Title: Efficiently Decodable Group Testing
Name: Hung Q. Ngo, Atri Rudra
Affil./Addr. Department of Computer Science and Engineering, University at Buffalo, the State University of New York, U.S.A.
Keywords: Non-adaptive group testing, sublinear-time decoding, coding theory
SumOriWork: 2011; Ngo, Porat, Rudra

Efficiently Decodable Group Testing

Hung Q. Ngo, Atri Rudra

Department of Computer Science and Engineering, University at Buffalo, the State University of New York, U.S.A.

Years aud Authors of Summarized Original Work

2011; Ngo, Porat, Rudra

Keywords

Non-adaptive group testing, sublinear-time decoding, coding theory

Problem Definition

The basic group testing problem is to identify the unknown set of "*positive items*" from a large population of "*items*" using as few "*tests*" as possible. A test is a subset of items. A test returns positive if there is a positive item in the subset. The semantics of "positives," "items," and "tests" depend on the application.

In the original context [3], group testing was invented to solve the problem of identifying syphilis infected blood samples from a large collection of WWII draftees' blood samples. In this case, items are blood samples, which are positive if they are infected. A test is a *pool* (group) of blood samples. Testing a group of samples at a time will save resources if the test outcome is negative. On the other hand, if the test outcome is positive then all we know is that at least one sample in the pool is positive but we do not know which one(s).

In *non-adaptive combinatorial group testing* (NACGT), we assume that the number of positives is at most d for some fixed integer d, and that all tests have to be specified in advance before any test outcome is known. The NACGT paradigm has found numerous applications in many areas of Mathematics, Computer Science, and Computational Biology [4; 9; 10].

A NACGT strategy with t tests on a universe of N items is represented by a $t \times N$ binary matrix $\mathbf{M} = (m_{ij})$, where $m_{ij} = 1$ iff item j belongs to test i. Let $\mathbf{M}_i$ and $\mathbf{M}^j$ denote row i and column j of $\mathbf{M}$, respectively. Abusing notation, we will also use $\mathbf{M}_i$ (respectively, $\mathbf{M}^j$) to denote the set of rows (respectively, columns) corresponding to the 1-entries of row i (respectively, column j). In other words, $\mathbf{M}_i$ is the ith pool, and $\mathbf{M}^j$ is the set of pools that item j belongs to.

Let $D \subset [N]$ be the unknown subset of positive items, where $|D| \leq d$. Let $\mathbf{y} = (y_i)_{i=1}^t \in \{0,1\}^t$ denote the test outcome vector, i.e. $y_i = 1$ iff the ith test is positive. Then, the test outcome vector is precisely the (boolean) union of the positive columns: $\mathbf{y} = \bigcup_{j \in D} \mathbf{M}^j$. The task of identifying the unknown subset D from the test outcome vector $\mathbf{y}$ is called *decoding*.

The main problem In many modern applications of NACGT, there are two key requirements for a NACGT scheme:

(1) *Small number of tests.* “Tests” are computationally expensive in many applications.

(2) *Efficient decoding.* As the item universe size N can be extremely large, it would be ideal for the decoding algorithm to run in time sub-linear in N, and more precisely in $\mathrm{poly}(d, \log N)$ time.

Key Results

To be able to uniquely identify an arbitrary subset D of at most d positives, it is necessary and sufficient for the test outcome vectors $\mathbf{y}$ to be different for distinct subsets D of at most d positives. A NACGT matrix with the above property is called *d-separable*. However, in general such matrices only admit the brute force $\Omega(N^d)$-time decoding algorithm. A very natural decoding algorithm called the *naïve decoding algorithm* runs much faster, in time $O(tN)$.

Definition 1 (Naïve decoding algorithm). *Eliminate all items that participate in negative tests, return the remaining items.*

This algorithm does not work for arbitrary d-separable matrices. However, if the test matrix $\mathbf{M}$ satisfies a slightly stronger property called *d-disjunct*, then the naïve decoding algorithm is guaranteed to work correctly.

Definition 2 (Disjunct matrix). *A $t \times N$ binary matrix $\mathbf{M}$ is said to be d-disjunct iff $\mathbf{M}^j \setminus \bigcup_{k \in S} \mathbf{M}^k \neq \emptyset$ for any set S of d columns and any $j \notin S$. (See Fig. 1.)*

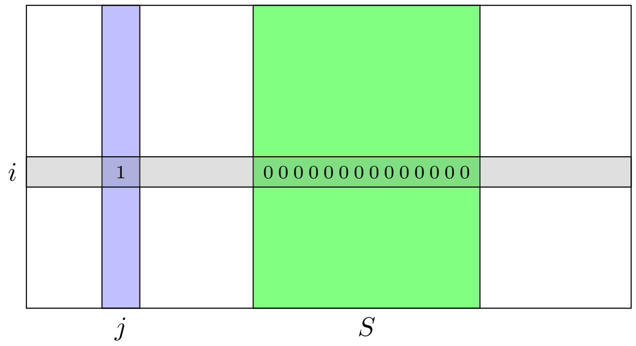

Fig. 1. A d-disjunct matrix has the following property: for any subset S of d (not necessarily contiguous) columns, and any column j that is not present in S, there exists a row i that has a 1 in column j and all zeros in S.

Minimize number of tests

It is remarkable that d-disjunct matrices not only allow for linear time decoding, which is a vast improvement over the brute-force algorithm for separable matrices, but also have asymptotically the same number of tests as d-separable matrices [4]. Let $t(d, N)$ denote the minimum number of rows of an N-column d-disjunct matrix. It has been known for about 40 years [5] that $t(\Omega(\sqrt{N}), N) = \Theta(N)$, and for $d = O(\sqrt{N})$ we have

$$\Omega\left(\frac{d^2}{\log d}\log N\right) \;\le\; t(d,N) \;\le\; O(d^2 \log N). \tag{1}$$

A $t \times N$ d-disjunct matrix with $t = O(d^2 \log N)$ rows can be constructed randomly or even deterministically (see [11]). However, the decoding time $O(tN)$ of the naïve decoding algorithm is still too slow for modern applications, where in most cases $d \ll N$ and thus $t \ll N$.

Efficient decoding

An ideal decoding time would be in the order of $\mathrm{poly}(d, \log N)$, which is sub-linear in N for practical ranges of d. Ngo, Porat, and Rudra [10] showed how to achieve this goal using a couple of ideas: (a) two-layer test matrix construction, and (b) code concatenation using a list recoverable code.

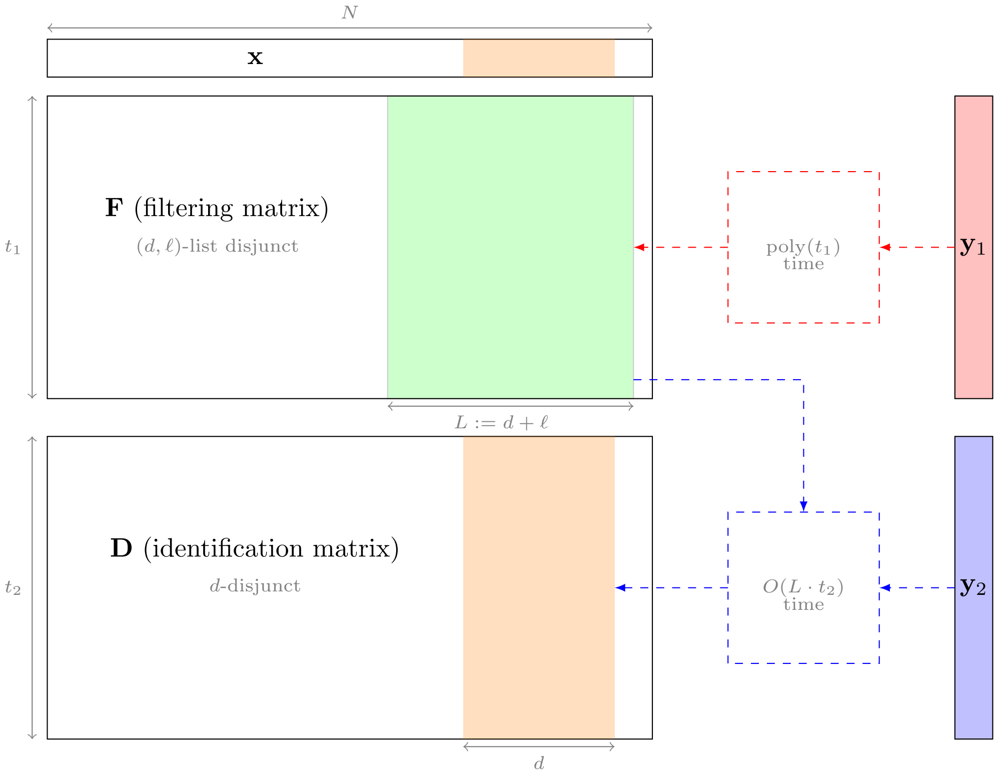

Fig. 2. The vector $\mathbf{x}$ denotes the characteristic vector of the d positives (illustrated by the orange box). The final matrix is the stacking of $\mathbf{F}$, which is a (d, ℓ)-list disjunct matrix, and $\mathbf{D}$, which is a d-disjunct matrix. The result vector is naturally divided into $\mathbf{y}_1$ (the part corresponding to $\mathbf{F}$ and denoted by the red vector) and $\mathbf{y}_2$ (the part corresponding to $\mathbf{D}$ and denoted by the blue vector). The decoder first uses $\mathbf{y}_1$ to compute a superset of the set of positives (denoted by green box), which is then used with $\mathbf{y}_2$ to compute the final set of positives. The first step of the decoding is represented by the red dotted box while the second step (naïve decoder) is denoted by the blue dotted box.

Two-layer test matrix construction. The idea is to construct $\mathbf{M}$ by stacking on top of one another two matrices: a "filtering" matrix $\mathbf{F}$ and an "identification" matrix $\mathbf{D}$. (See Fig. 2.) The filtering matrix is used to quickly identify a "small" set of L candidate items including *all* the positives. Then, the identification matrix is used to pinpoint precisely the positives. For example, let $\mathbf{D}$ be any d-disjunct matrix, and that from the tests corresponding to the rows of $\mathbf{F}$ we can produce a set S of $L = \text{poly}(d, \log N)$ candidate items in time $\text{poly}(d, \log N)$. Then, by running the naïve decoding algorithm on S using test results corresponding to the rows of $\mathbf{D}$, we can identify all the positives in time $\text{poly}(d, \log N)$. To formalize the notion of "filtering matrix," we borrow a concept from coding theory, where producing a small list of candidate codewords is the *list decoding problem* [6].

Definition 3 (List-disjunct matrix). *Let $d + \ell \leq N$ be positive integers. A matrix $\mathbf{F}$ is (d, ℓ)-list-disjunct if and only if $\bigcup_{j \in T} \mathbf{M}^j \setminus \bigcup_{k \in S} \mathbf{M}^k \neq \emptyset$ for any two disjoint sets S and T of columns of $\mathbf{F}$ with $|S| = d$ and $|T| = \ell$. (See Fig. 3.)*

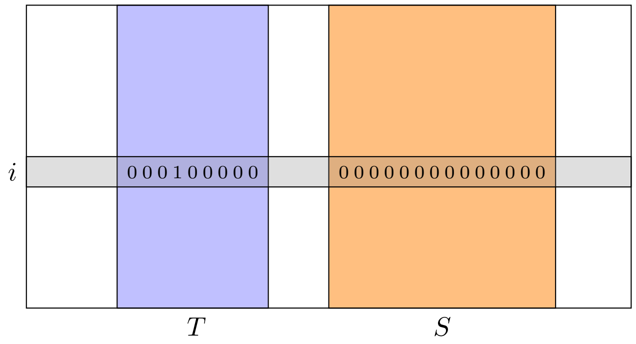

Fig. 3. A (d, ℓ)-list-disjunct matrix satisfies the following property: for any subset S of size d and any disjoint subset T of size ℓ, there exists a row i that has a 1 in at least one column in T and all zeros in S.

Note that a matrix is d-disjunct matrix iff it is $(d, 1)$-list-disjunct. However, the relaxation to $\ell = \Theta(d)$ allows the existence (and construction) of $(d, O(d))$-list disjunct matrices with $\Theta(d \log(N/d))$ rows. The existence of such small list disjunct matrices is crucially used in the second idea below.

(b) Code Concatenation with list recoverable codes A $t \times N$ (d, ℓ)-list-disjunct matrix admits $O(tN)$-decoding time using the naïve decoding algorithm. However, to achieve $\text{poly}(d, \log N)$ decoding time overall, we will need to construct list-disjunct matrices that allow for a $\text{poly}(d, \log N)$ decoding time. In particular, to use such a matrix as a filtering matrix, it is necessary that $\ell = \text{poly}(d)$. To construct efficiently decodable list disjunct matrices, we need other ideas. Ngo, Porat, and Rudra [10] used a connection to list recoverable codes [6] to construct such matrices. This connection was used to construct $(d, O(d^{3/2}))$-list disjunct matrices with $t = o(d^2 \log_d N)$ rows that can be decoded in $\text{poly}(t)$ time. This along with the construction in Fig. 2 implies the following result:

Theorem 1 ([10]). *Given any d-disjunct matrix, it can be converted into another matrix with $1 + o(1)$ times as many rows that is also efficiently decodable (even if the original matrix was not).*

Other constructions of list disjunct matrices with worse parameters were obtained earlier by Indyk, Ngo and Rudra [7] and Cheraghchi [1] using connections to expanders and randomness extractors.

Applications

Heavy hitter is one of the most fundamental problems in data streaming [8]. Cormode and Muthukrishnan [2] showed that a NACGT scheme that is efficiently decodable and is also *explicit* solves a natural version of the heavy hitters problem. An explicit construction means one needs an algorithm that outputs a column or a specific entry of $\mathbf{M}$ instead of storing the entire matrix $\mathbf{M}$ which can be extremely space consuming. This is possible with Theorem 1 by picking the filtering and decoding matrices to be explicit.

Another important generalization of NACGT matrices are those that can handle errors in the test outcomes. Again this is possible with the construction of Fig. 2 if the filtering and decoding matrices are also error-tolerant. The list disjunct matrices constructed by Cheraghchi are also error-tolerant [1].

Open Problems

The outstanding open problem in group testing theory is to close the gap (1). An explicit construction of (d, d)-list-disjunct matrices is not known; solving this problem will lead to a scheme that is (near-)optimal in all desired objectives.

Recommended Reading

1. Cheraghchi M (2013) Noise-resilient group testing: Limitations and constructions. Discrete Applied Mathematics 161(1-2):81–95
2. Cormode G, Muthukrishnan S (2005) What's hot and what's not: tracking most frequent items dynamically. ACM Trans Database Syst 30(1):249–278
3. Dorfman R (1943) The detection of defective members of large populations. The Annals of Mathematical Statistics 14(4):436–440
4. Du DZ, Hwang FK (2000) Combinatorial group testing and its applications, Series on Applied Mathematics, vol 12, 2nd edn. World Scientific Publishing Co. Inc., River Edge, NJ
5. D′yachkov AG, Rykov VV (1982) Bounds on the length of disjunctive codes. Problemy Peredachi Informatsii 18(3):7–13
6. Guruswami V (2004) List Decoding of Error-Correcting Codes (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition), Lecture Notes in Computer Science, vol 3282. Springer
7. Indyk P, Ngo HQ, Rudra A (2010) Efficiently decodable non-adaptive group testing. In: Proceedings of the Twenty First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2010), ACM, New York, pp 1126–1142
8. Muthukrishnan S (2005) Data streams: algorithms and applications. Foundations and Trends in Theoretical Computer Science 1(2)
9. Ngo HQ, Du DZ (2000) A survey on combinatorial group testing algorithms with applications to DNA library screening. In: Discrete mathematical problems with medical applications (New Brunswick, NJ, 1999), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol 55, Amer. Math. Soc., Providence, RI, pp 171–182
10. Ngo HQ, Porat E, Rudra A (2011) Efficiently decodable error-correcting list disjunct matrices and applications - (extended abstract). In: ICALP (1), pp 557–568
11. Porat E, Rothschild A (2011) Explicit nonadaptive combinatorial group testing schemes. IEEE Transactions on Information Theory 57(12):7982–7989