Hung Q. Ngo
Last update: May 27, 2011

Efficiently decodable non-adaptive group testing schemes

This lecture is on two methods for constructing group testing matrices which can be decoded efficiently (in time poly$(d, \log N)$). The first method, based on code concatenation, was first observed in [17] and exploited to its full potential in [19]. The second method, also from [19], is based on a simple recursive construction. One of the key ingredients of the first method is a combinatorial object called *list-disjunct matrix*. There is also a notion of *list-separable matrix* which is closely related to list-disjunct matrix. List-disjunct/separable matrices are interesting on their own, with applications beyond the group testing problem. We shall discuss the applications in a later lecture. The other key ingredient of the first method is the notion of list-recoverable codes, which will be briefly introduced here too.

1 Efficient decoding and list-disjunct/separable matrices

1.1 Top level view

To construct an efficiently decodable group testing matrix, the main idea is to stack on top of one another a "*filtering*" matrix $\mathbf{F}$ and an "*identification*" matrix $\mathbf{D}$. The filtering matrix is used to identify quickly a "small" set of L candidate items which include *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 $\mathbf{F}$ be any matrix satisfying the filtering property above. Further assume 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 poly$(d, \log N)$. Then, by running the naive decoding algorithm on S using $\mathbf{D}$, we can identify all the positives in time poly$(d, \log N)$. This idea is somewhat analogous to the constructions of (efficiently decodable or not) compressive sensing schemes [5, 8, 13].

1.2 List-separable and list-disjunct matrices

The obvious problem is to formalize the notion of "filtering matrix." In coding theory, producing a small list of candidate codewords is the *list decoding* problem []. We borrow the intuition from list decoding and from separable matrices to define the following "filtering" matrix.

Definition 1.1 ((d, l)-list-separable)**.** Given positive integers t, d, l, N where $d + l \leq N$, a $t \times N$ binary matrix $\mathbf{M}$ is said to be (d, l)*-list-separable* if it satisfies the following condition. For any $\mathbf{y} \in \{0, 1\}^t$ there exists a column set $R_{\mathbf{y}}$ such that, if T is any set of at most d columns of $\mathbf{M}$ whose union is $\mathbf{y}$, then $T \subseteq R_{\mathbf{y}}$ and $|R_{\mathbf{y}}| \leq |T| + l - 1$.

Note that a $(d, 1)$-list-separable matrix is precisely a d-separable matrix. Suppose we have a group testing matrix with a corresponding decoding algorithm which produces for every outcome vector $\mathbf{y} \in \{0, 1\}^t$ a candidate set $R_{\mathbf{y}}$ including all the positives plus $< l$ negative items, then the matrix is (d, l)-list-separable. (If $\mathbf{y}$ does not correspond to any outcome vector then we can set $R_{\mathbf{y}} = \emptyset$.)

The next natural line of development is to define and study list-disjunct matrices.

Definition 1.2 ((d, l)-list-disjunct)**.** A matrix $\mathbf{M}$ is called (d, l)*-list-disjunct* if the naive decoding algorithm always returns an item set which includes *all* the positives plus $< l$ negative items.

Exercise 1.3 (Another definition of (d, l)-list-disjunct matrices)**.** Let $d + l \leq N$ be positive integers. Show that a matrix is (d, l)-list-disjunct if and only if, for any two disjoint sets S and T of columns of $\mathbf{M}$ with $|S| = l$ and $|T| = d$, there exists a row of $\mathbf{M}$ in which some column in S has a 1 but all columns in T have 0s.

Proposition 1.4. *Every (d, l)-list-disjunct matrix is a (d, l)-list-separable matrix. Conversely, every (d, l)-list-separable matrix is a $(d - 1, l)$-list-disjunct matrix.*

Exercise 1.5. Prove Proposition 1.4.

1.3 Relations to earlier works and applications

The name "list disjunct" was coined in [17], though it was previously studied under the different names: (d, n, ℓ)*-super-imposed codes* in [7, 9], *list-decoding super-imposed codes* of strength d and list-size ℓ in [22].[1] We stick with the "list disjunct matrices" in this lecture.

Shortly prior to [17], Cheraghchi [6] studied the notion of *error-correcting measurement matrices* for d-sparse vectors, which are slightly more general than the notion of list-separable matrices. Since list-separable matrices are equivalent to list-disjunct matrices with a slight loss in parameters, the results in [6], though shown for list-separable matrices, apply for list-disjunct matrices as well. Conversely, our bounds also apply to error-correcting measurement matrices. We will prove lower bounds which slightly improve lower bounds in [6].

List-disjunct matrices were used in [17] to construct efficiently decodable disjunct matrices. They also presented a "stand alone" application of list disjunct matrices in constructing sparsity separators, which were used by Ganguly [12] to design data stream algorithms for the sparsity problem. Rudra and Uurtamo [23] used these objects to design data stream algorithms for tolerant testing Reed-Solomon codes. As observed in De Bonis et. al. [7], these objects can also be used to construct optimal two stage group testing algorithms (where one is allowed to perform tests in two stages, and the second stage of tests can depend on the the first stage's results).

List disjunct matrices are also similar to other combinatorial objects such as selectors [16] and multi-user tracing families [2]. Selectors have found numerous applications such as broadcasting in unknown directed networks and designing tests for coin-weighting problems [16] as well as designing optimal two stage group testing schemes [7]. Multi-user tracing families were used to construct monotone encodings in [2]. Monotone encodings can be used for designing secure vote storage systems [18]. Selectors' requirements are slightly stronger than list-disjunct matrices. Both selectors and list-disjunct matrices can be used to recover a super-set of the defective set, while multi-user tracing families are used to recover a sub-set of the defective set. It is not clear if any two of these objects are "equivalent." However, as we shall see, list-disjunct matrices can be used to construct some of these objects.

1.4 Simple bounds

From the above proposition, the optimal number of rows of a list-disjunct and list-separable matrices are asymptotically the same. Thus, we shall only study the optimal number of rows of a list-disjunct matrices.

[1]The authors of [17] were not aware of these previous works.

Let $t(d,l,N)$ denote the minimum number of rows of a (d,l)-list-disjunct matrix with N columns. This section derives a couple of simple bounds for this function.

Proposition 1.6 (Proposition 2 in [9]). *Given positive integers $N \geq d+l$, we have*

$$t(d,l,N) \geq \log \binom{N}{d} - \log \binom{d+l-1}{d}.$$

Exercise 1.7. Prove Proposition 1.6.

The following lower bound for (d,l)-list-disjunct matrices is better than the similar bound proved in [7] in two ways: (1) the actual bounds are slightly better, and (2) the bound in [7] requires a precondition that $N > d^2/(4l)$ while ours does not. We make use of the argument from Erdős-Frankl-Füredi [10, 11], while [7] uses the argument from Ruszinkó [24] as presented in Alon-Asodi [1].

Lemma 1.8. *For any positive integers N, d, l with $N \geq d+l$, we have*

$$t(d,l,N) > d \log \left(\frac{N}{d+l-1} \right). \tag{1}$$

When $d \geq 2l$, the following bound holds

$$t(d,l,N) > \frac{\lfloor d/l \rfloor (d+2-l)}{2 \log \left(e \lfloor d/l \rfloor (d+2-l)/2\right)} \log \left(\frac{N-d-2l+2}{l} \right). \tag{2}$$

Proof. Proposition 1.6 easily yields (1)

$$t(d,l,N) \geq \log \left(\frac{\binom{N}{d}}{\binom{d+l-1}{d}} \right) = \log \frac{N \cdots (N-d+1)}{(d+l-1) \cdots l} \geq \log \left(\frac{N}{d+l-1} \right)^d = d \log \frac{N}{d+l-1}.$$

Consider the case when $d \geq 2l$. Let $\mathbf{M}$ be a $t \times N$ (d,l)-list-disjunct matrix. Fix a positive integer $w \leq t$ to be determined later. Let $\mathcal{C}$ denote the collection of all columns of $\mathbf{M}$, and think of $\mathcal{C}$ as a set family on $[t]$. Then, $\mathcal{C}$ satisfies the property that the union of any l members of $\mathcal{C}$ is not covered by the union of any other d members of $\mathcal{C}$. For any $C \in \mathcal{C}$, a subset $X \subseteq C$ is called a *private subset* of C if X is not a subset of any other C' in $\mathcal{C}$. Partition $\mathcal{C}$ into three sub-collections

$$\mathcal{C} = \mathcal{C}^{\text{p}}_{\geq w} \cup \mathcal{C}^{\text{np}}_{\geq w} \cup \mathcal{C}_{<w}$$

defined as follows.

$$\begin{aligned}
\mathcal{C}^{\text{p}}_{\geq w} &:= \{C \in \mathcal{C} \;:\; |C| \geq w \text{ and } C \text{ has a private } w\text{-subset}\} \\
\mathcal{C}^{\text{np}}_{\geq w} &:= \{C \in \mathcal{C} \;:\; |C| \geq w \text{ and } C \text{ has no private } w\text{-subset}\} \\
\mathcal{C}_{<w} &:= \{C \in \mathcal{C} \;:\; |C| < w\}.
\end{aligned}$$

We make three claims.

Claim 1. If $w \leq t/2$ then $|\mathcal{C}^{\text{p}}_{\geq w}| + \left\lfloor \frac{|\mathcal{C}_{<w}|}{l} \right\rfloor \leq \binom{t}{w}$.

Claim 2. Let $C_1, \cdots, C_l$ be any l different members of $\mathcal{C}^{\text{np}}_{\geq w}$. For any integer $j \leq d/l - 1$ and any sub-collection $\mathcal{D} \subseteq \mathcal{C} \setminus \{C_1, \cdots, C_l\}$ such that $|\mathcal{D}| = jl$, we have

$$\left| \bigcup_{i=1}^{l} C_i \setminus \bigcup_{D \in \mathcal{D}} D \right| \geq (d - (j+1)l + 1)w + 1.$$

Claim 3. If $w \geq \frac{2(t - \lfloor d/l \rfloor)}{\lfloor d/l \rfloor (d+2-l)}$, then $|\mathcal{C}^{\text{np}}_{\geq w}| \leq d + l - 1$.

Let us complete the proof of the lemma before proving the claims. Set $w = \left\lceil \frac{2(t - \lfloor d/l \rfloor)}{\lfloor d/l \rfloor (d+2-l)} \right\rceil$. Then, $w \leq t/2$ when $d \geq 2l$. Note that $w < \bar{w} = \frac{2t}{\lfloor d/l \rfloor (d+2-l)}$ and the function $(te/w)^w$ is increasing in w when $w \in [0, t]$. From Claims 1 and 3,

$$\begin{aligned}
n = |\mathcal{C}| &= \left(|\mathcal{C}^{\text{p}}_{\geq w}| + |\mathcal{C}_{<w}| \right) + |\mathcal{C}^{\text{np}}_{\geq w}| \\
&\leq \left(l \left(|\mathcal{C}^{\text{p}}_{\geq w}| + \left\lfloor \frac{|\mathcal{C}_{<w}|}{l} \right\rfloor \right) + (l-1) \right) + d + l - 1 \\
&\leq l \binom{t}{w} + d + 2l - 2 \\
&\leq l(te/w)^w + d + 2l - 2 \\
&\leq l(te/\bar{w})^{\bar{w}} + d + 2l - 2.
\end{aligned}$$

Inequality (2) follows.

We now prove Claim 1. Let $\mathcal{P}_1$ be a collection of private w-subsets of sets in $\mathcal{C}^{\text{p}}_{\geq w}$ such that $\mathcal{P}_1$ contains exactly one private w-subset per set in $\mathcal{C}^{\text{p}}_{\geq w}$. Let $\mathcal{L}$ be an arbitrary sub-collection of exactly l different members of $\mathcal{C}_{<w}$, namely $\mathcal{L} \subseteq \mathcal{C}_{<w}$ and $|\mathcal{L}| = l$. Then, there must exist $C \in \mathcal{L}$ such that such that C is **not** a subset of any set in $\mathcal{P}_1 \cup \mathcal{C}_{<w} \setminus \mathcal{L}$. Otherwise, the union of sets in $\mathcal{L}$ will be covered by the union of at most $l \leq d$ sets in $\mathcal{C}$. We refer to such C as a *representative* of $\mathcal{L}$. For each $\mathcal{L}$, pick an arbitrary representative of $\mathcal{L}$ to be *the* representative of $\mathcal{L}$. Partition $\mathcal{C}_{<w}$ into $\left\lfloor \frac{|\mathcal{C}_{<w}|}{l} \right\rfloor$ sub-collections of cardinalities l each, plus possibly one extra sub-collection whose size is less than l. Let $\mathcal{P}_2$ be the set of the representatives of the first $\left\lfloor \frac{|\mathcal{C}_{<w}|}{l} \right\rfloor$ sub-collections. Then, $\mathcal{P}_1 \cup \mathcal{P}_2$ is a Sperner family, each of whose members is of cardinality at most w. For $w \leq t/2$, it is well-known (see, e.g., [4]) that $|\mathcal{P}_1 \cup \mathcal{P}_2| \leq \binom{t}{w}$. Noting that $|\mathcal{P}_2| = \left\lfloor \frac{|\mathcal{C}_{<w}|}{l} \right\rfloor$ and $|\mathcal{P}_1| = |\mathcal{C}^{\text{p}}_{\geq w}|$, Claim 1 follows.

Next, we prove Claim 2. Assume for the contrary that

$$\left| \bigcup_{i=1}^{l} C_i \setminus \bigcup_{D \in \mathcal{D}} D \right| \leq (d - (j+1)l + 1)w$$

for some $\mathcal{D}$ and j satisfying the conditions in the claim. For every $i \in [l]$, define

$$\begin{aligned}
C'_i &:= C_i \setminus \bigcup_{D \in \mathcal{D}} D \cup C_1 \cdots \cup C_{i-1}. \\
x_i &:= \left\lfloor \frac{|C'_i|}{w} \right\rfloor \\
y_i &:= |C'_i| \bmod w.
\end{aligned}$$

Then,

$$(d-(j+1)l+1)w \geq \left|\bigcup_{i=1}^{l} C_i \setminus \bigcup_{D\in\mathcal{D}} D\right| = \sum_{i=1}^{l}|C_i'| = \sum_{i=1}^{l}(x_i w + y_i) = w\left(\sum_{i=1}^{l} x_i\right) + \sum_{i=1}^{l} y_i.$$

Partition C_i' into x_i parts of size w each and one part of size $y_i \leq w-1$. First, assume $\sum_{i=1}^{l} y_i > 0$, then $\sum_{i=1}^{l} x_i \leq d-(j+1)l$. Because C_i has no private w-subset (and thus no private y_i-subset), the set C_i' can be covered by at most x_i+1 other sets in $\mathcal{C}$. The union $\bigcup_{i\in[l]} C_i'$ can be covered by at most $\sum_{i=1}^{l} x_i + l \leq d-jl$ sets in $\mathcal{C}$. Those $d-jl$ sets covering the C_i' along with jl sets in $\mathcal{D}$ cover the l sets $C_i, i\in[l]$, which is a contradiction. Second, when $\sum_{i=1}^{l} y_i = 0$ we only need $\sum_{i=1}^{l} x_i \leq d-(j+1)l+1 \leq d-jl$ sets to cover the C_i'. The same contradiction is reached.

Finally we prove Claim 3. Suppose $|\mathcal{C}^{\text{np}}_{\geq w}| \geq d+l$. Consider $d+l$ sets $C_1,\dots,C_{d+l}$ in $\mathcal{C}^{\text{np}}_{\geq w}$. For $j=0,1,\cdots,\lfloor d/l\rfloor-1$, define $\mathcal{D}_j = \{C_1,\cdots,C_{jl}\}$. ($\mathcal{D}_0=\emptyset$.) Then, noting Claim 2, we have

$$\begin{aligned} t &\geq \bigcup_{i=1}^{d+l} C_i \\ &\geq \sum_{j=0}^{\lfloor d/l\rfloor-1}\left|\bigcup_{i=jl+1}^{(j+1)l} C_i\setminus\mathcal{D}_j\right| + \left|\bigcup_{i=d+1}^{d+l} C_i \setminus \bigcup_{i=1}^{d} C_i\right| \\ &\geq \sum_{j=0}^{\lfloor d/l\rfloor-1}\left[\big(d-(j+1)l+1\big)w+1\right]+1 \\ &= w\lfloor d/l\rfloor\left[d+1-l(\lfloor d/l\rfloor+1)/2\right]+\lfloor d/l\rfloor+1 \\ &\geq \frac{1}{2}w\lfloor d/l\rfloor(d+2-l)+\lfloor d/l\rfloor+1, \end{aligned}$$

which contradicts the assumption that $w \geq \frac{2(t-\lfloor d/l\rfloor)}{\lfloor d/l\rfloor(d+2-l)}$. □

Theorem 1.9. *Given positive integers $N \geq d+l$. Then,*

$$t(d,l,N) \leq 2d\left(\frac{d}{l}+1\right)\left(\log\frac{N}{d+l}+1\right).$$

Proof. Fix positive integers n,q to be determined. Let $\mathbf{M}$ be the concatenation of the random code C_{out} and the identity code $C_{\text{in}} = \text{ID}_q$. The random code is of length n, each of whose positions is chosen randomly from an alphabet of size q. Consider a subset T of d codewords and a disjoint subset S of l codewords. For each position $i\in[n]$, let T_i and S_i denote the set of symbols codewords in T and S have at that position, respectively. The probability that $S_i \subseteq T_i$ is at most $(d/q)^l$. Hence, the probability that $S_i\subseteq T_i$ for all $i\in[n]$ is at most $(d/q)^{ln}$. Pick $n = 2\left(\frac{d}{l}+1\right)\left(\log\frac{N}{d+l}+1\right)$, $q=3d\geq ed$, and taking the union bound over all choices of S and T, we obtain

$$\begin{aligned} \text{Prob}[\mathbf{M}\text{ is not }(d,l)\text{-list-disjunct}] &\leq \binom{N}{d+l}\binom{d+l}{l}(d/q)^{ln} \\ &\leq \exp\left((d+l)\ln\frac{Ne}{d+l}+l\ln\frac{(d+l)e}{l}-ln\right) \\ &< 1. \end{aligned}$$

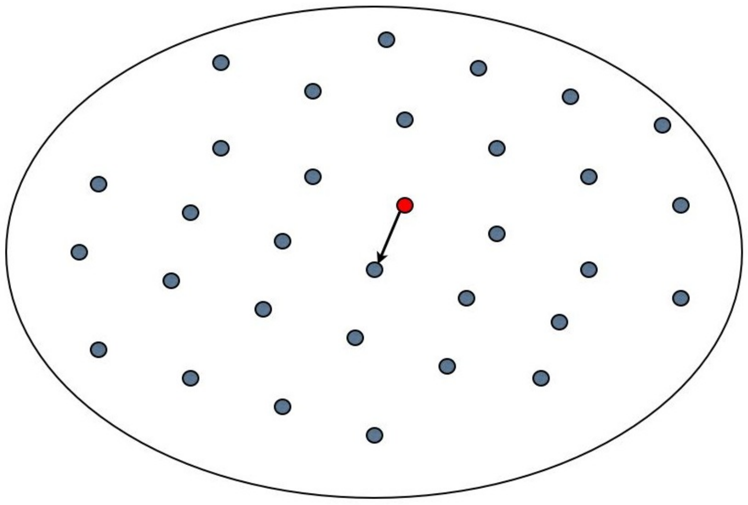
Figure 1: The usual decoding problem

□

Corollary 1.10. *When $l = \Omega(d)$, we do have a nice reduction in the number of tests compared to the d-disjunct case: $t(d, \Omega(d), N) = O(d \log(N/d))$.*

Corollary 1.11 (Optimal two-stage group testing)**.** *Consider the adaptive group testing problem where the tests are performed in two stages: the second set of tests can be designed after seeing the first test set results. Then, the optimal number of tests is $\Theta(d \log(N/d))$.*

Proof. For any adaptive group testing scheme, $\Omega(d \log(N/d))$ tests are necessary, because information theoretically there are $\sum_{i=0}^{d} \binom{n}{i} = 2^{\Omega(d \log(N/d))}$ possible sets of positives.

A two stage group testing scheme with $O(d \log(N/d))$ tests can be designed as follows. We first use a (d, d)-list-disjunct matrix to identify a set of at most $2d$ items including all the positives. Then, an identity matrix of order $2d$ is used for the second stage to identify precisely the positives. □

2 List recoverable codes and its role in designing efficient decoders

2.1 List recovery

The usual decoding problem is the following: given a *received word* $\mathbf{y}$ which is not necessarily a codeword, recover a near-by codeword $\mathbf{c}$. For example, if $\mathbf{y} =$ *com***t***lem***a***nt* we might want to recover $\mathbf{c} =$ *com***p***lem***e***nt*. See Figure 1 for an illustration.

In many cases, if we relax the unique decoding requirement, allowing the decoding algorithm to produce a small *list* of possible codewords, we will be able to design codes with a better rate/distance tradeoff. This is the *list decoding* problem, illustrated in Figure 2. For example, if $\mathbf{y} =$ *compl***b***ment* then we might want to recover the list { *compl***e***ment, compl***i***ment* }.

In the *list recovery* problem, each position $i \in [n]$ has a (small) set S_i of characters. We want to return a

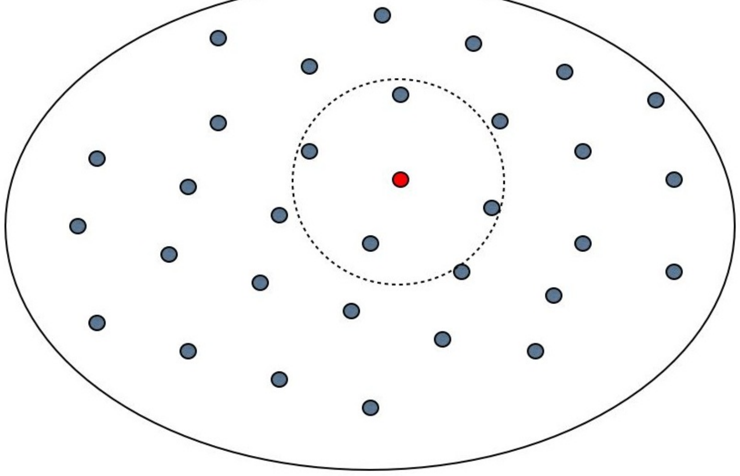

Figure 2: The list decoding problem

list of codewords agreeing with a large fraction of the sets. For example,

$$\begin{bmatrix}\{c,f\}\\ \{a,o\}\\ \{t,r\}\\ \{b,h\}\\ \{e,s\}\\ \{a,r\}\end{bmatrix} \Rightarrow \begin{bmatrix}f\\ a\\ t\\ h\\ e\\ r\end{bmatrix}, \begin{bmatrix}m\\ o\\ t\\ h\\ e\\ r\end{bmatrix}$$

Let $\ell, L \geq 1$ be integers and let $0 \leq \alpha \leq 1$. A q-ary code C of block length n is called (α, ℓ, L)-*list recoverable* if for every sequence of subsets $S_1, \ldots, S_n$ such that $|S_i| \leq \ell$ for every $i \in [n]$, there exists at most L codewords $\mathbf{c} = (c_1, \ldots, c_n)$ such that for at least αn positions i, $c_i \in S_i$. A $(1, \ell, L)$-list recoverable code will be henceforth referred to as (ℓ, L)-zero error list recoverable. We will need the following powerful result due to Parvaresh and Vardy[2]:

Theorem 2.1 ([20]). *For all integers $s \geq 1$, for all prime powers r and all powers q of r, every pair of integers $1 < k \leq n \leq q$, there is an explicit $\mathbb{F}_r$-linear map $E : \mathbb{F}_q^k \to \mathbb{F}_{q^s}^n$ such that:*

1. *The image of E, $C \subseteq \mathbb{F}_{q^s}^n$, is a code of minimum distance at least $n - k + 1$.*

2. *Provided*
$$\alpha > (s+1)(k/n)^{s/(s+1)}\ell^{1/(s+1)}, \tag{3}$$
C is an $(\alpha, \ell, O((rs)^s n\ell/k))$-list recoverable code. Further, a list recovery algorithm exists that runs in $\text{poly}((rs)^s, q, \ell)$ *time.*

In the above, the s'th "order" Parvaresh-Vardy code will be referred to as the PV^s code. PV^1 is the well-known Reed-Solomon codes and will be referred to as the RS code.

Remark 2.2. For RS codes, the multiplicative factor of $(s+1)$ can be removed from (3).

[2]This statement of the theorem appears in [14].

Because we will mostly use the above theorem for the $r = 2$ case, let us re-state this special case, and also re-state the special case of corresponding to the RS code.

Theorem 2.3 ([20]). *For all positive integers $s \geq 1$, $q = 2^m$ for some positive integer m, every pair of integers $1 < k \leq n \leq q$, there is an explicit $\mathbb{F}_2$-linear map $E : \mathbb{F}_{2^m}^k \to \mathbb{F}_{2^{ms}}^n$ such that:*

1. *The image of E, $C \subseteq \mathbb{F}_{2^{ms}}^n$, is a code of minimum distance at least $n - k + 1$.*
2. *Provided*
$$\alpha > (s+1)(k/n)^{s/(s+1)}\ell^{1/(s+1)}, \tag{4}$$
C is an $(\alpha, \ell, O(s^s n\ell/k))$-list recoverable code. Further, a list recovery algorithm exists that runs in $\text{poly}(s^s, q, \ell)$ time.
3. *When $s = 1$, the code is the RS code which is $(\alpha, \ell, O(nl/k))$-list-recoverable as long as*
$$\alpha > \sqrt{k\ell/n}.$$

2.2 Efficiently decodable list separable and disjunct matrices

Let C_{out} be the $[n, k]_q$-RS code for q some power of 2. Let C_{in} be any $(d, d+1)$-list-disjunct matrix with q columns and t_{in} rows. We know from Section 1.4, for example, that there exists a $(d, d+1)$-list-disjunct matrix with q columns and $t_{\text{in}} = O(d\log(q/d))$ rows.

Let $\mathbf{M} = C_{\text{out}} \circ C_{\text{in}}$. We claim that $\mathbf{M}$ is a list-separable matrix which can be efficiently decoded. The decoding algorithm works as follows. From the t_{in} test results for each position $i \in [n]$, we run the naive decoding algorithm for C_{in} to recover a set S_i of at most $l = 2d$ columns of C_{in}. These columns naturally correspond to a set S_i (overloading notation) of symbols of the outer code. As long as $1 > kl/n$, Theorem 2.3 ensures that there is a $\text{poly}(q, l)$-time algorithm which recovers a list $L = O(nl/k)$ codewords each of which agrees with all the S_i. These codewords certainly contain all of the positives.

To minimize the number of tests, which is $O(n \cdot t_{\text{in}}) = O(nd\log(q/d))$, we can choose the parameters as follows.

$$\begin{aligned} n &= q \\ q &= \frac{4d\log N}{\log(2d\log N)} \\ k &= \frac{\log N}{\log q}. \end{aligned}$$

We need to verify that $kl < n$ which is the same as $2d\log N < q\log q$. Note that

$$q\log q = \frac{4d\log N}{\log(4d\log N)}\log\left(\frac{4d\log N}{\log(4d\log N)}\right) = (2d\log N)\cdot 2\left(1 - \frac{\log\log(4d\log N)}{\log(4d\log N)}\right) > 2d\log N$$

with sufficiently large N. The total number of tests is

$$t = O\left(\frac{4d^2\log N}{\log(4d\log N)}\log\left(\frac{4\log N}{\log(4d\log N)}\right)\right) = O(d^2\log N).$$

The total decoding time is $O(nqt_{\text{in}} + \text{poly}(q, l)) = \text{poly}(t)$. Stacking this efficiently decodable (d, L)-list-separable matrix with any d-disjunct matrix, we obtain an efficiently decodable d-disjunct matrix with the best known number of tests.

Theorem 2.4. *By concatenating the RS code with a good list-disjunct inner code (i.e. matrix) and stack the result on top of a good d-disjunct matrix, we obtain a d-disjunct matrix with $t = O(d^2 \log N)$ rows which is decodable in* $\text{poly}(t)$*-time.*

There are two natural questions:

1. *If we also want an explicit or strongly explicit construction, can we still attain $t = O(d^2 \log N)$ tests?* If the list-disjunct inner matrix is (strongly) explicit, then certainly the overall construction is (strongly) explicit. Unfortunately, so far we do not know of any explicit (let alone strongly explicit) construction of $(d, O(d))$-list-disjunct matrices. For example, a construction along the Porat-Rothschild route [21] would be really nice to have. One can use the set restriction framework [3] to attain an explicit construction with running time $O(N^d)$, which is not very good. In later sections, we shall discuss several strongly explicit constructions of list-disjunct matrices which come close to the optimal number of tests.

2. *Some applications might require only efficiently decodable* list*-disjunct or* list*-separable matrices. The above construction is actually not very good compared to the optimal number of tests of a list-disjunct matrix. Can we do better?* One answer is to use the above strategy with PV^s codes for $s > 1$. We shall take this route below.

Open Problem 2.5. Find a (strongly or not) explicit construction of $(d, O(d))$-list-disjunct matrices attaining the probabilistic $t = O(d \cdot \log(N/d))$ bound. Or more generally construct (d, l)-list-disjunct matrices for any l.

We first prove a generic lemma where the outer code is the PV^s code and the inner code is an arbitrary (d, l)-list-disjunct matrix. Later we shall apply the lemma by "plugging-in" different values of s and different constructions of (d, l)-list-disjunct matrices. What is interesting about this lemma is that it shows a *black-box* conversion procedure which converts a (family of) list-disjunct matrix into another one which is efficiently decodable.

Lemma 2.6 (Black-box conversion using list-recoverable codes)**.** *Let $\ell, d \geq 1$ be integers. Assume that for every $Q \geq d$ there exists a (d, ℓ)-list-disjunct matrix with $\bar{t}(d, \ell, Q)$ rows and Q columns. For all integers $s \geq 1$ and $N \geq d$, define*

$$A(d, l, s) = (d + l)^{1/s}(s + 1)^2.$$

Let k be minimum integer such that $k \log(kA(d, l, s)) \geq \log N$, and q be the minimum power of 2 such that $q > kA(d, l, s)$. Then, there exists a (d, L)-list separable $t \times N$ matrix $\mathbf{M}$ with the following properties:

(i) $t = O\left(s^2 \cdot (d + \ell)^{1/s} \cdot \left(\frac{\log N}{\log q}\right) \cdot \bar{t}(d, \ell, q^s)\right)$

(ii) $L = s^{O(s)} \cdot (d + \ell)^{1+1/s}$.

(iii) It is decodable in time $t^{O(s)}$.

Furthermore, if the $\bar{t}(d, l, Q) \times Q$ matrix is (strongly) explicit then $\mathbf{M}$ is (strongly) explicit.

Proof. Let $\mathbf{M}$ be the concatenation of $C_{\text{out}} = \text{PV}^s$ with C_{in} which is a (d, l)-list-disjunct matrix with $\bar{t}(d, \ell, Q)$ rows and $Q = q^s$ columns. We will have to choose parameters $1 < k \leq n \leq q$ so that the followings hold:

$$\begin{aligned} N &\leq q^k \\ 1 &> (s+1)^{s+1}(k/n)^s(d+l) \text{ (to satisfy (3))} \end{aligned}$$

The inequalities are satisfied when we pick $n = q$ and q, k satisfy the conditions stated in the statement of the lemma. The number of rows of $\mathbf{M}$ is

$$\begin{aligned} t &= n\bar{t}(d,l,Q) \\ &\le 2kA(d,l,s)\bar{t}(d,l,Q) \\ &\le 2\frac{\log N}{\log(kA(d,l,s))}A(d,l,s)\bar{t}(d,l,Q) \\ &\le 2\frac{\log N}{\log(q/2)}A(d,l,s)\bar{t}(d,l,Q) \\ &= 2\frac{\log N}{\log(q/2)}A(d,l,s)\bar{t}(d,l,Q) \\ &\le 4\frac{\log N}{\log q}A(d,l,s)\bar{t}(d,l,Q). \end{aligned}$$

The last inequality holds because $q \ge (s+1)^2 \ge 4$.

To show that the matrix is list-separable, we describe the (very natural) decoding algorithm. We run the naive decoding algorithm for each position $i \in [n]$ of the outer code, which gives a list of columns of the inner code. Naturally the column list corresponds to a set S_i of size at most $d + l$. Then, we run the list-recovery algorithm for the outer code to obtain the list of at most $L = O(s^s n(d+l)/k)$ codewords. □

Corollary 2.7 (Random inner code). *For every $\epsilon > 0$, there exists an efficiently decodable $(d, (1/\epsilon)^{O(1/\epsilon)}d^{1+\epsilon})$-list-disjunct matrix with N columns and $t = O\left(\frac{1}{\epsilon^2}d^{1+\epsilon}\log N\right)$ rows.*

Proof. We use the random inner code from Theorem 1.9 in Lemma 2.6, and set $s = 1/\epsilon$. □

3 Recursive constructions of efficiently decodable list-disjunct matrices

We have seen in Lemma 2.6 a procedure converting a list-disjunct matrix into an efficiently decodable one. This section describe another method of converting a family of list-disjunct matrices into ones which are efficiently decodable. There will be a slight loss in the number of tests.

3.1 Main idea

The main idea behind the second conversion procedure is as follows. Say we are trying to construct a (d, ℓ)-list-disjunct matrix $\mathbf{M}^*$ with N columns that is efficiently decodable from a family of matrices, where for any $k \ge d$, there is a (d, ℓ)-list-disjunct $t(k) \times k$ matrix $\mathbf{M}_k$ (that is not necessarily efficiently decodable). For example, if we are not concerned about the construction time then the randomized construction from Theorem 1.9 gives such a family of list-disjunct matrices. If we were also content with a linear time decoding algorithm, then we could just use $\mathbf{M}^* = \mathbf{M}_N$ and apply the naive decoder. However, we want efficient decoding. Towards this end, say we somehow knew that all the positive items are contained in a subset $\mathcal{S} \subseteq [N]$. Then the naive decoder would run in time $O(t(N) \cdot |\mathcal{S}|)$, which would be sublinear if $|\mathcal{S}|$ and $t(n)$ are sufficiently small. Of course, the trick is in getting our hands on $\mathcal{S}$. The main idea is to construct this small set $\mathcal{S}$ recursively.

Fix $N \ge d \ge 1$. Assume there exists a (d, ℓ)-list disjunct $t_1 \times \sqrt{N}$ matrix $\mathbf{M}^{(1)}$ that is efficiently decodable and let $\mathbf{M}^{(2)}$ be a (d, ℓ)-list disjunct $t_2 \times N$ matrix (that is not necessarily efficiently decodable). Let $\mathbf{M}^L$ be the $t_1 \times N$ matrix where the ith column (for $i \in [N]$) is identical to the jth column of $\mathbf{M}^{(1)}$

such that the *first* $\frac{1}{2}\log N$ bits of i is j (where we think of i and j as their respective binary representations). Similarly, let $\mathbf{M}^R$ be the $t_1 \times N$ matrix where the *last* $\frac{1}{2}\log N$ bits of i is j.

Let $S \subseteq [N], |S| \leq d$, be an arbitrary set of positives. Let the vector $\mathbf{r}^L$ ($\mathbf{r}^R$, resp.) be the vector that results from applying $\mathbf{M}^L$ ($\mathbf{M}^R$, resp.) on S. Apply the decoding algorithm for $\mathbf{M}^{(1)}$ to $\mathbf{r}^L$ ($\mathbf{r}^R$, resp.) and obtain the set S_L (S_R, resp.) of $\frac{1}{2}\log N$-bit vectors such that, for every $i \in S$, the first (last, resp.) $\frac{1}{2}\log N$ bits of i belongs to S_L (S_R, resp.). In other words, $\mathcal{S} = S_L \times S_R$ contains all the indices $i \in S$. Further, note that both $|S_L|$ and $|S_R|$ have less than $d + \ell$ elements.

Now, our final matrix $\mathbf{M}^*$ is simple: just vertically stack $\mathbf{M}^L$, $\mathbf{M}^R$ and $\mathbf{M}^{(2)}$ together. Note that $\mathbf{M}^*$ is (d, ℓ)-list disjunct because $\mathbf{M}^{(2)}$ is (d, ℓ)-list disjunct. Finally, decoding $\mathbf{M}^*$ can be done efficiently: first decode the part of the result matrix corresponding to $\mathbf{M}^L$ and $\mathbf{M}^R$ to obtain S_L and S_R respectively – this is efficient as $\mathbf{M}^{(1)}$ is efficiently decodable. Finally computing the output item set (containing S) can be done with an additional $O(t_2 \cdot (d+\ell)^2)$-time as we only need to run the naive decoder for $\mathbf{M}^{(2)}$ over $\mathcal{S} = S_L \times S_R$. To achieve a tradeoff between the number of tests and the decoding time, we choose the parameters of the recursion more carefully.

3.2 Technical details

Unlike the code concatenation construction, the second procedure gives a more general tradeoff between the blow-up in the number of tests and the resulting decoding time. On the other hand, this conversion uses multiple matrices from a given family of error-tolerant matrices unlike the previous procedure, which only used one error-tolerant list disjunct matrix.

Lemma 3.1 (Black-box conversion using recursion)**.** *Let $N \geq d \geq 1$ be integers. Assume for every $k \geq d + l$, there is a (d, ℓ)-list disjunct $t(k) \times k$ matrix $\mathbf{M}_k$ for integers $1 \leq \ell \leq N - d$. Let $1 \leq a \leq \log N$ and $1 \leq b \leq \frac{\log N}{a}$ be integers. Then there exists a $t_{a,b} \times N$ matrix $\mathbf{M}_{a,b}$ that is (d, ℓ)-list disjunct and that can be decoded in time $D_{a,b}$ where*

$$t_{a,b} = \sum_{j=0}^{\left\lceil \log_b\left(\frac{\log N}{a}\right)\right\rceil - 1} b^j \cdot t\left(\sqrt[b^j]{N}\right) \tag{5}$$

and

$$D_{a,b} = O\left(t_{a,b} \cdot \left(\frac{\log N \cdot 2^a}{a} + (d+\ell)^b\right)\right). \tag{6}$$

Finally, if the family of matrices $\{\mathbf{M}_k\}_{k \geq d}$ is (strongly) explicit then so is $\mathbf{M}_{a,b}$.

Proof. We will construct the final matrix $\mathbf{M}_{a,b}$ recursively. In particular, let such a matrix in the recursion with m columns be denoted by $\mathbf{M}_{a,b}(m)$. Note that the final matrix is $\mathbf{M}_{a,b} = \mathbf{M}_{a,b}(N)$. (For notational convenience, we will define $D_{a,b}(m)$ and $t_{a,b}(m)$ to be the decoding time for and the number of rows in $\mathbf{M}_{a,b}(m)$ respectively). Next, we define the recursion.

If $m \leq 2^a$, then set $\mathbf{M}_{a,b}(m) = \mathbf{M}_m$. Note that in this case, $t_{a,b}(m) = t(m)$. Further, we will use the naive decoder in the base case, which implies that $D_{a,b}(m) = O(t_{a,b}(m) \cdot m) \leq O(2^a \cdot t_{a,b}(m))$. It is easy to check that both (5) and (6) are satisfied. Finally because $\mathbf{M}_m$ is a (d, ℓ)-list disjunct matrix, so is $\mathbf{M}_{a,b}(m)$.

Now consider the case when $m > 2^a$. For $i \in [b]$, define $\mathbf{M}^{(i)}$ to be the $t_{a,b}(\sqrt[b]{m}) \times m$ matrix whose jth column (for $j \in [m]$) is identical to the kth column in $\mathbf{M}_{a,b}(\sqrt[b]{m})$ where k is the ith chunk of $\frac{1}{b}\log m$ bits in j (we think of j and k as their respective binary representations). Define $\mathbf{M}_{a,b}(m)$ to be the stacking of

$\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots, \mathbf{M}^{(b)}$ and $\mathbf{M}_m$. Since $\mathbf{M}_m$ is a (d, ℓ)-list disjunct matrix, so is $\mathbf{M}_{a,b}(m)$. Next, we verify that (5) holds. To this end note that

$$t_{a,b}(m) = b \cdot t_{a,b}(\sqrt[b]{m}) + t(m). \tag{7}$$

In particular, (by induction) all the $\mathbf{M}^{(i)}$ contribute

$$b \cdot \sum_{j=0}^{\left\lceil \log_b \left(\frac{\log \sqrt[b]{m}}{a} \right) \right\rceil - 1} b^j \cdot t\left(\sqrt[b^j]{\sqrt[b]{m}} \right) = \sum_{j=1}^{\lceil \log_b (\frac{\log m}{a}) \rceil - 1} b^j \cdot t\left(\sqrt[b^j]{m} \right)$$

rows. Since $\mathbf{M}_m$ adds another $t(m)$ rows, $\mathbf{M}_{a,b}(m)$ indeed satisfies (5).

Finally, we consider the decoding of $\mathbf{M}_{a,b}(m)$. The decoding algorithm is natural: we run the decoding algorithm for $\mathbf{M}_{a,b}(\sqrt[b]{m})$ (that is guaranteed by induction) on the part of the outcome vector corresponding to each of the $\mathbf{M}^{(i)}$ ($i \in [b]$) to compute sets S_i with the following guarantee: each of the at most d defective indices $k \in [m]$ projected to the ith chunk of $\frac{1}{b} \log m$ is contained in S_i. Finally, we run the naive decoding algorithm for $\mathbf{M}_m$ on $\mathcal{S} \stackrel{def}{=} S_1 \times S_2 \times \cdot \times S_b$ (note that by definition of S_i all of the defective items will be in $\mathcal{S}$). To complete the proof, we need to verify that this algorithm takes time as claimed in (6). Note that

$$D_{a,b}(m) = b \cdot D_{a,b}(\sqrt[b]{m}) + O\left(|\mathcal{S}| \cdot t(m)\right).$$

By induction, we have

$$D_{a,b}(\sqrt[b]{m}) = O\left(t_{a,b}(\sqrt[b]{m}) \cdot \left(\frac{\log m \cdot 2^a}{a} + (d+\ell)^b \right) \right),$$

and since $\mathbf{M}_{a,b}(\sqrt[b]{m})$ is a (d, ℓ)-list disjunct matrix, we have

$$|\mathcal{S}| \leq (d+\ell)^b.$$

The three relations above along with (7) show that $D_{a,b}(m)$ satisfies (6), as desired.

Finally, the claim on explicitness follows from the construction. $\square$

The bound in (5) is somewhat unwieldy. We note in Corollary 3.2 that when $t(i) = d^x \log^y i$ for some reals $x, y \geq 1$, we can achieve efficient decoding with only a log-log factor increase in number of tests. We will primarily use this result in our applications.

Note that in this case the bound in (5) can be bounded as

$$\sum_{j=0}^{\lceil \log_b (\frac{\log N}{a}) \rceil - 1} b^j \cdot d^x \cdot \left(\frac{\log N}{b^j} \right)^y \leq \left\lceil \log_b \left(\frac{\log N}{a} \right) \right\rceil \cdot d^x \log^y N.$$

Lemma 3.1 with $b = 2$ and $a = \log d$, along with the observation above, implies the following:

Corollary 3.2. *Let $N \geq d \geq 1$ be integers and $x, y \geq 1$ be reals. Assume for every $k \geq d$, there is a (d, ℓ)-list disjunct $O(d^x \log^y k) \times k$ matrix for integers $1 \leq \ell \leq N - d$. Then there exists a $t \times N$ matrix that is (d, ℓ)-list disjunct that can be decoded in* $\text{poly}(t, \ell)$ *time, where*

$$t \leq O\left(d^x \cdot \log^y N \cdot \log \log_d N \right).$$

Finally, if the original matrices are (strongly) explicit then so is the new one.

In other words, the above result implies that we can achieve efficient decoding with only a log-log factor increase in number of tests. We will primarily use the above result in our applications.

To show the versatility of Lemma 3.1 we present another instantiation. For any $0 \le \epsilon \le 1$, if we pick $a = b\log(d+\ell)$ and $b = (\log N/a)^{\epsilon}$, we get the following result:

Corollary 3.3. *Let $N \ge d \ge 1$ be integers and $x, y \ge 1$, $0 < \epsilon \le 1$ be reals. Assume for every $k \ge d$, there is a (d,ℓ)-list-disjunct $O(d^x \log^y k) \times k$ matrix for integers $1 \le \ell \le N-d$. Then there exists a $t \times N$ matrix that is (d,ℓ)-list-disjunct that can be decoded in*

$$\mathrm{poly}(t,\ell) \cdot 2^{\sqrt[1+\epsilon]{\log^{\epsilon} N \cdot \log(d+\ell)}}$$

time, where

$$t \le O\left(\frac{1}{\epsilon} \cdot d^x \log^y N\right).$$

Finally, if the original matrices are (strongly) explicit then so is the new one.

Note that the above result implies that with only a constant factor blow-up in the number of tests, one can perform sub-linear in N time decoding when $(d+\ell)$ is polynomially small in N.

4 Strongly explicit constructions of list-disjunct matrices

4.1 Construction using expanders

A W-left regular bipartite graph $[N] \times [W] \to [T]$ 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 $(1-\epsilon)|S|W$. Given such a bipartite expander G, consider the $T \times N$ incidence matrix M_G of G. We have the following simple observation.

Proposition 4.1. *Let G be a $(n, w, t, 2d, w/2+1)$-expander. Then M_G is a (d,d)-list disjunct matrix.*

Proof. Recall that by definition a matrix M is (d,d)-list disjunct if the following is true: for every two disjoint S_1 and S_2 subsets of columns of size exactly d, both $\cup_{i\in S_1} M_i \not\subseteq cup_{j\in S_2} M_j$ and $\cup_{j\in S_2} M_j \not\subseteq \cup_{i\in S_1} M_i$ hold. Note that this property for M_G translates to the following for G: $\Gamma_G(S_1) \not\subseteq \Gamma_G(S_2)$ and vice-versa. We now argue that the latter is true if G has an expansion of $w/2+1$. Indeed this follows from the facts that $|\Gamma_G(S_1 \cup S_2)| \ge wd + 2d$ and $|\Gamma_G(S_1)|, |\Gamma_G(S_2)| \le wd$. □

To construct an *explicit* expander for our purposes, we will use the following two constructions.

Theorem 4.2 ([15]). *Let $\epsilon > 0$. There exists an* explicit *$(N_1, W_1, T_1, D_1, W_1(1-\epsilon))$ expander with $T_1 \le (4D_1)^{\log W_1}$ and $W_1 \le 2\log N_1 \log D_1/\epsilon$.*

Theorem 4.3 ([25]). *Let $\epsilon > 0$ be a* constant*. Then there exists an* explicit *$(N_2, W_2, T_2, D_2, W_2(1-\epsilon))$-expander with $T_2 = O(D_2 W_2)$ and $W_2 = 2^{O(\log\log N_2 + (\log\log D_2)^3)}$.*

We will combine the above two expanders using the following well known technique.

Proposition 4.4. *Let G_1 be an $(N, W_1, T_1, D, W_1(1-\epsilon))$-expander and G_2 be an $(T_1, W_2, T_2, DW_1, W_2(1-\epsilon))$-expander. Then there exists an $(N, W_1W_2, T_2, D, W_1W_2(1-2\epsilon))$-expander G. Further, if G_1 and G_2 are explicit then so is G.*

Proof. The graph G is constructed by "concatenating" G_1 and G_2. In particular, construct the following intermediate tripartite graph G' (on the vertex sets $[N]$, $[T_1]$ and $[T_2]$ respectively), where one identifies $[T_1]$ once as the right vertex set for G_1 and once as the left vertex set of G_2. The final graph G is bipartite graph on $([N], [T_2])$ where there is an edge if and only if there is a corresponding path of length 2 in G'. It is easy to check that G is an $(N, W_1W_2, T_2, D, W_1W_2(1-2\epsilon))$-expander. $\square$

Next, we prove the following result by combining all the ingredients above.

Theorem 4.5. *There exists an explicit $(N, W, T, D, W/2+1)$-expander with $T = O(D \log N \cdot f(D, N))$, where*

$$f(D, N) = 2^{O((\log\log D)^3 + (\log\log\log N)^3)}.$$

Note that $f(D, N) = (D \log N)^{o(1)}$.

Proof. By Theorem 4.2, there exists an explicit $(N, W_1, T_1, D, 3W_1/4+1)$-expander, where

$$W_1 \le 16 \log N \log D,$$

and

$$T_1 \le (4D)^{4+\log\log N + \log\log D}.$$

By Theorem 4.3, there exists an explicit $(T_1, W_2, T_2, D_2, 3W_2/4+1)$-expander, where

$$D_2 = DW_1 \le 16D \log N \log D,$$

W_2 is $2^{O(\log\log T_1 + (\log\log D_2)^3)}$, which in turn is at most $2^{O((\log\log D)^3 + (\log\log\log N)^3)}$, and T_2 is

$$\begin{aligned} W_2 D_2 &\le 16D \log N \log D \cdot 2^{O((\log\log D)^3 + (\log\log\log N)^3)} \\ &\le D \log N \cdot f(D, N), \end{aligned}$$

as desired. $\square$

Theorem 4.5 and Proposition 4.1 leads to the following construction.

Theorem 4.6 ([17]). *Let $1 \le d \le N$ be integers and $\delta > 0$ be any given constant. Then there exists a strongly-explicit $t \times N$ matrix that is $(d, \delta d)$-list disjunct with $t = O((d \log N)^{1+o(1)})$ rows.*

Combining Lemma 2.6 and Theorem 4.6, we get the following result.

Corollary 4.7. *Let $\epsilon > 0$ be a real number and let $1 \le d \le N$ be integers. Then there exists a strongly-explicit $t \times N$ matrix that is $(d, (1/\epsilon)^{O(1/\epsilon)} \cdot d^{1+\epsilon})$-list-disjunct with $t = (1/\epsilon)^{O(1/\epsilon)} \cdot d^{1+\epsilon} \cdot (\log N)^{1+o(1)}$ rows that can be decoded in time $t^{O(1/\epsilon)}$.*

Next, we instantiate Corollary 3.2 to obtain efficiently decodable list disjunct matrices that will be used to construct strongly explicit disjunct matrices and imply other applications. In particular, Corollary 3.2 along with Theorem 4.6 implies the following:

Corollary 4.8. *Let $1 \le d \le N$ be integers. For any constant $\delta > 0$ there exists a strongly-explicit $t \times N$ matrix that is $(d, \delta d)$-list-disjunct with $t = O((d \log N)^{1+o(1)})$ rows and can be decoded in* $\text{poly}(t)$ *time.*

By stacking an explicit $(d, O(d))$-list-disjunct matrix on top of a d-disjunct matrix (say, the one from [21]), we obtain the following result.

Theorem 4.9 (Explicit construction of efficiently decodable disjunct matrices). *Let $1 \le d \le N$ be integers. Then there exists a $t \times N$ d-disjunct matrix with $t = O(d^2 \log N)$ that can be decoded in* $\text{poly}(t)$*-time. Further, the matrix can be computed in time $\tilde{O}(Nt)$.*

4.2 Construction using extractors

This construction is from [6]. The number of tests is better than that in Theorem 4.6, however we cannot precisely control the list size as in Theorem 4.6. Some applications requires the precision in list size bound. To describe the construction, we need a few notions.

Randomness extractors are functions which "convert" biased and correlated random bits into almost uniform random bits. Extractors have numerous applications in (theoretical) Computer Science[3]. In this section, we will use extractors to construct good list-disjunct matrices.

Let $\mathcal{D}$ be a distribution on a finite sample space Ω. The *min entropy* of $\mathcal{D}$ is defined to be

$$H_\infty(\mathcal{D}) := \min_{\omega\in\Omega}\left\{\frac{1}{\log_2 \text{Prob}_{\mathcal{D}}(\omega)}\right\}.$$

Here, $\text{Prob}_{\mathcal{D}}(\omega)$ is the probability mass the distribution assigns to ω. If $H_\infty(\mathcal{D}) \geq k$, then $\text{Prob}_{\mathcal{D}}(\omega) \leq 1/2^k$ for every $\omega \in \Omega$.

The *total variational distance* between two distributions $\mathcal{P}$ and $\mathcal{Q}$ on Ω is

$$\|\mathcal{P} - \mathcal{Q}\|_{\text{TV}} = \max_{A\subseteq\Omega} |\mathcal{P}(A) - \mathcal{Q}(A)| = \frac{1}{2}\sum_{\omega\in\Omega} |\mathcal{P}(\omega) - \mathcal{Q}(\omega)|.$$

The second inequality can easily be proved from first principles. Two distributions are *ϵ-close* if their variational distance is at most ϵ. Let $\mathcal{U}_n$ denote the uniform distribution on $\mathbb{F}_2^n$.

A function $C : \mathbb{F}_2^a \times \mathbb{F}_2^b \to \mathbb{F}_2^m$ is called a *strong $k \to_\epsilon k'$ condenser* if it satisfies the following: for every distribution $\mathcal{A}$ on $\mathbb{F}_2^a$ with $H_\infty(\mathcal{A}) \geq k$, random variable $A \leftarrow \mathcal{A}$, *seed* variable $B \leftarrow \mathcal{U}_b$, the distribution of $(B, C(A,B))$ is ϵ-close to some distribution $(\mathcal{U}_b, \mathcal{Z})$ with min entropy at least $b + k'$. Here, ϵ is called the *error*, $k - k'$ is called the *entropy loss*, and $m - k'$ is called the *overhead* of the condenser. A *lossless condenser* is a condenser with no entropy loss. A *strong (k,ϵ)-extractor* is a a condenser with no overhead.

From a function $C : \mathbb{F}_2^a \times \mathbb{F}_2^b \to \mathbb{F}_2^m$ we can defined the corresponding *induced code* $\mathcal{I}(C)$ as follows. This code has alphabet $\Sigma = \mathbb{F}_2^m$, length $n = 2^b$, and size $N = 2^a$. For $A \in \mathbb{F}_2^a$, the Ath codeword of the code is defined to be the vector whose Bth component is $C(A,B)$, where the components of the codewords are indexed by $B \in \mathbb{F}_2^b$.

For any finite alphabet Σ and a positive integer n. a sequence $S = (S_1, \cdots, S_n)$ where $\emptyset \neq S_i \subseteq \Sigma$ is called a *mixture* on Σ^n. Define

$$\rho(S) = \frac{|S_1| + \cdots + |S_n|}{n|\Sigma|}.$$

For any word $w \in \Sigma^n$, the *agreement* of w with S is defined to be

$$\text{Agr}(w,S) := \frac{|\{i \in [n] \mid w_i \in S_i\}|}{n}.$$

It is not hard to see that $\rho(S)$ is the expected agreement of a randomly chosen word in Σ^n with S. Consider a code $C \subseteq \Sigma^n$ and $\alpha \in [0,1]$. The list of codewords whose agreement with S is $> \alpha$ is denoted by $\text{LIST}_C(S,\alpha)$. When $\alpha = 1$, the list consists of codewords with 100% agreement.

The following important theorem was first observed in [] (see also [15]).

Theorem 4.10. *Let $C : \mathbb{F}_2^a \times \mathbb{F}_2^b \to \mathbb{F}_2^m$ be a $k \to_\epsilon k'$ condenser. For any mixture S on $(\mathbb{F}_2^m)^{2^b}$, if $\rho(S)2^{m-k'} + \epsilon < 1$ then*

$$\text{LIST}_{\mathcal{I}(C)}(S, \rho(S)2^{m-k'} + \epsilon) < 2^k.$$

[3]http://people.seas.harvard.edu/ salil/pseudorandomness/extractors.pdf

Proof. Assume to the contrary that $\textsc{list}_{\mathcal{I}(C)}(S,\rho(S)2^{m-k'}+\epsilon) \geq 2^k$. Let $A = (A_1,\dots,A_{2^b})$ be a random codeword uniformly chosen from $\textsc{list}_{\mathcal{I}(C)}(S,\rho(S)2^{m-k'}+\epsilon)$. Then, as a distribution on $\mathbb{F}_2^a$ the random variable A has min entropy at least k. Let $B \leftarrow \mathbb{U}_b$ be a uniformly random variable chosen from $\mathbb{F}_2^b$. Then, by definition $(B, C(A,B)) = (B, A_B)$ is supposed to be ϵ-close to some distribution on $\mathbb{F}_2^{b+m}$ with min entropy at least $b+k'$. Let $\mathcal{D}$ be any distribution on $\mathbb{F}_2^{b+m}$ with min entropy at least $b+k'$. We will show that $\mathcal{D}$ and the distribution of (B,A_B) are not ϵ-close by specifying an event f on $\mathbb{F}_2^{b+m}$ for which the two distributions differ by more than ϵ.

The event f is defined by a function $f : \mathbb{F}_2^{b+m} \to \{0,1\}$, where for $i \in \mathbb{F}_2^b$ and $X \in \mathbb{F}_2^m$, we define $f(i,X) = 1$ iff $X \in S_i$.

Next, let us estimate the probabilities that the distribution of (B,A_B) and the distribution $\mathcal{D}$ assign to the event f.

First, consider the random point (B,A_B):

$$\operatorname*{Prob}_{A,B}[f(B,A_B)=1] = \operatorname*{Prob}_{A,B}[A_B \in S_B] = \mathrm{Agr}(A,S) > \rho(S)2^{m-k'}+\epsilon.$$

Next, consider the distribution $\mathcal{D}$. Let $(i,X) \in \mathbb{F}_2^b \times F_2^m$ be drawn according to $\mathcal{D}$. Then,

$$\begin{aligned}\operatorname*{Prob}_{\mathcal{D}}[f(i,X)=1] &= \sum_{(i,X)\in\mathbb{F}_2^b\times F_2^m} \mathbf{1}_{X\in S_i} \operatorname*{Prob}_{\mathcal{D}}[(i,X)] \\ &\leq 2^{-b-k'} \sum_{i\in\mathbb{F}_2^b} |S_i| \\ &= \rho(S)2^{m-k'}.\end{aligned}$$

□

From the theorem, Cheraghchi [6] observed the following, which was the main result in that paper. The language that Cheraghchi used was not code concatenation, and he did not used the term *list-separable*, but we can easily see the analogy.

Corollary 4.11 ([6]). *Let $C : \mathbb{F}_2^a \times \mathbb{F}_2^b \to \mathbb{F}_2^m$ be a $k \to_\epsilon k'$ condenser. Then, the concatenation $\mathcal{I}(C) \circ ID_{2^m}$ is a $(d, 2^k)$-list-separable matrix for any d satisfying the following constraints: $d \leq 2^m$, $d < (1-\epsilon)2^{k'}$. Furthermore, the total decoding time is $O(2^{a+b+m})$.*

Proof. We simply specify a decoding algorithm. We decode a set S_i for each position $i \in \mathbb{F}_2^b$. Note that $|S_i| \leq d$ for each i because there are at most d positives. Thus, $\rho(S) = \sum_i |S_i|/2^{b+m} \leq d/2^m$. From Theorem 4.10 we know $\textsc{list}_{\mathcal{I}(C)}(S, d/2^m+\epsilon) < 2^k$. Furthermore, all positive items correspond to codewords with 100% agreement with S. Hence, we can simply output all such codewords. The running time is $O(2^{a+b+m})$, and the number of codewords outputted is at most 2^k. □

Next, we apply a construction from [15].

Theorem 4.12 ([15]). *For integers $a \geq k$, and $\epsilon > 0$, there exists an explicit strong (k,ϵ)-extractor $Ext : \mathbb{F}_2^a \times \mathbb{F}_2^b \to \mathbb{F}_2^m$ with $m = k - 2\log(1/\epsilon) - O(1)$ and $b = \log a + O(\log k \cdot \log(k/\epsilon))$.*

The result along with Corollary 4.11 leads to the following.

Theorem 4.13 ([6]). *Let $1 \leq d \leq N$ be integers. Then there exists a strongly-explicit $t \times N$ matrix that is $(d, O(d))$-list disjunct with $t = O(d^{1+o(1)}\log N)$ rows.*

Combining Lemma 2.6 and Theorem 4.13, we get the following result.

Corollary 4.14. *Let $\epsilon > 0$ be a real number and let $1 \le d \le N$ be integers. Then there exists a strongly-explicit $t \times N$ matrix that is $(d, (1/\epsilon)^{O(1/\epsilon)} \cdot d^{1+\epsilon})$-list-disjunct with $t = (1/\epsilon)^{O(1/\epsilon)} \cdot d^{1+\epsilon} \cdot \log N$ rows that can be decoded in time $t^{O(1/\epsilon)}$.*

Corollary 3.2 along with Theorem 4.13 implies the following:

Corollary 4.15. *Let $1 \le d \le N$ be integers. For any constant $\alpha \in (0, 1)$ there exists a strongly-explicit $t \times N$ matrix that is $(d, O(d))$-list disjunct with $t = O(d^{1+o(1)} \log N \log \log N)$ rows and can be decoded in $\mathrm{poly}(t)$ time.*

References

[1] N. Alon and V. Asodi, *Learning a hidden subgraph*, SIAM J. Discrete Math., 18 (2005), pp. 697–712 (electronic).

[2] N. Alon and R. Hod, *Optimal monotone encodings*, in ICALP '08: Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part I, Berlin, Heidelberg, 2008, Springer-Verlag, pp. 258–270.

[3] N. Alon, D. Moshkovitz, and S. Safra, *Algorithmic construction of sets for k-restrictions*, ACM Trans. Algorithms, 2 (2006), pp. 153–177.

[4] B. Bollobás, *Combinatorics*, Cambridge University Press, Cambridge, 1986. Set systems, hypergraphs, families of vectors and combinatorial probability.

[5] E. J. Candès and T. Tao, *Near-optimal signal recovery from random projections: Universal encoding strategies?*, IEEE Transactions on Information Theory, 52 (2006), pp. 5406–5425.

[6] M. Cheraghchi, *Noise-resilient group testing: Limitations and constructions*, in FCT, 2009, pp. 62–73.

[7] A. De Bonis, L. Gąsieniec, and U. Vaccaro, *Optimal two-stage algorithms for group testing problems*, SIAM J. Comput., 34 (2005), pp. 1253–1270 (electronic).

[8] D. L. Donoho, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2006), pp. 1289–1306.

[9] A. G. D'yachkov and V. V. Rykov, *A survey of superimposed code theory*, Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform., 12 (1983), pp. 229–242.

[10] P. Erdős, P. Frankl, and Z. Füredi, *Families of finite sets in which no set is covered by the union of r others*, Israel J. Math., 51 (1985), pp. 79–89.

[11] Z. Füredi, *On r-cover-free families*, J. Combin. Theory Ser. A, 73 (1996), pp. 172–173.

[12] S. Ganguly, *Data stream algorithms via expander graphs*, in 19th International Symposium on Algorithms and Computation (ISAAC), 2008, pp. 52–63.

[13] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss, *Approximate sparse recovery: optimizing time and measurements*, in STOC, 2010, pp. 475–484.

[14] V. Guruswami and A. Rudra, *Soft decoding, dual BCH codes, and better list-decodable ϵ-biased codes*, in Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC), 2008, pp. 163–174.

[15] V. Guruswami, C. Umans, and S. P. Vadhan, *Unbalanced expanders and randomness extractors from parvaresh-vardy codes*, in Proceedings of the 22nd Annual IEEE Conference on Computational Complexity, 2007, pp. 96–108.

[16] P. Indyk, *Explicit constructions of selectors and related combinatorial structures, with applications*, in SODA, 2002, pp. 697–704.

[17] P. Indyk, H. Q. Ngo, and A. Rudra, *Efficiently decodable non-adaptive group testing*, in Proceedings of the Twenty First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2010), New York, 2010, ACM, pp. 1126–1142.

[18] T. Moran, M. Naor, and G. Segev, *Deterministic history-independent strategies for storing information on write-once memories*, Theory of Computing, 5 (2009), pp. 43–67.

[19] H. Q. Ngo, E. Porat, and A. Rudra, *Efficiently decodable error-correcting list disjunct matrices and applications*, in ICALP, 2011.

[20] F. Parvaresh and A. Vardy, *Correcting errors beyond the guruswami-sudan radius in polynomial time*, in Proceedings of the 46th Annual IEEE Symposium on Foundations of Compu ter Science (FOCS), 2005, pp. 285–294.

[21] E. Porat and A. Rothschild, *Explicit non-adaptive combinatorial group testing schemes*, in ICALP (1), 2008, pp. 748–759.

[22] A. M. Rashad, *Random coding bounds on the rate for list-decoding superimposed codes*, Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform., 19 (1990), pp. 141–149.

[23] A. Rudra and S. Uurtamo, *Data stream algorithms for codeword testing*, in Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP), 2010. To appear.

[24] M. Ruszinkó, *On the upper bound of the size of the r-cover-free families*, J. Combin. Theory Ser. A, 66 (1994), pp. 302–310.

[25] A. Ta-Shma, C. Umans, and D. Zuckerman, *Lossless condensers, unbalanced expanders, and extractors*, Combinatorica, 27 (2007), pp. 213–240.