Group Testing

Amit Kumar Sinhababu* and Vikraman Choudhury†

Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur

April 21, 2013

1 Motivation

Out original motivation in this project was to study "coding theory in data streaming", which has two aspects.

- Applications of theory correcting codes to efficiently solve problems in the model of data streaming.
- Solving coding theory problems in the model of data streaming. For example, "Can one recognize a Reed-Solomon codeword in one-pass using only poly-log space?" [1]

As we started, we were directed to a related combinatorial problem, "Group testing", which is important on its own, having connections with "Compressed Sensing", "Data Streaming", "Coding Theory", "Expanders", "Derandomization".

This project report surveys some of these interesting connections.

2 Group Testing

The group testing problem is to identify the set of "positives" ("defectives", or "infected", or 1) from a large set of population/items, using as few tests as possible.

*amitks@cse.iitk.ac.in
†vikraman@cse.iitk.ac.in

2.1 Definition

There is an unknown stream $x \in \{0,1\}^n$ with at most d ones in it. We are allowed to test any subset S of the indices. The answer to the test tells whether $x_i = 0$ for all $i \in S$, or not (at least one $x_i = 1$). The objective is to design as few tests as possible (t tests) such that x can be identified as fast as possible.

Group testing strategies can be either adaptive or non-adaptive. A group testing algorithm is non-adaptive if all tests must be specified without knowing the outcome of other tests. Schemes in which the tests for the next stage depend on the results of the previous stage are known as adaptive procedures.

In this report, "positive" items are same as "defective" or "infected" or 1. On the other hand, "negative" items are same as "good" or 0.

$t(d,n)$ denotes the minimum number of non-adaptive tests that one would have to make to detect all "positive" items given a set of n items with at most d "positives".

$t^a(d,n)$ denotes the minimum number of adaptive tests that one would have to make to detect all "positive" items given a set of n items with at most d "positives".

Clearly, as any non-adaptive test can be converted to a 1-stage adaptive test, $1 \leq t^a(d,n) \leq t(d,n) \leq n$.

Lower bound on the number of adaptive tests is given by $t^a(d,n) \geq d \log n/d$.

Upper bound on $t^a(d,n)$ is given by $t^a(d,n) \leq O(d \log n)$. This can be achieved using binary search at most d times.

A non-adaptive group testing algorithm can be representated as $t \times N$ matrix M, where each row corresponds to the characteristic vector of the subset of $[N]$ to be tested. The answers to the test can be interpreted as Mx where the addition is logical OR and multiplication is logical AND.

Sufficient conditions for matrices representing uniquely decodable non-adaptive group testing algorithms are discussed as follows.

2.2 d-separable matrices

For the non-adaptive group testing strategy to be able to uniquely identify the set of at most d positives, the measurement matrix M must satisfy certain properties. Let the test outcome vector be $y = (y_i)_{i=1}^t$, where $y_i = 1$ if and only if the i^{th} test is positive. To be able to identify an arbitrary subset $D \subseteq [N]$ of at most d positives, the test outcome vectors y have to be distinct. Hence, we can derive the notion of d-separable matrix.

Definition 1. *For positive integers t, d, N, where $d \leq N$, a $t \times N$ binary matrix M is d-separable[2] if the unions of upto d columns of M are all distinct.*

If we use a d-separable matrix M, then brute-force decoding will have to check every subset ($\subseteq N$) of cardinality $\leq d$. This gives it a time complexity of $O(n^d)$. If we use lookup tables, then we need the same amount of space. Clearly, this time/space complexity is too high. So, we need a different sufficient condition which will enable more efficient decoding.

2.3 d-disjunct matrices

Definition 2. *For positive integers t, d, N, where $d \leq N$, a $t \times N$ binary matrix M is d-disjunct[2] if the union of arbitrary $\leq d$ columns does not contain another column.*

2.3.1 Naive decoding

for $j = 1$ *to* N **do**
  if *item j belongs to at least one negative test* **then**
    mark j as a negative item
  end
end
return R, the set of remaining items

Correctness:
Let $P(S) = \cup_{j \in S} M_j$.
If j is a negative item, then by definition of d-disjunctness, $P(S)$ does not contain j, i.e., j is in some negative test. On the contrary, if j is a positive item, then it is in no negative test as evident from the definition of test.
Time complexity is $O(tn)$.
Example of d-disjunct matrix:
When $d = 1$, the bit test matrix B is 1-disjunct. The i^{th} column of B, $i \in 0..\log N - 1$ is the binary representation of i. In this case, $t = \log N$ and this matches the lower bound for the minimum number of tests to locate a single positive item from N. For $N = 8$, the bit test matrix looks as follows:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

We consider the following example:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Here, bit 1 was in the second position and the resultant vector is binary representation of 2. In general, the bit test matrix times the unknown vector gives the location of the positive item in binary. So, decoding is easier and faster in this specific case of $d = 1$.

2.3.2 Construction of d-disjunct matrices

It is known that $t \times n$ d-disjunct matrices can be constructed for $t = O(d^2 \log n)$. In fact, a lower bound $t = \Omega(d^2 / \log d \log n)$ is also known.

It can be proved that a randomly chosen matrix is d-disjunct with $t = O(d^2 \log n)$ rows with high probability. But this fact does not tell us how to obtain such a matrix explicitly. By derandomization of probabilistic proof of Gilbert-Varshamov bound [3], one can construct a d-disjunct matrix in time $\tilde{O}(d^2 \log n)$. Using concatenated codes, one can have strongly explicit construction of d-disjunct matrix with $O(d^2 \log^2 n)$ rows, i.e. this bound is worse by a $\log n$ factor.

2.4 (d, l)-list separable and disjunct matrices

It's difficult to improve upon the decoding time if we only consider properties of generic disjunct matrices. So, a possible direction to improve decoding time is to impose some more structure into disjunct matrices.

The key idea [3] is to first use a "filtering" matrix F. This matrix "quickly" (in time $poly(d, \log n)$) filters a "small set" (containing $poly(d, \log n)$ items), including all the positives and few negatives. Then, another matrix D, also called "identification" matrix exactly detects the positives from the filtered set L using naive decoding.

This idea (motivated from list recoverable codes) is beautiful, and can possibly be applied to many other problems. We thought of a small extension from 2-stage filtering-identification to multi-stage recursive filtering-identification. Similar idea has been used in designing efficiently decodable compressed sensing schemes [4].

Definition 3. *For positive integers t, d, l, N, where $d + l \leq N$, a $t \times N$ binary matrix M is said to be (d, l)-**list-separable** if it satisfies the following property. For any $y \in 0, 1^t$, there exists a column set R_y such that, if T is any set of at most d columns of M whose union is y, then $T \subseteq R_y$ and $|R_y| < |T| + l$.*

Definition 4. *For positive integers t, d, l, N, where $d + l \leq N$, a $t \times N$ binary matrix M is said to be (d, l)-**list-disjunct** if and only if, for any two disjoint sets S and T of columns of M with $|S| = l$ and $|T| = d$, there exists a row of M in which some column in S has a 1 but all columns in T have 0s.*

2.5 Applications

Historically, group testing was first used during World War II for testing Syphilis among soldiers in United States. Here, items are blood samples which are "positive" if they're infected. A test is a group of blood samples. Testing a group of samples at a time saves many tests if the test outcome is "negative". If the test outcome is "positive", then we come to know, that at least in the group is positive, but not which one. Afterwards, group testing has found various applications. Many applications are covered in the book. [5]

Some other applications are:

- Computational Biology [6]
- DNA Library screening [6]
- Multiple access control protocols [7]
- Data pattern mining [8]
- Data Forensics [9]
- Function learning [10]
- Traitor tracing problem [2]

Some more applications are mentioned in Cormode [11].

3 Connections to other problems

3.1 Heavy Hitter

Definition 5. *Given a sequence of m items from $[n]$, an item is considered "hot" or "heavy" if it occurs $> m/(d+1)$ times.*

From the definition, there can be at most d "heavy-hitters". In Cormode and Muthukrishnan [12], non-adaptive group testing is used to solve the problem in data-streaming model assuming the following "small-tail" property: all of the non-hot items occur at most $m/(d+1)$ times.

create a counter c
for *each test* $i \in [t]$ **do**
  create a counter c_i
end

for *each arriving (leaving) item* $j \in [n]$ **do**
  increment (decrement) all the counters c_i such that $M_{ij} = 1$
  (i.e. all counters c_i for which test i contains item j)
  increment counter c
end

To compute the "heavy-hitters", we consider the binary vector x such that $x_i = 1$ if and only if $c_i > m/(d+1)$. Since a test's outcome is positive if and only if it contains a hot item, decoding the vector x corresponds to solving the problem.

Group testing algorithms motivated by this data streaming application must satisfy the following requirements: [2]

- *Small number of tests.* Since number of tests t is proportional to the memory space needed by the streaming algorithm, we should minimize it.

- *Strongly explicit construction.* We should be able to compute a column of M quickly, i.e., given two indices $(i, j) \in [t] \times [N]$, m_{ij} can be computed in time $poly(t, \log N)$.
- *Sub-linear decoding time.* We want an ideal disjunct or separable matrix which can be decoded in sublinear time in N, preferably poly-log in N, compared to $O(tN)$ time for the naive decoder.
- *Error-tolerance.* The "small-tail" property does not always hold. Since the test outcome can contain false-positives, we want error-tolerant group testing strategies.

During the publication of Cormode and Muthukrishnan [12], the problem of sublinear decoding d-disjunct matrices was still open. So, the authors provided alternate algorithms for "heavy-hitters" inspired by the group testing idea, leaving open the problem of designing efficiently decodable group testing matrix. In Indyk, Ngo, and Rudra [13], this question was resolved. But the solution of the problem using their results is not as good as the best known results for the "heavy-hitters" problem. Still, Indyk, Ngo, and Rudra [13] mention that "heavy-hitters" is illustrative of many other applications of non-adaptive group testing to data streaming algorithms. New applications are possible along these lines.

3.2 Compressed Sensing

Compressed Sensing and non-adaptive combinatorial group testing are closely related. Their main goal is to recover a sparse vector x from an underdetermined set of linear equations $Mx = y$ given M and y. But they use different models of arithmetic.

Compressed Sensing is performed over complex/real arithmetic whereas group testing is performed over boolean arithmetic.

d-separable matrix M can be used to solve compressed sensing problems [14]. These ideas are also extended to fourier sampling case [15].

3.3 Expanders in Compressed Sensing

There are two main algorithmic approaches to sparse signal recovery or compressed sensing. One is utilising geometric properties (example, restricted isometry property, i.e. mapping M preserves the equilibrium norm of sparse signals) of measurement matrix M. Random dense matrices (gaussian, fourier) satisfy this property with high probability. In geometric method, the recovery methods are geometric, for example, l_1 minimisation, LP-decoding [16].

The second approach is combinatorial approach, utilising sparse matrices interpreted as adjacency matrices of sparse graphs. Combinatorial techniques are used for recovery. In Berinde, Gilbert, Indyk, Karloff, and Strauss [16], expanders are used in compressed sensing. Jafarpour, Xu, Hassibi, and Calderbank [17] also used expanders for efficiently decodable compressed sensing schemes.

3.4 Bipartite Expander Graphs

Definition 6. *A W-left regular bipartite graph $[N]x[W] \to [T]$ is an $(N, W, T, D, (1-\epsilon)W)$ expander if every subset $S \subset [N]$ of size at most D has a neighborhood (denoted by $\Gamma(S)$) of size at least $(1-\epsilon)|S|W$.*

Proposition 1. *Let G be a $(n, w, t, 2, w(1-\epsilon))$-expander. Then M_G is a d-disjunct matrix.[13]*

Proof. Let us consider two different vertices $i, j \in [N], i \neq j$. Now, $|\Gamma(i)| = w = |\Gamma(j)|$, and $|\Gamma(i,j)| \geq 2w(1-\epsilon))$. This implies that, $|\Gamma(i) \cap \Gamma(j)| \leq 2\epsilon w$. Now, if M_g has to be a d-disjunct matrix, then $(2\epsilon w)d + 1 \leq w$, or, $\epsilon \leq \frac{1}{d+1}$. □

Proposition 2. *Let G be a $(n, w, t, 2d, w/2+1)$-expander. Then M_G is a (d, d)-list disjunct matrix.[13]*

Proof. Let us consider two disjoint subsets of columns of size exactly d, S_1 and S_2. Since M_G is (d, d)-list disjunct, $\cup_{i \in S_1} M_i \not\subseteq \cup_{j \in S_2} M_j$ and $\cup_{j \in S_2} M_j \not\subseteq \cup_{i \in S_1} M_i$. Hence, $\Gamma(S_1) \not\subseteq \Gamma(S_2)$ and $\Gamma(S_2) \not\subseteq \Gamma(S_1)$. This is true if $|\Gamma(S_1 \cup S_2)| > wd$ since $|\Gamma(S_1)|, |\Gamma(S_2)| \leq wd$. Taking $|\Gamma(S_1 \cup S_2) \geq wd + 2d|$, expansion factor of $G = w/2 + 1$. □

3.5 Bipartite Disperser Graphs

Definition 7. *A D-left regular bipartite graph $[N]x[D] \to [T]$ is an (N, L, T, D, ϵ) disperser if every subset $S \subset [N]$ of size at least L has a neighborhood of size at least $\epsilon|T|$.*

Proposition 3. *Let G be a $(n, l, t, \epsilon t/(d+1), \epsilon)$-disperser. Then M_G is a (d, l)-list disjunct matrix.[13]*

Proof. Let us consider two disjoint subsets of columns, S_1 and S_2, with $|S_1| \geq l$ and $|S_2| = d$. Now, $|\cup_{i \in S_2} M_i| \leq \frac{dt\epsilon}{d+1} \leq \epsilon t$. Also, $|\cup_{i \in S_1} M_i| \geq \epsilon t$, which along with the above inequality implies that, $\cup_{i \in S_1} M_i \not\subseteq \cup_{i \in S_2} M_i$. □

3.6 Sparsity Separator

Given a stream of m items from the domain $[n]$, let f be the vector of non-negative frequencies, i.e., for every $i \in [n]$, f_i denotes the number of occurences of i in the stream.

Definition 8. *A (d, l)-sparsity separator[18] is a data structure which can determine if the frequency vector corresponding to a stream has at most d non-zero entries or it has at least l non-zero entries.*

Proposition 4. *Any $(d, l-d)$-list disjunct $t \times n$ matrix M can be used to build a (d, l)-sparsity separator structure.[13]*

Algorithm sketch:

We maintain t counters, one for each test, $c_{j\,j\in[t]}$. Whenever an item i arrives/leaves, we increment/decrement all counters c_j such that the j^{th} test contains i. At the end, we convert the counters to a result vector $r \in 0, 1^t$, such that $r_j = 1$ if and only if $c_j > 0$. We decode the result vector r according to M and count the number of ones. If the number is at most $l-1$ then we declare the sparsity to be at most d, otherwise we declare the sparsity to be at least l.

Proof of correctness:

We first define a binary vector $x \in 0, 1^n$ such that $x_i = 1$ if and only if $f_i > 0$. Then, the result vector r is exactly the same as Mx. Now, we consider the following cases:

- x has Hamming weight at most d. Since, M is $(d, l-d)$-list disjunct, the decoder for M will output a vector with at most $d + (l-d) - 1 = l - 1$ ones in it.
- x has Hamming weight at least l. Each $x_i = 1$ will contribute a one to all the r_j such that i is contained in test j. Thus, the decoder for M will output a vector with at least l ones in it.

4 Open Problems

From personal communication with Atri Rudra, we came to know about the following open problems.

4.1

Find a (strongly or not) explicit construction of (d, d)-list-disjunct matrices attaining the probabilistic bound $O(d \log n/d)$.

4.2

Closing the gap between upper and lower bound of number of tests for d-disjunct matrices.

4.3

Find a (strongly or not) explicit construction of $(d, poly(d)$-list-disjunct matrices attaining the probabilistic bound $O(d \log n/d)$ and are efficiently decodable.

4.4

Find a sufficient condition to determine whether a matrix is $(d, O(d))$-list disjunct in time $n^{O(d)}$.

5 Conclusion

We have seen the connections between group testing, compressed sensing, expander, streaming algorithms. Possibly these connections can be used to solve new problems. Deeper exploration of coding theory, expander, compressed sensing, etc. can give new insights in data streaming algorithms and more generally in complexity theory. From this project, we also came to know about the usefulness of expanders. In future, we want to see if we can use these ideas to the problem of Sparse Fourier Transform.

References

[1] Atri Rudra. Problem 57: Coding theory in the streaming model, . URL `http://sublinear.info/57`.

[2] Atri Rudra. Cse 709: Compressed sensing and group testing, part i (fall 2011 seminar), . URL `http://www.cse.buffalo.edu/~hungngo/classes/2011/709/schedule.html`.

[3] Atri Rudra Hung Q. Ngo. Efficient decoding and list group testing. URL `http://www.cse.buffalo.edu/~hungngo/classes/2011/709/lectures/6.pdf`.

[4] Hung Q. Ngo, Ely Porat, and Atri Rudra. Efficiently decodable compressed sensing by list-recoverable codes and recursion. In *STACS*, pages 230–241, 2012.

[5] Ding Zhu Du and Frank Hwang. *Combinatorial group testing and its applications*. World Scientific Publishing Company, 1993.

[6] Hung Q Ngo and Ding-Zhu Du. A survey on combinatorial group testing algorithms with applications to dna library screening. *Discrete mathematical problems with medical applications*, 55:171–182, 2000.

[7] Toby Berger, Nader Mehravari, Don Towsley, and Jack Wolf. Random multiple-access communication and group testing. *Communications, IEEE Transactions on*, 32(7):769–779, 1984.

[8] Anthony J Macula and Leonard J Popyack. A group testing method for finding patterns in data. *Discrete applied mathematics*, 144(1):149–157, 2004.

[9] Michael T Goodrich, Mikhail J Atallah, and Roberto Tamassia. Indexing information for data forensics. In *Applied Cryptography and Network Security*, pages 206–221. Springer, 2005.

[10] Anna C Gilbert, Mark A Iwen, and Martin J Strauss. Group testing and sparse signal recovery. In *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, pages 1059–1063. IEEE, 2008.

[11] Graham Cormode. Distributed streams. URL `www.cse.iitk.ac.in/users/sganguly/slides/cormode.ppt`.

[12] Graham Cormode and S Muthukrishnan. What's hot and what's not: tracking most frequent items dynamically. *ACM Transactions on Database Systems (TODS)*, 30(1):249–278, 2005.

[13] Piotr Indyk, Hung Q Ngo, and Atri Rudra. Efficiently decodable non-adaptive group testing. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1126–1142. Society for Industrial and Applied Mathematics, 2010.

[14] Graham Cormode and S Muthukrishnan. Combinatorial algorithms for compressed sensing. In *Structural Information and Communication Complexity*, pages 280–294. Springer, 2006.

[15] Mark A Iwen. A deterministic sub-linear time sparse fourier algorithm via non-adaptive compressed sensing methods. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 20–29. Society for Industrial and Applied Mathematics, 2008.

[16] Radu Berinde, Anna C Gilbert, Piotr Indyk, H Karloff, and Martin J Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 798–805. IEEE, 2008.

[17] Sina Jafarpour, Weiyu Xu, Babak Hassibi, and Robert Calderbank. Efficient and robust compressed sensing using optimized expander graphs. *Information Theory, IEEE Transactions on*, 55(9):4299–4308, 2009.

[18] Sumit Ganguly. Data stream algorithms via expander graphs. In *Algorithms and Computation*, pages 52–63. Springer, 2008.