x})_j = 0\right\}$. Let $t_1 \triangleq |\mathcal{T}_1|, t_2 \triangleq |\mathcal{T}_2|$. Note that $t_1 - t_2 = k_1 - k_2$.

Let the queried subset correspond to the row vector $\mathbf{q}$ be $\mathcal{S}$. Note that the *confusion event* $\{\mathbf{q}\tilde{\mathbf{x}} = \mathbf{q}\mathbf{x}\}$ happens if and only if $|\mathcal{S}\cap\mathcal{T}_1| = |\mathcal{S}\cap\mathcal{T}_2|$, that is, the # of items sampled in $\mathcal{T}_1$ is equal to the # of items sampled in $\mathcal{T}_2$. Hence, denoting this number by s, we can write down $\mathbb{P}\{\mathbf{q}\tilde{\mathbf{x}} = \mathbf{q}\mathbf{x}\}$ explicitly:

$$\begin{aligned}&\sum_{s=0}^{t} \tbinom{t_1}{s}\tbinom{t_2}{s} q^{2s}(1-q)^{t_1+t_2-2s} \qquad (t \triangleq \min(t_1,t_2))\\ &= \sum_{s=0}^{t} \tbinom{t}{s}\tbinom{t+\ell}{s} q^{2s}(1-q)^{\ell+2t-2s}. \qquad (\ell \triangleq |k_1-k_2|)\end{aligned}$$

Since this expression only depends on (k_1, k_2, t, q), we denote it by $\mathsf{p}_{(k_1,k_2,t,q)} \triangleq \mathbb{P}\{\mathbf{q}\tilde{\mathbf{x}} = \mathbf{q}\mathbf{x}\}$. Let $k \triangleq \min(k_1, k_2)$ and note that $t_1 = k_1 - k + t$, $t_2 = k_2 - k + t$. Therefore,

$$\begin{aligned}(5) = &\sum_{k_2\neq k_1, k_2=0}^{n} \tbinom{k_1}{k}\tbinom{n-k_1}{k_2-k}\left(\mathsf{p}_{(k_1,k_2,0,q)}\right)^m\\ &+ \sum_{k_2=0}^{n}\sum_{t=1}^{\bar{t}} \tbinom{k_1}{k-t}\tbinom{n-k_1}{k_2-k+t}\left(\mathsf{p}_{(k_1,k_2,t,q)}\right)^m,\end{aligned}$$

where $\bar{t} \triangleq \min\{k_1, k_2, n-k_1, n-k_2\}$ is the maximum value that $t = \min(t_1, t_2)$ can take.

Let us denote the above upper bound (5) by $\overline{\mathsf{P}}_{\mathsf{e}}^{(n,m,q)}(\mathbf{X})$ to stress its dependency on (n, m, q). We focus on the case $q = 1/2$, where $\mathsf{p}_{(k_1,k_2,t,q)}$ is greatly simplified:

$$\mathsf{p}_{(k_1,k_2,t,1/2)} = \left(\tfrac{1}{2}\right)^{\ell+2t}\sum_{s=0}^{t}\tbinom{t}{s}\tbinom{t+\ell}{s} = \left(\tfrac{1}{2}\right)^{\ell+2t}\tbinom{\ell+2t}{t}.$$

Lemma 5.1: For $t \ge 1$, $\mathsf{p}_{(k_1,k_2,t,1/2)} \le \overline{\mathsf{p}}_{(k_1,k_2,t,1/2)}$, where

$$\overline{\mathsf{p}}_{(k_1,k_2,t,1/2)} \triangleq \begin{cases}\left(\sqrt{\pi t}\right)^{-1} & \ell \le 1\\ \left(\max\left\{1, 2^\ell \ell^{-t}\right\}\sqrt{\pi t}\right)^{-1} & \text{otherwise}\end{cases}$$

For $t=0$, $\mathsf{p}_{(k_1,k_2,0,1/2)}=2^{-\ell}$.

With Lemma 5.1, we now have

$$\overline{\mathsf{P}}_{\mathsf{e}}^{(n,m,1/2)}(\mathbf{X}) \le \sum_{k_2\neq k_1,k_2=0}^{n} \binom{k_1}{k}\binom{n-k_1}{k_2-k}\left(\tfrac{1}{2}\right)^{\ell m} \quad (6)$$

$$+\sum_{k_2=0}^{n}\sum_{t=1}^{\bar{t}} \binom{k_1}{k-t}\binom{n-k_1}{k_2-k+t}\left(\overline{\mathsf{p}}_{(k_1,k_2,t,1/2)}\right)^{m} \quad (7)$$

The following lemma gives a bound on (7).

Lemma 5.2: When n is large enough, $\exists\, c_1, c_2 > 0$ such that,

$$\sum_{t=1}^{\bar{t}} \binom{k_1}{k-t}\binom{n-k_1}{k_2-k+t}\left(\overline{\mathsf{p}}_{(k_1,k_2,t,1/2)}\right)^{m} \le e^{(-c_1 m\log n + c_2 n)}.$$

Proofs of Lemma 5.1 and 5.2 are in Appendix of [1].

To complete the proof of Theorem 3.2, by Lemma 5.2,

$$(7) \le n e^{(-\Theta(\log n)m+\Theta(n))} = e^{(-\Theta(\log n)m+\Theta(n))} \to 0$$

as $n\to\infty$ if $m=\Theta\left(\frac{n}{\log n}\right)$. For the other part (6),

$$\begin{aligned}(6) &= \sum_{k_2=0}^{k_1-1}\binom{k_1}{k_2}2^{-(k_1-k_2)m} + \sum_{k_2=k_1+1}^{n}\binom{n-k_1}{k_2-k_1}2^{-(k_2-k_1)m}\\ &= \left(1+2^{-m}\right)^{k_1} - 1 + \left(1+2^{-m}\right)^{n-k_1} - 1\\ &\le 2\left\{\left(1+2^{-m}\right)^{n}-1\right\}.\end{aligned}$$

If $m=\Theta\left(\frac{n}{\log n}\right)$, $\lim_{n\to\infty}\left(1+2^{-m}\right)^n \le \lim_{n\to\infty}\left(1+2^{-\sqrt{n}}\right)^n$. Since $\lim_{n\to\infty} n\log\left(1+2^{-\sqrt{n}}\right)=0$ (simple calculation; details in Appendix of [1]), we have $\lim_{n\to\infty}\left(1+2^{-\sqrt{n}}\right)^n = 1$, and hence (6) $\to 0$ as $n\to\infty$. Proof complete.

VI. Discussions and Related Works

In this work, we establish the fundamental limit on the number of queries required to achieve lossless data extraction, in both the histogram-query setting and the arithmetic-mean-query setting. It turns out that the number of queries required is *sub-linear* in the size of the data set, when the data curator responds honestly and the alphabet size is fixed with respect to the data set size. When the alphabet size d grows up with the data set size n, both the converse and the achievability part shall be improved. This direction is left as future work.

Our result has applications beyond the analyst-curator framework described above. To name a few:

- *Crowd sensing*: In this problem, one would like to learn the opinions of a crowd by polling multiple subsets. The responses are recorded anonymously, and hence only the histogram of the opinions is collected.
- *Sensor network*: Consider a sensor network with a centralized server collecting data from multiple data fusion centers. Each fusion center, in charge of a subset of sensors, only reports the histogram or the arithmetic mean of sensors' data, due to various concerns such as privacy.

In these application scenarios, it is natural to address further constraints on the query matrix. In particular, the number of people/sensors in each poll/measurement may be constrained from above or below, corresponding to constraint on the number of 1's in each row of the query matrix. The number of times that one person/sensor is polled/measured can also be limited, corresponding to constraint on the number of 1's in each column. It turns out not difficult to address these *sparsity* constraints on the query matrix, and it is an onoging work.

Related Works

Our problem can be viewed as generalization of *group testing* [2] if the alphabet is binary and the data set is assumed to be *sparse*. In group testing, the response is the "OR" of the sampled bits, while in our setting, the response is the "SUM" (in $\mathbb{R}$) of the sampled bits. Fixing the number of 1's, a recent line of works have taken an information theoretic approach towards group testing problems [3]–[7]. The information theoretic and algorithmic investigation of the sparse recovery problem via histogram queries is an ongoing work, where we not only extend the framework in [3]–[7] to investigate the fundamental limit of random sampling but also propose an adaptive algorithm to achieve exact extraction.

In [8] the sum query model is also investigated for the binary alphabet, where the data curator perturbs the response to ensure privacy. A $\mathsf{Poly}(n)$ attack algorithm is proposed, which reconstructs the size-n data set as long as the perturbation is $o(\sqrt{n})$. Their work is different from ours in two aspects. First, [8] is mainly interested in whether the computational complexity of data extraction is polynomial or not, while our work is focused on the fundamental limit of the number of queries. Second, [8] allows perturbation in the response, while in our work the data curator is honest. A future direction is to investigate the fundamental trade-off between the amount of perturbation and the number of queries required.

Acknowledgment

The work of I-Hsiang Wang was supported by National Taiwan University under Grants NTU-ERP-104R89084E and NTU-ERP-105R89084.

References

[1] I.-H. Wang, S.-L. Huang, K.-Y. Lee, and K.-C. Chen, "Data extraction via histogram and arithmetic mean queries: Fundamental limits and algorithms," *Manuscript*, 2016. Available at http://homepage.ntu.edu.tw/%7Eihwang/Research/Eprints/ISIT16DE.pdf

[2] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[3] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *Information Theory, IEEE Transactions on*, vol. 58, no. 3, pp. 1880–1901, 2012.

[4] M. Aldridge, L. Baldassini, and O. Johnson, "Group testing algorithms: bounds and simulations," *Information Theory, IEEE Transactions on*, vol. 60, no. 6, pp. 3671–3687, 2014.

[5] V. Y. Tan and G. K. Atia, "Strong impossibility results for sparse signal processing," *Signal Processing Letters, IEEE*, vol. 21, no. 3, pp. 260–264, 2014.

[6] T. Laarhoven, "Asymptotics of fingerprinting and group testing: Tight bounds from channel capacities," *Information Forensics and Security, IEEE Transactions on*, vol. 10, no. 9, pp. 1967–1980, 2015.

[7] J. Scarlett and V. Cevher, "Phase transitions in group testing," in *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2016.

[8] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2003, pp. 202–210.