Optimal Group Testing using Left and Right regular sparse-graph codes

Avinash Vem, Nagaraj T. Janakiraman, Krishna R. Narayanan
Department of Electrical and Computer Engineering
Texas A&M University
{vemavinash@gmail.com, tjnagaraj@tamu.edu, krn@tamu.edu}

***Abstract*—To be added.**

I. Introduction

The problem of Group Testing refers to testing a large population for sick/defective individual people when the fraction of sick people is known to be small. This problem was first introduced to the literature of statistics by Dorfman [1] during World War II for testing the soldiers for syphilis without having to test each soldier individually. Since then many schemes and algorithms were designed for this problem.

In [2] Lee, Pedarsani, Ramchandran applied the sparse-graph codes and a simple peeling decoder, which are popular tools in the error control coding community, to the non-adaptive group testing problem. [3], [4], [2]

II. Problem Statement

Formally, we define the group testing problem as follows. Let N be the total number of items, K be the number of defective items and m be the number of tests for identifying the defective items. For now we consider only the noiseless group testing problem i.e., the result of each test is exactly equal to the boolean OR of all the items participating in the test.

Formally, let the support vector $\mathbf{x} \in \{0,1\}^N$ denote the list of items with non-zero indices indicating the defective items. A non-adaptive testing scheme of m measurements can be denoted by a matrix $\mathbf{A} \in \{0,1\}^{m \times N}$ where each row $\mathbf{a}_i$ corresponds to a test. The non-zero indices in row $\mathbf{a}_i$ correspond to the items that participate in i^{th} test. With these notations the output corresponding to $\mathbf{A}$ can be expressed in matrix form as:

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x}$$

where $\odot$ is the usual matrix multiplication in which the arithmetic multiplications are replaced by the boolean AND operation and the arithmetic additions are replaced by the boolean OR operation.

III. Review: Prior Work

In [2] Lee, Pedarsani and Ramchandran introduced a framework based on left-regular sparse graph codes (referred to as SAFFRON) for non-adaptive group testing problem. We will briefly review their SAFFRON testing scheme, decoder and their main results. The SAFFRON testing scheme consists of two stages: the first stage is based on a left-regular sparse graph code which groups the variable nodes into non-disjoint M_1 bins where each variable node belongs to exactly l bins. The second stage comprises of producing h testing outputs at each bin where the h different combinations of the pooled variables at the respective bin(from the first stage) are defined according to a universal signature matrix. For the first stage consider a bipartite graph with N variable nodes and M_1 bin nodes. Each variable node is connected to l bin nodes chosen uniformly at random from the M_1 available bin nodes. All the variable nodes (historically depicted on the left side of the graph) have a degree l, hence the left-regular, whereas the degree of a bin node on the right is a random variable ranging from $[1:n]$.

Definition 1 (Left-regular sparse graph ensemble)**.** We define $\mathcal{G}_l(N, M_1)$ to be the ensemble of left-regular graphs where, for each variable node, the l right node connections are chosen uniformly at random from the M_1 right nodes.

Let $\mathbf{T}_G \in \{0,1\}^{M_1 \times N}$ be the adjacency matrix corresponding to a graph $G \in \mathcal{G}_l(N, M_1)$ i.e., each column in $\mathbf{T}_G$ corresponding to a variable node has exactly l ones. And let us denote the universal signature matrix defining the h tests at each bin by $\mathbf{U} \in \{0,1\}^{h \times N}$. Thus the total number of tests is $M = M_1 \times h$. More formally, the overall testing matrix $\mathbf{A} := [\mathbf{A}_1^T, \ldots, \mathbf{A}_{M_1}^T]^T$ where $\mathbf{A}_i = \mathbf{U} \circ \text{diag}(\mathbf{t}_i)$ defines the h tests at i^{th} bin.

The signature matrix $\mathbf{U}$ in a more general setting with a free parameter r can be given by

$$\mathbf{U_r} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_r \\ \overline{\mathbf{b}}_1 & \overline{\mathbf{b}}_2 & \cdots & \overline{\mathbf{b}}_r \\ \mathbf{b}_{i_1} & \mathbf{b}_{i_2} & \cdots & \mathbf{b}_{i_r} \\ \overline{\mathbf{b}}_{i_1} & \overline{\mathbf{b}}_{i_2} & \cdots & \overline{\mathbf{b}}_{i_r} \\ \mathbf{b}_{j_1} & \mathbf{b}_{j_2} & \cdots & \mathbf{b}_{j_r} \\ \overline{\mathbf{b}}_{j_1} & \overline{\mathbf{b}}_{j_2} & \cdots & \overline{\mathbf{b}}_{j_r} \end{bmatrix} \tag{1}$$

where $\mathbf{b}_i \in \{0,1\}^{\lceil \log_2 r \rceil}$ is the binary expansion vector for i and $\overline{\mathbf{b}}_i$ is the complement of $\mathbf{b}_i$. $i_1, i_2, \ldots, i_r$ and $j_1, j_2, \ldots, j_r$ are two random permutations of $[1:r]$. For the SAFFRON

scheme the parameter is chosen to be $r = N$ thus resulting in $\mathbf{U}$ of size $h \times N$ where $h \approx 6 \log_2 N$.

Now we can define the ensemble of left-regular sparse-graph code based group testing matrices described by the SAFFRON scheme via the ensembles $\mathcal{G}_l(N, M_1)$ and $\mathbf{U}_N$.

Decoding

Before describing the decoding process let us review some terminology. We refer to a bin as a singleton if there is exactly one non-zero variable node connected to the bin. We refer to a bin as a resolvable double-ton if there are exactly two non-zero variable nodes connected to the bin and we know the identity of one of them leaving to the decoder to decode the identity of the other one. And if the bin has more than two non-zero variable nodes attached we refer to it as a multi-ton. First part of the decoder which we refer to as bin decoder will be able to detect and decode the identity of the non-zero variable nodes connected in the bin exactly if the bin is a singleton or a resolvable double-ton. If the bin is a multi-ton the bin decoder will detect it as a multi-ton, i.e., the bin decoder output is not a singleton or a resolvable double-ton, with asymptotically high probability of atleast 1-$O(\frac{1}{N^2})$. For details of the decoder we refer the reader to [2]. The second part of the decoder which is commonly referred to as peeling decoder [5], using the outputs from the bin decoder, uncovers the identities of the non-zero variable nodes (referred to as peeling off from the graph historically) in an iterative manner.

The overall group testing decoder comprises of these two decoders working in conjunction as follows. In the first and foremost step, given the M tests output, we run the bin decoder on the M_1 bins and we are given the set of singletons i.e., the set of decoded non-zero variable nodes denoted as $\mathcal{D}$. Now in an iterative manner, at each iteration, we consider a variable node from $\mathcal{D}$ and apply the bin decoder on the bins connected to this variable node and hopefully one of them is a resolvable double-ton thus resulting in us decoding one more non-zero variable node. We will move the said considered variable node in the previous iteration from $\mathcal{D}$ to a set of peeled off variable nodes $\mathcal{P}$. And we will place the newly decoded non-zero variable node in the previous iteration, if any, in $\mathcal{D}$ and continue the next iteration. The deocoder terminates if $\mathcal{D}$ is empty and the decoder is declared successful if the set $\mathcal{P}$ equals the set of defective items.

Remark 2. Note that we are not literally peeling off the decoded nodes from the graph because of the *non-linear* OR operation on the non-zero variable nodes at each bin thus preventing us in subtracting the effect of the non-zero node from the measurements of the bin node unlike in the problems of compressed sensing or LDPC codes on binary erasure channel.

Now we state the series of lemmas and theorems, without proofs, from [2] that enabled the authors Lee, Pedarsani and Ramchandran to show that this left-regular sparse-graph code based scheme with the described peeling decoder solves the group testing problem with $\Omega(K \log N)$ tests and with $O(K \log N)$ computational complexity.

Lemma 3 (Bin decoder analysis)**.** For a signature matrix $\mathbf{U}_r$ as described in (1), the bin decoder successfully detects and resolves if the bin is either a singleton or a resolvable double-ton. If the bin is a multi-ton the bin decoder declares a wrong non-zero variable node with a probability no greater than $O(\frac{1}{r^2})$.

For the ease of analysis, the error probability performance analysis of the peeling decoder is done independent of the bin decoder i.e., the peeling decoder is analyzed under the assumption that the bin decoder is working accurately which will be referred to as oracle based peeling decoder. To simplify further, a pruned graph is considered where all the zero variable nodes and their respective edges are removed from the graph and a simplified version of peeling decoder which iteratively considers a variable node as decoded perfectly if it is connected to a right-node with degree one or a right-node with degree two where one of the variables connected is already decoded. Anything with more than degree two is untouched by the oracle based peeling decoder. It is easy to verify that the original decoder with accurate bin decoding is equivalent to this simplified peeling decoder on a pruned graph.

Definition 4 (Pruned graph ensemble)**.** We will define the pruned graph ensemble $\tilde{\mathcal{G}}_l(N, K, M_1)$ as the set of all graphs obtained from removing a random $N - K$ subset of variable nodes from a graph from the ensemble $\mathcal{G}_l(N, M_1)$. Note that graphs from the pruned ensemble have K variable nodes.

Before we analyze the pruned graph ensemble let us define the right-node degree distribution (d.d) as $R(x) = \sum_i R_i x^i$ where R_i is the probability that a randomly chosen right-node has degree i. Now similarly the edge d.d $\rho(x) = \sum_i \rho_i x^{i-1}$ is defined where ρ_i is the probability that a randomly chosen edge in the graph is connected to a right-node of degree i. Note that the left-degree distribution is regular even for the pruned graph case and hence we don't specifically mention it.

Lemma 5 (Edge d.d of Pruned graph)**.** For the pruned ensemble $\tilde{\mathcal{G}}_l(N, K, M_1)$, it can be shown in the limit $K, N \to \infty$ that $\rho_1 = e^{-\lambda}$ and $\rho_2 = \lambda e^{-\lambda}$ where $\lambda = \frac{l}{C}$ if $M_1 = CK$ for a constant C.

Lemma 6. For the pruned graph ensemble $\tilde{\mathcal{G}}_l(N, K, M_1)$ the oracle-based peeling decoder fails to peel off atleast $(1 - \epsilon)$ fraction of the variable nodes with exponentially decaying probability if $M_1 = C(\epsilon)K$ where $C(\epsilon)$ for various ϵ is given in Table. III.

Proof. Instead of reworking the whole proof here from [2], we will list the main steps involved in the proof which we will use further along. If we let p_j be the probability that a

ϵ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
$C(\epsilon)$	6.13	7.88	9.63	11.36	13.10	14.84	16.57
l	7	9	10	12	14	15	17

TABLE I
CONSTANTS FOR VARIOUS ERROR FLOOR VALUES

random defective item is not identified at iteration j, in the limit N and $K \to \infty$. We can write density evolution (DE) equation relating p_{j+1} to p_j as

$$p_{j+1} = \left[1 - (\rho_1 + \rho_2(1 - p_j))\right]^{l-1}.$$

For this DE, we can see that 0 is not a fixed point and hence $p_j \nrightarrow 0$ as $j \to \infty$. Therefore numerically optimizing the values of C and l such that $\lim_{j\to\infty} p_j \leq \epsilon$ gives us the optimal values for $C(\epsilon)$ and l given in Table. III. □

Combining the lemmas and remarks above, the main result from [2] can be summarized as follows.

Theorem 7. The SAFFRON framework recovers atleast a $(1-\epsilon)$ fraction of the defective items for arbitrarily-small ϵ with high probability $1 - O(\frac{K}{N^2})$. The number of tests is $m = 6C(\epsilon)K\log_2 N$ where $C(\epsilon)$ is given in Table. III and the computational complexity of the decoding is $O(K \log N)$.

Note that the computational complexity is order optimal for both the noiseless and noisy settings as mentioned in [2]. Regarding the optimality of the number of tests for the noiseless setting where both K and N scale satisfying $K = o(N)$, it was shown [3] that the number of tests need to be atleast as large as $CK\log(\frac{N}{K})$ for some constant C such that the probability of error approaches zero. As far as we are aware this is the tightest lower bound. In the same work it is shown that $CK \log N$ is the sufficient number of tests. In our work we show that in fact $C(\epsilon)K\log(\frac{N}{K})$ tests is sufficient to recover $(1-\epsilon)$ fraction of the defective items with high probability. More survey needs to be done regarding the lower and upper bounds for the number of tests in noiseless and noisy settings especially under different performance evaluation criteria. For e.g., in [3] the upper bound(achievable) on the minimal number of tests $O(K \log N)$ is when the performance metric considered is the average probability of error that the decoded support set is not exactly equal to the original support set. But for the framework where ϵ-fraction of the defective items are allowed to be missed, only the lower bound on the number of tests required is given.

IV. PROPOSED SCHEME

The main difference between the SAFFRON scheme and our scheme is that we use a left and right regular sparse graphs in the first stage for the binning operation.

Definition 8 (Left-and-right-regular sparse graph ensemble)**.** We define $\mathcal{G}_{l,r}(N, M_1)$ to be the ensemble of left-and-right-regular graphs where the Nl edge connections from the left and $M_1 r (= Nl)$ edge connections from the right are paired up according to a permutation π_{Nl} chosen at random.

Let $\mathbf{T}_G \in \{0,1\}^{M_1 \times N}$ be the adjacency matrix corresponding to a graph $G \in \mathcal{G}_{l,r}(N, M_1)$ i.e., each column in $\mathbf{T}_G$ corresponding to a variable node has exactly l ones and each row corresponding to a bin node has exactly r ones. And let the universal signature matrix be $\mathbf{U}_r \in \{0,1\}^{h\times r}$. Thus the total number of tests is $M = M_1 \times h$ where $h = 6\log r$. Then the overall testing matrix $\mathbf{A} := [\mathbf{A}_1^T, \ldots, \mathbf{A}_{M_1}^T]^T$ where $\mathbf{A}_i$ defining the h tests at i^{th} bin is given by

$$\begin{aligned} \mathbf{A}_i &= [\mathbf{0}, \ldots, \mathbf{0}, \mathbf{u}_1, \mathbf{0}, \ldots, \mathbf{u}_2, \mathbf{0}, \ldots, \mathbf{u}_r], \quad \text{where} \\ \mathbf{t}_i &= [0, \ldots, 0,\ 1,\ 0, \ldots,\ 1,\ 0, \ldots,\ 1]. \end{aligned} \tag{2}$$

Note that $\mathbf{A}_i$ is defined by placing the r columns of $\mathbf{U}_r$ at the r non-zero indices of $\mathbf{t}_i$ and the remaining are zero columns.

Definition 9 (Regular SAFFRON)**.** We define the ensemble of testing matrices for our scheme to be $\mathcal{G}_{l,r}(N, M_1) \times \mathbf{U}_r$ where a graph G is chosen from $\mathcal{G}_{l,r}(N, M_1)$, a signature matrix is chosen from $\mathbf{U}_r$ and the testing matrix is defined as given in Eq. (2). Note that the total number of tests for this testing scheme is $6M_1 \log r$ where $r = \frac{Nl}{M_1}$.

For the regulr SAFFRON testing ensemble defined in Def. 9, we employ the same peeling based decoder described in Sec. III.

Now we consider the performance analysis of the regular SAFFRON scheme under the peeling based decoder. Similar to the SAFFRON scheme we will analyze the peeling decoder and the bin decoder separately and use union bound to bound the total error probability. As we have already mentioned the analysis of just the peeling decoder can be carried out by considering a simplified peeling decoder on a pruned graph with only the non-zero variable nodes remaining.

Definition 10 (Pruned graph ensemble)**.** We will define the pruned graph ensemble $\tilde{\mathcal{G}}_{l,r}(N, K, M_1)$ as the set of all graphs obtained from removing a random $N - K$ subset of variable nodes from a graph from the ensemble $\mathcal{G}_{l,r}(N, M_1)$. Note that graphs from the pruned ensemble have K variable nodes with a degree l whereas the right degree is not regular anymore.

Lemma 11 (Edge d.d of pruned graph)**.** For the pruned graph ensemble $\tilde{\mathcal{G}}_{l,r}(N, K, M_1)$ it can be shown in the limit $K, N \to \infty$ that edge d.d coefficients $\rho_1 = e^{-\lambda}$ and $\rho_2 = \lambda e^{-\lambda}$ where $\lambda = \frac{l}{C}$ for $M_1 = CK$ for a constant C. Note that even if our initial ensemble is left-and-right-regular the pruned graph has asymptotically same degree distribution as in the SAFFRON scheme where the initial graph is from left-regular ensemble.

Lemma 12. For the pruned graph ensemble $\tilde{\mathcal{G}}_{l,r}(N, K, M_1)$ the oracle-based peeling decoder fails to peel off atleast $(1-\epsilon)$ fraction of the variable nodes with exponentially decaying probability for $M_1 = C(\epsilon)K$ where $C(\epsilon)$ for various ϵ is given in Table. III.

Proof. From Lemma. 11 we know that the edge degree distribution coefficients ρ_1 and ρ_2 are identical to that of the SAFFRON scheme and hence the same DE equations can be used here. Therefore the exact same proof as the proof of Lemma. 6 can be employed here. □

Theorem 13. The regular SAFFRON framework we proposed based on left-and-right-regular sparse graphs recovers atleast a $(1-\epsilon)$ fraction of the defective items for arbitrarily-small ϵ with high probability $1 - O(\frac{K}{N^2})$. The number of tests is $m = 6C(\epsilon)K\log_2\frac{N}{K}$ where $C(\epsilon)$ is given in Table. III and the computational complexity of the decoding is $O(K\log N)$.

Proof of Lem. 11. We will first derive $R(x)$ for the pruned graph ensemble and then use the relation[6] $\rho(x) = \frac{R'(x)}{R'(1)}$ to derive the edge d.d . Note that all the check nodes have a uniform degree r before pruning. When pruning we are removing a $N-K$ subset of variable nodes at random i.e., asymptotically this is equivalent to removing each edge from the graph with a probability $1-\epsilon$ where $\epsilon := \frac{K}{N}$. Under this process the right-node d.d can be written as

$$R_1 = r\epsilon(1-\epsilon)^{r-1}, \quad \text{and similarly}$$
$$R_i = \binom{r}{i}\epsilon^i(1-\epsilon)^{r-i},$$

thus giving us $R(x) = (\epsilon x + (1-\epsilon))^r$. This gives us

$$\begin{aligned}\rho(x) &= \frac{r\epsilon(\epsilon x + (1-\epsilon))^{r-1}}{r\epsilon} \\ &= (\epsilon x + (1-\epsilon))^{r-1}.\end{aligned}$$

Thus we can compute that $\rho_1 = (1-\epsilon)^{r-1}$ and $\rho_2 = (r-1)\epsilon(1-\epsilon)^{r-2}$. We evaluate these quantities asymptotically as $K, N \to \infty$ and $M_1 = CK$.

$$\begin{aligned}\lim_{K,N\to\infty}\rho_1 &= \lim_{K,N\to\infty}(1-\frac{K}{N})^{\frac{Nl}{CK}-1} \\ &= e^{-\lambda} \qquad \text{where } \lambda = \frac{l}{C}\end{aligned}$$

Similarly we can show $\lim_{K,N\to\infty}\rho_2 = \lambda e^{-\lambda}$. □

V. Proofs

References

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

[2] K. Lee, R. Pedarsani, and K. Ramchandran, "Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes," *arXiv preprint arXiv:1508.04485*, 2015.

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

[4] A. Mazumdar, "Nonadaptive group testing with random set of defectives via constant-weight codes," *arXiv preprint arXiv:1503.03597*, 2015.

[5] X. Li, S. Pawar, and K. Ramchandran, "Sub-linear time compressed sensing using sparse-graph codes," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 1645–1649, 2015.

[6] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.