EFFICIENT APPROXIMATE RECOVERY FROM POOLED DATA USING DOUBLY REGULAR POOLING SCHEMES

MAX HAHN-KLIMROTH, DOMINIK KAASER, MALIN RAU

Abstract. In the pooled data problem we are given n agents with hidden state bits, either 0 or 1. The hidden states are unknown and can be seen as the underlying *ground truth* σ. To uncover that ground truth, we are given a querying method that queries multiple agents at a time. Each query reports the sum of the states of the queried agents. Our goal is to learn the hidden state bits using as few queries as possible.

So far, most literature deals with *exact reconstruction* of *all* hidden state bits. We study a more relaxed variant in which we allow a small fraction of agents to be classified incorrectly. This becomes particularly relevant in the *noisy* variant of the pooled data problem where the queries' results are subject to random noise. In this setting, we provide a doubly regular test design that assigns agents to queries. For this design we analyze an approximate reconstruction algorithm that estimates the hidden bits in a greedy fashion. We give a rigorous analysis of the algorithm's performance, its error probability, and its approximation quality. As a main technical novelty, our analysis is uniform in the degree of noise and the sparsity of σ. Finally, simulations back up our theoretical findings and provide strong empirical evidence that our algorithm works well for realistic sample sizes.

1. Introduction

In this paper we consider the *pooled data problem* with *additive queries*, defined as follows. We are given n agents $x_1, \dots, x_n$. Each agent x_i has a *hidden state bit* $\sigma_i \in \{0, 1\}$. The vector $\boldsymbol{\sigma} \in \{0, 1\}^n$ is called the *ground truth* and is given as a sequence on n independent Be(p)−coin-flips. The goal is to identify the agents with bit one, i.e., infer the ground truth. To this end, we can *query* sets of agents: each query pools a certain number of agents together, measures their hidden states, and returns the sum of the queried agents' states. Our task is two-fold. First, we have to design the *query graph* G that assigns agents to queries. Second, we have to devise an algorithm that reconstructs the agents' bits given this pooling design and the queries' results.

There is a substantial body of work on the pooled data problem and the related problem of *group testing* in many communities, including statistical physics [2, 15], information theory [5, 33, 9], machine learning [36, 27], and distributed computing [18, 19]. However, all these related works typically consider the problem of *exact reconstruction*. Coming from a more applied direction, we observe that in many practical applications (e.g., in medicine or machine learning) queries are typically subject to (random) errors. Therefore in this paper, we analyze a relaxation: *approximate* recovery. Our goal is to correctly identify *most* of the agents' bits: we allow a fraction of εk many agents to be classified incorrectly for some small $\varepsilon > 0$, where $k = ||\boldsymbol{\sigma}||_1$ is the number of agents with bit

The research was supported by the German Research Council, grant DFG FOR 2975.

one. As we will see, approximate recovery can be done much more efficiently than exact recovery given realistic sample sizes.

1.1. **Background.** The problem fits well into the so-called *teacher-student model* introduced by Gardner and Derrida [17]. This model provides the fundamental means of our analysis. The setup is the following: a teacher aims to convey some *ground truth* to a student. Rather than directly providing the ground truth to the student, the teacher generates observable data from the ground truth via some statistical model and passes both the data and the model to the student. The student now aims to infer the ground truth from the observed data and the model.

In many real-world scenarios the time to reconstruct the ground truth from the queries' results is dominated by the time to perform a single query. For instance, in the setting of a life-sciences laboratory queries may be performed by automated pipetting machines that pool samples together and run an automated bio-medical test such as DNA screening [7, 32] or PCR tests [6]. In a technological setting, queries may involve computations by a neural network on GPUs in a GPU cluster system [28, 29, 36]. Finally, various tasks in computational biology [14, 7, 32], traffic monitoring [34], or confidential data transfer [1, 12] use decoding techniques from pooled data. Since these biological or technological processes are typically quite time-consuming and in order to obtain a substantial speed-up we focus on *non-adaptive* schemes where all queries are specified upfront and executed simultaneously in parallel.

In both the technological setting and the life-sciences setting we observe that queries are subject to random noise. Therefore we consider the noisy channel model introduced by Gebhard et al. [19] and assume that the agents' hidden bits are subject to random bit flips when read by the queries. In particular, we assume that with a certain probability $S(1,0)$ a one-bit is read but reported as zero (*false negative*), and with a certain probability $S(0,1)$ a zero-bit is read but reported as one (*false positive*). Our model also includes the Z-channel with $S(0,1) = 0$ where only false negatives occur, i.e., errors of the form $1 \to 0$. The Z-channel is particularly important from a practical point of view: it captures the fact that typically in many applications the false-positive probability is insignificant [11, 35].

1.2. **Related Work.** The pooled data problem finds its roots in the 1970s [13] and has since then been frequently studied. This reconstruction task of the ground truth is commonly studied from two angles. From an *information-theoretic* point of view one asks for the minimum number of queries that are necessary and sufficient such that, given unlimited computational power, inference is possible with high probability. From an *algorithmic* point of view one asks for an efficient decoding algorithm. In both settings we distinguish between *exact* inference where an algorithm (efficient or not) reconstructs the ground truth completely and *approximate* inference where all but εnp entries are correctly recovered.

In the literature it is common to distinguish between the *linear regime* ($p = \Theta(1)$) and the *sublinear regime* ($p = n^{-(1-\theta)}$ for some $\theta \in (0,1)$). Typically, techniques for those regimes vary dramatically and the critical phase-transition $\theta \to 1$ is not studied. Furthermore, many related works [2, 15, 18, 19, 21] assume not only knowledge of p but also of k, the exact number of agents with bit one. In the absence of noise knowing k or knowing p is essentially equivalent: the number of agents with bit one is strongly

concentrated around the mean, and a single global query outputs the exact value of k. Under noise, however, things become much more complicated.

Djackov [13] provided an information-theoretic lower bound m_{PD} which states that, even when given unlimited computational power, no algorithm can infer the ground truth with less than

$$m_{PD} \sim 2n\mathcal{H}\left(\frac{k}{n}\right)\frac{\ln \frac{n}{k}}{\ln k}$$

queries. Here, $\mathcal{H}(\alpha) = -\alpha\ln(\alpha) - (1-\alpha)\ln(1-\alpha)$ denotes the entropy. In the early 2000s, Grebinski and Kucherov [20] proved that exponential-time constructions can solve the pooled data problem with no more than $2m_{PD}$ queries. More recently it was established by Scarlett and Cevher [31] (for the case $k = \Theta(n)$) and by Gebhard et al. [18], Feige and Lellouche [16] (for the case $k \sim n^\theta$) that m_{PD} queries suffice from an information-theoretic point of view.

Less is known from the algorithmic perspective. For a long time, only algorithms that exceed m_{PD} by a factor of $\Omega(\ln n)$ were known: in the linear regime message-passing algorithms were proposed, and non-rigorous ideas from statistical physics suggest that those algorithms require $\Theta(n)$ queries [15]. In the sublinear regime ideas from coding theory [26, 25], thresholding algorithms [18, 19] and algorithms for group testing [5] were studied, all of which require $\mathrm{O}(k\ln n)$ queries. Recently an efficient algorithm that requires $\mathrm{O}(k)$ queries was proposed [21], closing the $\ln n$-gap in the sublinear regime in the absence of noise. While this algorithm is efficient from a theoretical point of view, the proofs and techniques used in the construction require tremendously large values of n. In the linear regime this gap is still an open question [16].

Most contributions use an almost regular random graph as a pooling design. This means that either queries select agents uniformly at random, or agents select tests uniformly at random or, sometimes, the mapping is based on independent coin-flips. For large values of n, all those approaches are expected to become equivalent on dense pooling designs due to strong concentration properties. However, on small instances, fluctuations in the number of queries per agent or the number of agents per query might contribute to the performance of algorithms. Works that explicitly focus on small values of n are only known in the related *group testing* problem. In group testing, a query returns the information whether at least one agent in the query has bit one. The findings suggest that almost-regular designs perform best if the fluctuations are small and queries are not too similar [33, 30].

To the best of our knowledge, all rigorous algorithmic results with respect to the pooled data problem concern *exact* recovery. Rigorous results with respect to information-theoretic aspects are studied by Scarlett and Cevher [31], and heuristics from statistical physics suggest that approximate recovery is as hard as exact recovery [15]. Nevertheless, in the group testing problem, the class of low-degree polynomials (including important classes of algorithms such as local algorithms) are understood with respect to approximate recovery [10].

1.3. **Our contribution.** We study an algorithm based on the thresholding approach by Gebhard et al. [18, 19] for the approximate recovery problem under noise and for a broad range of prior probabilities p. More precisely, we only require $p \gg n^{-1}\ln n$, and

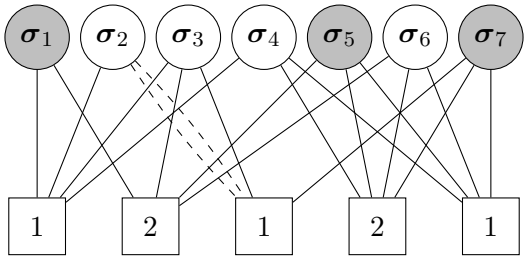

FIGURE 1. A small example with $n = 7$ agents and ground truth $\boldsymbol{\sigma} = (1, 0, 0, 0, 1, 0, 1)$. A gray vertex color indicates that the agent has bit one. The underlying multi-graph defines the queries. Ultimately, the goal is to reconstruct $\boldsymbol{\sigma}$ as well as possible given $\boldsymbol{G}$ and the queries' results.

we explicitly do not require knowledge of k, the number of agents with bit one. We give an explicit bound on the number of queries required for the algorithm to recover all but εnp agents correctly with probability at least $1 - \delta - o(1)$. To this end we modify the pooling design proposed by Gebhard et al. [18, 19] such that all agents are in the same number of queries and each query contains the same number of agents (up to ± 1 due to divisibility conditions). This doubly regular design is harder to study since it introduces stochastic dependencies. However, as our simulations show, the design is superior when the number of agents is small and the queries are subject to noise.

Outline. In the next section we formally introduce our pooling design, present our reconstruction algorithm, and state our main theorem. Then in Section 3 we present implementation details and empirical results of our simulations. The formal analysis of our main result is presented in Section 4. Finally, in Section 5 we discuss our findings and conclusions, and we present some open problems.

2. Model and Contribution

Let $x_1, \dots, x_n$ denote the n agents such that any agent has bit one with probability p independently from each other. We let $\boldsymbol{k}$ denote the (random) number of agents with hidden bit one, thus $\mathbb{E}[\boldsymbol{k}] = np$. We assume in the following that $p = \omega\big(n^{-1} \ln n\big)$.

Recall that the pooling design is defined via a bipartite (multi-) graph that assigns agents to queries. In our pooling design we assume that the number of agents per query Γ is fixed, and we define the following pooling design $\boldsymbol{G}_\Gamma$. Given the number of queries m and the number of agents per query Γ with $n^\delta \sqrt{n(mp)^{-1}} \le \Gamma \le n^{1-\delta}, \delta > 0$, we set the sequence of agent-degrees $\Delta_1, \dots, \Delta_n$ such that

$$|\Delta_i - \Delta_j| \le 1 \quad \text{and} \quad \sum_{i=1}^{n} \Delta_i = m\Gamma.$$

With these agent degrees we let $\boldsymbol{G} = \boldsymbol{G}_\Gamma$ be a bipartite multi-graph chosen uniformly at random chosen according to the configuration model (see, for instance [23]). An example of such a pooling design can be seen in Figure 1. Given $\boldsymbol{G}$, let $\Delta_i^\star$ denote the number of distinct queries agent i is part of. Furthermore, let $\Delta = n^{-1} \sum_{i=1}^{n} \Delta_i$ and $\Delta^\star = n^{-1} \sum_{i=1}^{n} \Delta_i^\star$ denote the average agent-degrees.

Algorithm 1: Reconstruction Algorithm with threshold T

I. Perform Measurements

add m query nodes to the network

at *query node* a_j **do**

sample agents $\{v_1, \dots, v_\Gamma\}$ u.a.r. with replacement

measure $\hat{\boldsymbol{\sigma}}_j \simeq \sum_{i=1}^{\Gamma} \boldsymbol{\sigma}(v_i)$ // $\hat{\boldsymbol{\sigma}}_j$ is subject to noise!

send message $\hat{\boldsymbol{\sigma}}_j$ to each distinct neighbor

at *agent* x_i **with** *incoming message* $\hat{\boldsymbol{\sigma}}_j$ **do**

update score $\Psi_i \leftarrow \Psi_i + \hat{\boldsymbol{\sigma}}_j$

II. Output Estimate

at *agent* x_i **with** *score* Ψ_i **do**

if $\Psi_i \leq T$ **then** agent x_i outputs 0

else agent x_i outputs 1

2.1. **Thresholding Algorithm.** We study a distributed combinatorial reconstruction algorithm that makes use of the following observation: if an agent has hidden bit one it increases the queries' results every time it is queried. The expected sum of an agent's queries' results is therefore larger if the agent has bit one. The expected difference in this *score* between agents with bit one and agents with bit zero becomes larger if more queries are conducted. Therefore, the algorithm uses a threshold value T and declares all agents with a score larger than T as having bit one. This algorithm was originally introduced to study the noiseless variant of the pooled data problem for exact recovery [18] and was later studied for the purpose of exact recovery under noise [19]. It is formally specified in Algorithm 1.

2.2. **Formal Results.** Our main theorem includes results for a general noise model, the noisy channel model. It is formally defined as follows.

Definition 2.1 (Noisy-Channel)**.** *Under the noisy channel a query reads the bit of an agent according to the following noise channel with probability given by S, where*

$$S(i,j) = \mathbb{P}(\textit{bit } j \textit{ is read } \mid \textit{bit } i \textit{ was sent }).$$

We assume that all entries of S are real values not dependent on n with $S(1,1) - S(0,1) > 0$.

The last assumption simply says that reading a single bit as one is more likely if the true signal was present as well. From here on, we let $\boldsymbol{G}$ be an instance of the almost (Γ, Δ)-doubly regular pooling design. Next, we properly define what is meant by *approximate reconstruction*. Fix a failure probability δ and an approximation fraction $\varepsilon > 0$.

Definition 2.2 (ε-recovery)**.** *An algorithm is said to be an ε-recovery algorithm if, given the pooling design $\boldsymbol{G}$ and the queries' results $\hat{\boldsymbol{\sigma}}$, it outputs an estimate $\tilde{\sigma}$ such that $\mathrm{d}_H(\boldsymbol{\sigma}, \tilde{\sigma}) \leq 2\varepsilon ||\boldsymbol{\sigma}||_1$.*

Observe that, if $||\boldsymbol{\sigma}||_1 = ||\tilde{\sigma}||_1$, an ε-recovery algorithm declares at most $\varepsilon \boldsymbol{k} = \varepsilon||\boldsymbol{\sigma}||_1$ agents with bit one to have bit zero and vice versa. Our main result states an upper bound on the required number of queries such that Algorithm 1 achieves ε-recovery with failure probability $\delta + o_n(1)$. Here, $o_n(1)$ denotes a function that tends to zero with $n \to \infty$.

Theorem 2.3. *Let S be the channel matrix and let*

$$L = L(n, p, S) = \frac{(S(1,1) - S(0,1))^2}{2n(S(0,1) + p(S(1,1) - S(0,1)))}$$

and

$$T^\star = \Delta S(1,1) - \left(\frac{1}{2} - \frac{\ln\left(\frac{1}{p}\right)}{2Lm}\right)\Delta(S(1,1) - S(0,1)).$$

Then Algorithm 1 with threshold $T^\star$ is with probability $1-\delta-o_n(1)$ an ε-recovery algorithm on the almost (Γ, Δ)-doubly regular pooling design if the number of queries m is at least

$$m > \frac{1}{L}\left(\ln\left(\frac{1}{p}\right) + 2\ln\left(\frac{2}{\varepsilon\delta}\right) + 2\sqrt{\ln\left(\frac{2}{\varepsilon\delta}\right)\ln\left(\frac{2}{\varepsilon\delta p}\right)}\right).$$

While our main theorem is an asymptotic result (due to the $o_n(1)$ in the failure probability), the next section analyses the performance of Algorithm 1 for a decent number of agents empirically.

3. Simulations

There are essentially three closely related strategies to generate a random graph with expected number of queries per agent Δ and expected number of agents per query Γ:

- *Bernoulli*: for any query and any agent, a random coin is flipped whether the agent is part of the query.
- *one-sided regular*: every query chooses exactly Γ agents uniformly at random.
- *doubly regular*: establish a random mapping between agents and queries such that any agent is in $\Delta \pm 1$ queries and every query contains Γ agents.

Both regular designs can be defined in a way such that an agent appears multiple times in a query (the underlying pooling graph has multi-edges) and in a way such that the underlying pooling graph is a simple graph. Our formal results are in line with recent literature and allow agents to be queried multiple times. From a theoretical, asymptotic point of view, it is indeed known that our algorithm performs better when multi-edges are allowed [18, 19]. However, for realistic sizes of n, it is far from clear whether agents appearing multiple times in queries help.

We implemented our simulations in the C++ programming language. As a source of randomness, we use the Mersenne Twister `mt19937_64` provided by the C++11 <`random`> library. In a simulation run, we first generate the random pooling scheme by constructing a random bipartite graph according to one of the three strategies described above. Then we perform a faithful simulation of the distributed system where we compute the agents' scores as defined in Algorithm 1. Note that for the sake of reproducibility and comparability we did not use a different, random number of agents with bit one for each simulation run. Instead we set k to either $n^{1/8}$, $n^{1/4}$, or $n^{1/2}$. In the following, we present data for $n = 10^3$ agents. Each data point stems from 100 independent simulation runs.

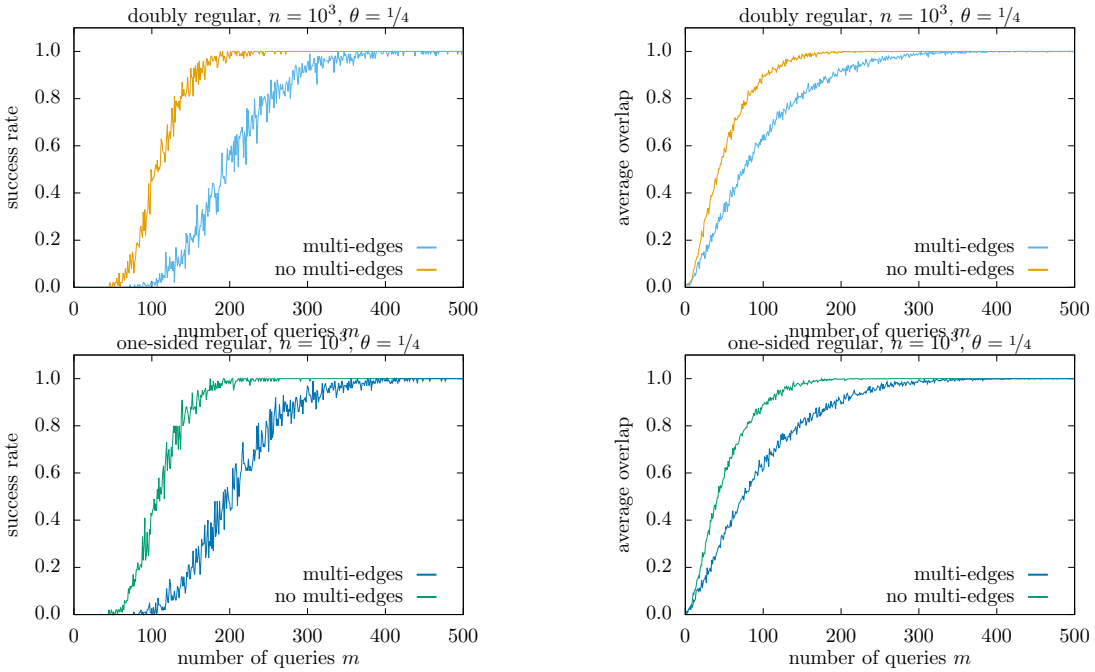

FIGURE 2. Comparison of the algorithm's performance on designs with multi-edges and the corresponding designs without multi-edges. The left plots show the fraction of trials in which at least 90 % of the agents with hidden bit one were learned correctly. The right plots show the average overlap, i.e., the average number of correctly classified agents with bit one. All simulations are conducted without noise.

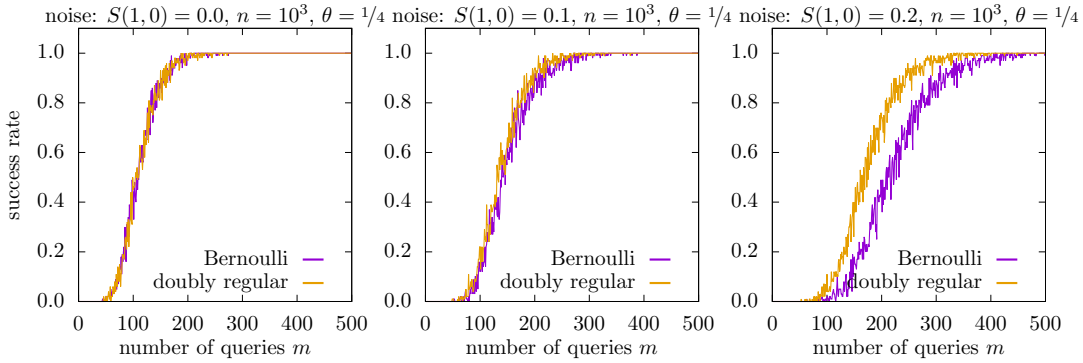

FIGURE 3. Comparison of the algorithm's performance on designs with a different degree of regularity. If no noise is present, the Bernoulli model and the doubly regular model perform equally well. Under noise, however, the additional regularity on the agents improves the performance of the algorithm visibly.

We show the success rate, defined as the fraction of simulations in which more than 90% of agents with hidden bit one were classified correctly, and the average overlap, defined

as the overall fraction of correctly classified agents with hidden bit one. The simulation software and all necessary tools to reproduce our plots will be made available on our public GitHub repository.

The first question of our empirical analysis is whether the appearance of multi-edges is beneficial. Our simulations reveal that this is not the case. Adding insult to injury, our empirical data show that multi-edges in the pooling graph are actually harmful, see Figure 2.

The second question of interest in the influence of the regularity of the pooling scheme. In group testing, the more regular the design is, the fewer queries are required [33, 30]. The next set of results studies this effect in the pooled data problem. Interestingly, the degree of regularity has only a limited impact on the performance if all queries report perfect information: in the noiseless setting, the algorithm performs equally well on the three model variants. In contrast, if queries are subject to noise, the same effect as in group testing is visible: doubly regular designs dramatically outperform non-regular models, see Figure 3.

4. Formal Analysis

In this section, we first prove some basic properties of the pooling scheme and pin down the distribution of the score used in the algorithm. Afterwards, we analyze how an agent's score depends on the agent's hidden bit. It turns out that the scores between agents of both types follow a multivariate hypergeometric distribution and the corresponding expected values differ. We finally optimize a threshold such that due to concentration properties of the hypergeometric distribution, most scores below the threshold belong to agents with bit zero and most scores above the threshold belong to agents with bit one.

4.1. Properties of the Pooling Scheme.

The first lemma shows that although there is a substantial number of multi-edges in the pooling graph, any individual agent appears in almost only distinct queries.

Lemma 4.1. *If $\Gamma = o(n)$, we have with probability $1 - o(1)$ that for all $1 \le i \le n$,*

$$\Delta_i^\star = (1 + o(1))\Delta.$$

Proof. The probability that a specific agent x_i is part of a specific query a_j is given by

$$p_c = 1 - \frac{\binom{\sum_j \Delta_j - \Delta_i}{\Gamma}}{\binom{\sum_j \Delta_j}{\Gamma}} = (1 + o(1))\left(1 - \frac{\binom{n\Delta - \Delta}{\Gamma}}{\binom{n\Delta}{\Gamma}}\right).$$

A short calculation that uses the equality $n\Delta = m\Gamma$ verifies

$$\begin{aligned} p_c &= (1 + o(1))\left(1 - \prod_{j=1}^{\Delta}\left(1 - \frac{\Gamma - j}{n\Delta - j}\right)\right) \\ &= (1 + o(1))\left(1 - \left(1 - \frac{1}{m}\right)^{\frac{m\Gamma}{n}}\right) \\ &= (1 + o(1))\left(1 - \exp\left(-\frac{\Gamma}{n}\right)\right). \end{aligned}$$

As the events of being part of a specific query are negatively associated under the configuration model, the Chernoff bound establishes that $\Delta_i^\star = (1+o(1))mp_c$ [8]. On the other hand, $\Delta = \frac{m\Gamma}{n}$ by construction. If $\Gamma = o(n)$ as supposed, we have by a Taylor expansion that $p_c = (1+o(1))\frac{\Gamma}{n}$, thus $\Delta_i^\star = (1+o(1))\frac{m\Gamma}{n}$ and the lemma follows. □

Next, we define the agents' *scores* $\boldsymbol{\psi}_i$. In the end, the score will be used to decide whether an agent's hidden bit is zero or one. For an agent x_i, given the pooling graph $\boldsymbol{G}$ and the sequence of queries' results $\hat{\boldsymbol{\sigma}}$, let

$$\boldsymbol{\psi}_i = \sum_{j=1}^{m} \mathbf{1}\{x_i \in a_j\}\hat{\boldsymbol{\sigma}}_j$$

be the score of x_i. Clearly, this score is expected to be larger by the additive number $\Delta_i^\star$ if $\boldsymbol{\sigma}_i = 1$. In previous works, the pooling design had no restrictions on the agent's degree which induced a lot of stochastic independence on $\boldsymbol{\psi}$. Moreover, the designs were not built on the configuration model and it was supposed that the exact number of agents with bit one was known a priori. In the current contribution, we need to be much more careful. Throughout this section, we implicitly condition on the $\sigma-$algebra generated by the edges of the underlying random graph.

The next lemma describes the distribution of the agents' scores under perfect queries. While in the actual model we assume to queries to be noisy, this case will help to understand the (much more involved) general distribution.

Lemma 4.2 (Score distribution without noise)**.** *If the channel matrix S is the identity matrix (no noise), we find that*

$$\boldsymbol{\psi}_i \overset{d}{=} \boldsymbol{X}_i + \Delta_i \mathbf{1}\{\boldsymbol{\sigma}_i = 1\}$$

where $\boldsymbol{X}_i$ is distributed as follows. Given $\boldsymbol{\sigma}$, it is hypergeometrically distributed with population size $\sum_{j\neq i}\Delta_j$, $\sum_{j\neq i}\mathbf{1}\{\boldsymbol{\sigma}_j = 1\}\Delta_j$ successes and $\Gamma\Delta_i^\star - \Delta_i$ draws.

Proof. The score $\boldsymbol{\psi}_i$ consists of two parts. First, the contribution of x_i. Second, the contribution of all agents in the second neighborhood of x_i. Observe that, due to the appearance of multi-edges, x_i can be part of this second neighborhood itself. We need to calculate the distribution of the number of agents with bit one connected to $\partial_{\boldsymbol{G}}(x_i)$ in the subgraph $\boldsymbol{G} - x_i$. By definition of the configuration model, we find that in this graph, the queries need to choose exactly $\Gamma\Delta_i^\star - \Delta_i$ distinct half-edges. The score is increased by one if and only if a half-edge is connected to an agent with bit one under the ground truth. There are exactly $\sum_{j\neq i}\mathbf{1}\{\boldsymbol{\sigma}_j = 1\}\Delta_j$ half-edges with this property while in total, there remain $\sum_{j\neq i}\Delta_j$ half-edges in $\boldsymbol{G} - x_i$. Finally, if $\boldsymbol{\sigma}_i = 1$, thus x_i has bit one, then the score $\boldsymbol{\psi}_i$ is increased by Δ_i. □

For the sake of intuition, we remark that under the knowledge of $\boldsymbol{k}$,

$$\sum_{j\neq i}\Delta_j \approx (n-1)\Delta$$

$$\sum_{j\neq i}\mathbf{1}\{\boldsymbol{\sigma}_j = 1\}\Delta_j \approx (\boldsymbol{k} - \mathbf{1}\{\boldsymbol{\sigma}_i = 1\})\Delta$$

and so $\boldsymbol{X}_i$ has approximately as good concentration properties as a

$$\mathrm{Bin}\left(\Delta^\star\Gamma - \Delta, \frac{\boldsymbol{k} - \mathbf{1}\{\boldsymbol{\sigma}(j) = 1\}}{n-1}\right)$$

random variable.

The next lemma generalizes the distribution of the agents' scores to the more general model in which queries are exposed to noise. Observe that for S being the identity matrix, this boils down to the previous lemma.

Lemma 4.3 (Score distribution under noise)**.** *Given the channel matrix S and $\boldsymbol{k}$, let $N_i = \sum_{j\neq i}\Delta_j$, $K_i = \sum_{j\neq i}\mathbf{1}\{\boldsymbol{\sigma}_j = 1\}\Delta_j$ and $n_i = \Gamma\Delta_i^\star - \Delta_i$. Let $\boldsymbol{Y}_i$ be a multivariate hypergeometrically distributed random variable on four types in a population of size N_i with n_i draws. We let $K_i(1,1) = K_iS(1,1)$, $K_i(0,1) = (N_i - K_i)S(0,1)$, $K_i(1,0) = K_iS(1,0)$ and $K_i(0,0) = (N_i - K_i)S(0,0)$ be the number of half-edges in the population of the four types. We have*

$$\begin{aligned}\boldsymbol{\psi}_i \overset{d}{=} \boldsymbol{Y}_i(1,1) + \boldsymbol{Y}_i(0,1) &+ \mathrm{Bin}(\Delta_i, S(1,1))\mathbf{1}\{\boldsymbol{\sigma}_i = 1\}\\ &+ \mathrm{Bin}(\Delta_i, S(0,1))\mathbf{1}\{\boldsymbol{\sigma}_i = 0\}.\end{aligned}$$

Furthermore, we have that $\boldsymbol{\psi}_i$ is a sum of negatively associated Bernoulli random variables.

Proof. As a first step, we prove that

$$\begin{aligned}\boldsymbol{\psi}_i \overset{d}{=} &\mathrm{Bin}(\boldsymbol{X}_i, S(1,1)) + \mathrm{Bin}(N_i - \boldsymbol{X}_i, S(0,1))\\ &+ \mathrm{Bin}(\Delta_i, S(1,1))\mathbf{1}\{\boldsymbol{\sigma}_i = 1\} + \mathrm{Bin}(\Delta_i, S(0,1))\mathbf{1}\{\boldsymbol{\sigma}_i = 0\}.\end{aligned} \tag{1}$$

For any of the half-edges connected to agents in $\boldsymbol{G} - x_i$, any of the $\boldsymbol{X}_i$ one-bits is transmitted correctly independently from everything else with probability $S(1,1)$, so given $\boldsymbol{X}_i$, the number of such bits is binomially distributed. Analogously, any of the $N_i - \boldsymbol{X}_i$ zero-bits is transmitted falsely with probability $S(0,1)$. Finally, for any of the Δ_i edges connected to x_i, the same channel argument is applied as the transmission is independent from the rest. This proves (1).

The lemma now follows from (1) and the following combinatorial insight. Given N balls out of which K are red and $N-K$ are blue, some red balls will be colored lime and some blue balls will be colored orange. In the end, we want to describe the distribution of the sum of the red and orange balls in a sample. The two following random experiments yield the same distribution of the sum of orange and red balls.

- First random experiment
  - Draw n balls without replacement out of the N balls, let S be the sample.
  - For any red ball in S, color it independently from anything else with probability p in lime.
  - For any blue ball in S, color it independently from anything else with probability q in orange.
- Second random experiment
  - For any red ball out of the N balls, color it independently from anything else with probability p in lime.
  - For any blue ball out of the N balls, color it independently from anything else with probability q in orange.

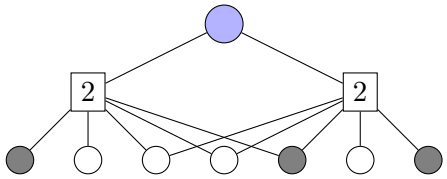

FIGURE 4. Example of a second neighborhood of the blue agent in the pooling graph. The distribution of the number of agents with bit one is independent from the agent's bit.

– Draw n balls without replacement out of the N balls.

The distribution of Equation (1) is given by the first experiment while the distribution described in Lemma 4.3 is given by the second experiment. This concludes the distributional equality.

The second statement, namely the negative association, is a well-known property of the (multivariate) hypergeometric distribution [24]. □

4.2. **Differences Based on the Hidden Bit.** Finally, we need to make sure that at most $2\varepsilon\boldsymbol{k}$ agents are classified incorrectly. We will order the agents by their centralized scores $\boldsymbol{\psi}_i - \mathbb{E}[\boldsymbol{X}_i]$ and declare any agent with a score above a certain threshold T as having bit one and to have bit zero otherwise. We only need to conduct enough queries such that the concentration properties of $\boldsymbol{\psi}_i - \mathbb{E}[\boldsymbol{X}_i]$ are good enough such that only $2\varepsilon\boldsymbol{k}$ errors occur with probability $1-\delta$.

It is easily verified that

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{\psi}_i \mid \boldsymbol{\sigma}] \\
&= n_i \frac{K_i}{N_i} S(1,1) + n_i \frac{N_i - K_i}{N_i} S(0,1) \\
&\quad + \Delta_i S(1,1)\mathbf{1}\{\boldsymbol{\sigma}_i = 1\} + \Delta_i S(0,1)\mathbf{1}\{\boldsymbol{\sigma}_i = 0\} \\
&= (\Gamma\Delta_i^\star - \Delta_i) \\
&\quad \cdot \left(\frac{\boldsymbol{k}}{n}(S(1,1) - S(0,1)) + S(0,1) + \mathrm{O}\big(n^{-1}\big) \right) \\
&\quad + \Delta_i S(1,1)\mathbf{1}\{\boldsymbol{\sigma}_i = 1\} + \Delta_i S(0,1)\mathbf{1}\{\boldsymbol{\sigma}_i = 0\}
\end{aligned} \tag{2}$$

and thus

$$\mathbb{E}[\boldsymbol{\psi}_i \mid \boldsymbol{\sigma}_i = 1] - \mathbb{E}[\boldsymbol{\psi}_i \mid \boldsymbol{\sigma}_i = 0] = \Delta_i(S(1,1) - S(0,1)).$$

Observe that this difference depends only on the hidden bit of agent x_i and its degree in the underlying graph. It does explicitly not depend on the prior p nor the number of positive hidden bits $\boldsymbol{k}$.

We, therefore, need to establish conditions, such that for all but $\varepsilon\boldsymbol{k}$ agents x_i with bit one and all but $\varepsilon\boldsymbol{k}$ agents x_j with bit zero the contribution

$$|(\boldsymbol{Y}_i(1,1) + \boldsymbol{Y}_i(0,1)) - (\boldsymbol{Y}_j(1,1) + \boldsymbol{Y}_j(0,1))|$$

is, with probability at least $1-\delta$, smaller than $\Delta(S(1,1) - S(0,1)) + \mathrm{O}(1)$ (see Lemma 4.3). In the last step, we used that $|\Delta_i - \Delta| \leq 1$ by definition of the pooling design.

We will give an alternative sufficient condition, namely that for all agents x_i, we find that $(\boldsymbol{Y}_i(1,1) + \boldsymbol{Y}_i(0,1))$ is sufficiently well concentrated around its mean. If we denote

$$p_S = p(S(1,1) - S(0,1)) + S(0,1),$$

then as in (2),

$$\mathbb{E}\left[\boldsymbol{Y}_i(1,1) + \boldsymbol{Y}_i(0,1)\right] = \Gamma\Delta^\star\big(p_S + \mathrm{O}\big(n^{-1}\big)\big).$$

Lemma 4.4. *Let $\boldsymbol{\Xi}_i = \boldsymbol{Y}_i(1,1) + \boldsymbol{Y}_i(0,1)$. For all $\beta > 0$, we find that*

$$\mathbb{P}(|\boldsymbol{\Xi}_i - \mathbb{E}[\boldsymbol{\Xi}_i]| > \beta\Delta) \leq 2\exp\left(-\frac{(1+o(1))\beta^2 m}{2np_S}\right)$$

Proof. By a standard application of the Chernoff bound for the hypergeometric distribution [22] and (2) we have

$$\begin{aligned}&\mathbb{P}(|\boldsymbol{\Xi}_i - \mathbb{E}[\boldsymbol{\Xi}_i \mid \boldsymbol{k}]| > \beta\Delta \mid \boldsymbol{k})\\ &\quad\leq 2\exp\left(-(1+o(1)\frac{\beta^2\Delta^2}{(2+\beta/(\Gamma p_S))\mathbb{E}[\boldsymbol{\Xi}_i \mid \boldsymbol{k}]}\right)\end{aligned}$$

Because $\mathbb{E}[\boldsymbol{\Xi}_i \mid \boldsymbol{k}] = \Delta^\star\Gamma p_S$, and $\beta/(\Gamma p_S) = o(1)$ for any fixed $\beta \in (0,1)$, by Lemma 4.1, the fact that $\Gamma = \frac{n\Delta}{m}$ and $\boldsymbol{k} = pn \pm \mathrm{O}\big(\sqrt{np\ln n}\big)$ with exponentially high probability by the Chernoff bound for the binomial distribution, the lemma follows. □

4.3. **Thresholding the Score.**

Outline. Next, we construct a threshold $T = T(p,m,n)$ with the goal that $\boldsymbol{\psi}_j - \mathbb{E}[\boldsymbol{\Xi}_j] > T$ for all but $\varepsilon\boldsymbol{k}$ agents with hidden bit one and $\boldsymbol{\psi}_i - \mathbb{E}[\boldsymbol{\Xi}_i] < T$ for all but $\varepsilon\boldsymbol{k}$ agents with hidden bit one. Observe that for $\alpha \in (0,1)$,

$$\begin{aligned}&\Delta_j S(0,1) + \alpha\Delta_j(S(1,1) - S(0,1))\\ &\qquad= \Delta_j S(1,1) - (1-\alpha)\Delta_j(S(1,1) - S(0,1)).\end{aligned}$$

Therefore, we optimize the threshold T with respect to α which can be seen as an interpolation between the expected difference in the neighborhood sum for agents of different states. This gives us the bound

$$m > \frac{1}{L}\left(\ln\left(\frac{1}{p}\right) + 2\ln\left(\frac{2}{\varepsilon\delta}\right) + 2\sqrt{\ln\left(\frac{2}{\varepsilon\delta}\right)\ln\left(\frac{2}{\varepsilon\delta p}\right)}\right). \tag{3}$$

The main theorem follows from (3) and the fact that conditioning on $\mathcal{K}$ does not harm the result. Indeed, by (3), the algorithm is with probability $1-\delta$ an ε-recovery algorithm. The analysis is valid, conditioned on $\mathcal{K}$. As discussed earlier, $\mathbb{P}(\mathcal{K}) = 1 - n^{-\Omega(1)}$ by Chernoff's bound. Consequently, the algorithm only fails to be an ε-recovery algorithm with probability $1-\delta$ on the event $\neg\mathcal{K}$ which has probability $o_n(1)$. Therefore, the algorithm is, unconditional, an ε-recovery algorithm with probability at least $(1-\delta)(1-\mathbb{P}(\mathcal{K}))$ which implies Theorem 2.3. □

Formal derivation. It remains to formally prove Equation (3). To this end, define

$$T_\alpha = \Delta_j S(1,1) - (1-\alpha)\Delta_j(S(1,1) - S(0,1))$$

and observe that by Lemmas 4.1 and 4.4, we have

$$\mathbb{P}\big(\boldsymbol{\psi}_j > T_\alpha \mid \boldsymbol{\sigma}_j = 0\big) \leq 2\exp\left((-1+o(1))\frac{\alpha^2(S(1,1)-S(0,1))^2 m}{2np_S}\right) \tag{4}$$

for agents with bit zero and analogously

$$\mathbb{P}\big(\boldsymbol{\psi}_j < T_\alpha \mid \boldsymbol{\sigma}_j = 1\big) \leq 2\exp\left(-(1+o(1))\frac{(1-\alpha)^2(S(1,1)-S(0,1))^2}{2np_S}\right). \tag{5}$$

Let $\tilde{\sigma}^{(\alpha)}$ be the estimate of $\boldsymbol{\sigma}$ by Algorithm 1 using threshold T_α. For brevity, define

$$L = L(n,p,S) = \frac{(S(1,1)-S(0,1))^2}{2np_S}, \quad N_\alpha = \alpha^2 Lm, \quad \text{and} \quad P_\alpha = (1-\alpha)^2 Lm.$$

Define $\mathcal{K}$ as the event that $\boldsymbol{k} \in \big(np - \sqrt{np}\ln n, np + \sqrt{np}\ln n\big)$. This means that conditioned on $\mathcal{K}$ there are roughly as many agents with bit one as we expect. Clearly, Chernoff's bound directly implies $\mathbb{P}(\mathcal{K}) = 1 - n^{-\Omega(1)}$.

We let $\mathcal{P}(\alpha) = \Big\{x_i : \boldsymbol{\sigma}_i = 0, \tilde{\sigma}_i^{(\alpha)} = 1\Big\}$ be the set of false positives and $\mathcal{N}(\alpha) = \Big\{x_i : \boldsymbol{\sigma}_i = 1, \tilde{\sigma}_i^{(\alpha)} = 0\Big\}$ the set of false negatives. By (4) and (5) we find

$$\mathbb{E}[|\mathcal{P}(\alpha)| \mid \mathcal{K}] \leq 2n(1-p)\exp(-(1+o(1))N_\alpha) \tag{6}$$

and

$$\mathbb{E}[|\mathcal{N}(\alpha)| \mid \mathcal{K}] \leq 2np\exp(-(1+o(1))P_\alpha). \tag{7}$$

Therefore, Markov's inequality yields

$$\begin{aligned}\mathbb{P}[|\mathcal{P}(\alpha)| > \varepsilon np \mid \mathcal{K}] &\leq \frac{2(1-p)\exp(-(1+o(1))N_\alpha)}{\varepsilon p}, \\ \mathbb{P}[|\mathcal{N}(\alpha)| > \varepsilon np \mid \mathcal{K}] &\leq \frac{2\exp(-(1+o(1))P_\alpha)}{\varepsilon}.\end{aligned} \tag{8}$$

Thus, it is sufficient to determine the value of α which minimizes the number of queries conducted such that

$$\frac{2\exp(-N_\alpha)}{\varepsilon p} < \delta \quad \text{and} \quad \frac{2\exp(-P_\alpha)}{\varepsilon} < \delta$$

or, equivalently,

$$N_\alpha > \ln\left(\frac{2}{\varepsilon\delta p}\right) \quad \text{and} \quad P_\alpha > \ln\left(\frac{2}{\varepsilon\delta}\right). \tag{9}$$

Because one criterion is increasing in α while the other one is decreasing, the α minimizing both equations can be easily calculated. We find as an optimal value for α that

$$\alpha^\star = \frac{Lm - \ln\left(\frac{2}{\varepsilon\delta}\right) + \ln\left(\frac{2}{\varepsilon\delta p}\right)}{2Lm} = \frac{1}{2} + \frac{\ln\left(\frac{1}{p}\right)}{2Lm}$$

Because α needs to be between zero and one, we directly require

$$m > \frac{\ln\left(\frac{1}{p}\right)}{L} \tag{10}$$

Consequently, the algorithm's threshold reads

$$T_\alpha = \Delta_j S(1,1) - \left(\frac{1}{2} - \frac{\ln\left(\frac{1}{p}\right)}{2Lm}\right)\Delta_j(S(1,1) - S(0,1)). \tag{11}$$

Because

$$N_{\alpha^\star} = (\alpha^\star)^2 Lm \quad \text{and} \quad P_{\alpha^\star} = (1-\alpha^\star)^2 Lm,$$

we have

$$N_{\alpha^\star} = \left(\frac{1}{2} + \frac{\ln\left(\frac{1}{p}\right)}{2Lm}\right)^2 Lm = \frac{1}{4}Lm + \frac{1}{2}\ln\left(\frac{1}{p}\right) + \frac{1}{4Lm}\ln\left(\frac{1}{p}\right)^2,$$

and

$$P_{\alpha^\star} = \left(\frac{1}{2} - \frac{\ln\left(\frac{1}{p}\right)}{2Lm}\right)^2 Lm = \frac{1}{4}Lm - \frac{1}{2}\ln\left(\frac{1}{p}\right) + \frac{1}{4Lm}\ln\left(\frac{1}{p}\right)^2.$$

As a direct consequence, we obtain Equation (3) from (9):

$$m > \frac{1}{L}\Big(\ln(1/p) + 2\ln(2/(\varepsilon\delta)) + 2\sqrt{\ln(2/(\varepsilon\delta))\ln(2/(\varepsilon\delta p))}\Big).$$

Observe that (3) does outnumber (10) and is therefore a sufficient condition.

5. Discussion and Conclusion

We studied the approximate recovery from pooled data under a very generic noise model and state a performance guarantee for a distributed combinatorial algorithm. In the noiseless case when the channel-matrix S becomes the identity matrix our bound on the required number of queries in Theorem 2.3 boils down to

$$m \sim 2k\left(\ln\left(\frac{n}{k}\right) + 2\ln\left(\frac{2}{\varepsilon\delta}\right) + 2\sqrt{\ln\left(\frac{2}{\varepsilon\delta}\right)\ln\left(\frac{2n}{\varepsilon\delta k}\right)}\right),$$

where we set $p = k/n$. In the sublinear regime where the number of agents with a hidden bit one scales as is $k = n^\theta$ for some $\theta \in (0,1)$) our result aligns well with the existing literature. As previously stated, the algorithm was already analyzed under exact recovery. More precisely, Gebhard et al. [18, 19] prove that for exact recovery $m \sim 2.43\left(1+\sqrt{\theta}\right)^2 k\ln(n)$ is sufficient, and their proof reveals that this is also necessary [18]. In our paper we obtain $m \sim 2(1-\theta)k\ln(n)$ for approximate recovery. The leading constant is much smaller than in the exact recovery task. A similar observation holds

with respect to the analysis in [19] for exact recovery under noise. Again, the approximate variant decreases the required constant in front of $k \ln n$ significantly if p becomes larger, but the order of magnitude is preserved.

In the empirical analysis we observe three facts that might be surprising at first glance. First, multi-edges in the pooling graph are harmful to the algorithm's performance. Second, the degree of regularity has only a vanishing influence on the number of queries required in the noiseless variant. Third, the degree of regularity becomes relevant if the queries are subject to noise.

Regarding the existence of multi-edges, asymptotic results show that given multi-edges the algorithm should require slightly fewer queries. More precisely, it is known [18, 19] that a multiplicative factor of $2(1 - \exp(-1/2)) \sim 0.8$ in the number of required queries arises in exact recovery tasks. A similar calculation could be carried out in our analysis: the factor $\Delta^\star \Delta^{-1} = 1 + o(1)$ appears in our formal analysis as well. Of course, this observation is with respect to a solely asymptotic analysis in which, for instance, $\sqrt{m \ln m} \ll m$ is required (see [19]). It is not likely that such assumptions hold for realistic instances. Furthermore, the appearance of the multi-edges reduces the entropy of the test design significantly for smaller values of n. It is known in the related group testing problem that less queries are needed if the system's entropy rises [30]. This is a plausible explanation for the different behavior on small samples in contrast to asymptotic considerations.

Moreover, in group testing it was also observed that similar thresholding approaches work better on doubly regular designs for small n [33]. It is also known that one-sided regular designs and doubly regular designs outperform the simplistic Bernoulli variant substantially [3, 4]. In contrast, in the pooled data problem there is no difference in the performance of the different designs from an information-theoretic point of view [16, 18]. This is due to the high density of the pooling graph compared to the one of group testing: all agents are, more or less, exchangeable in the pooled data graphs such that the entropy of the different models is comparable. Therefore it is well explainable that no strong differences are observed.

Finally, we observed that if noise arises the regularization of the pooling scheme helps significantly. Indeed, if two agents join differently many queries, their chance to be read falsely at least once is different. Therefore, in non-regular designs, the single agents are not exchangeable anymore and the system's entropy decreases for small n. This additional variation appearing in non-regular designs seems to be the reason why doubly regular designs perform better.

Summary. The proven performance guarantee is uniform in the expected number of agents with hidden bit one. It was found that substantially fewer queries are required using the same algorithm as necessary for exact recovery. Guided by recent findings in group testing, we introduced a doubly regular pooling design. Our formal proof is designed explicitly to this scheme but straight-forward adjustments would yield the same bounds on dense Bernoulli designs or one-sided regular schemes. Furthermore, our empirical analysis reveals that on small systems, designs that allow agents to join queries multiple times are inferior to corresponding designs in which this is not allowed. Moreover, doubly regular designs outperform non-regular designs if the queries are subjected to noise but do not improve upon the number of queries required in the noiseless variant.

One obvious follow-up question is whether the agents' bits could be learned approximately even better with the use of neural networks instead of plain combinatorial constructions. A second open question relates to an extension of the model: in the pooled data problem we assume a query to report the exact number of agents with bit one (up to noise). In *semi-quantitative group testing*, much less information is given: a query outputs one specific value for a whole range of inputs. It is exciting future research whether a similar approach can be carried out under this model.

References

[1] N. R. Adam and J. C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Comput. Surv.*, 21(4):515–556, 1989.

[2] A. Alaoui, A. Ramdas, F. Krzakala, L. Zdeborová, and M. I. Jordan. Decoding from pooled data: Phase transitions of message passing. *IEEE Trans. Information Theory*, 65(1):572–585, 2019.

[3] M. Aldridge. The capacity of bernoulli nonadaptive group testing. *IEEE Trans. Information Theory*, 63(11):7142–7148, 2017.

[4] M. Aldridge, O. Johnson, and J. Scarlett. Improved group testing rates with constant column weight designs. *IEEE International Symposium on Information Theory ISIT*, pages 1381–1385, 2016.

[5] M. Aldridge, O. Johnson, and J. Scarlett. Group testing: An information theory perspective. *Foundations and Trends in Communications and Information Theory*, 15(3–4):196–392, 2019.

[6] R. Ben-Ami, A. Klochendler, M. Seidel, et al. Large-scale implementation of pooled rna extraction and rt-pcr for sars-cov-2 detection. *Clinical Microbiology and Infection*, 26(9):1248–1253, 2020.

[7] C. C. Cao, C. Li, and X. Sun. Quantitative group testing-based overlapping pool sequencing to identify rare variant carriers. *BMC Bioinformatics*, 15:195, 2014.

[8] X. Chen. Concentration inequalities for bounded random vectors, 2013.

[9] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick. Optimal group testing. *Combinatorics, Probability and Computing*, pages 1–38, 2021.

[10] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, A. Wein, and I. Zadik. Statistical and computational phase transitions in group testing. *Proc. 35th Conference on Learning Theory (COLT)*, 178:4764–4781, 2022.

[11] S. D. Constantin and T. Rao. On the theory of binary asymmetric error correcting codes. *Information and Control*, 40(1):20–36, 1979.

[12] I. Dinur and K. Nissim. Revealing information while preserving privacy. *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[13] A. G. Djackov. On a search model of false coins. *Topics in Information Theory. Hungarian Acad. Sci*, 16:163–170, 1975.

[14] D. Du, F. K. Hwang, and F. Hwang. *Combinatorial group testing and its applications*, volume 12. World Scientific, 2000.

[15] A. El Alaoui, A. Ramdas, F. Krzakala, L. Zdeborová, and M. I. Jordan. Decoding from pooled data: Phase transitions of message passing. *IEEE Trans. Inf. Theor.*, 65(1):572–585, 2019. ISSN 0018-9448. doi: 10.1109/TIT.2018.2855698.

[16] U. Feige and A. Lellouche. Quantitative group testing and the rank of random matrices, 2020.
[17] E. Gardner and B. Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983–1994, 1989. doi: 10.1088/0305-4470/22/12/004.
[18] O. Gebhard, M. Hahn-Klimroth, D. Kaaser, and P. Loick. On the parallel reconstruction from pooled data. *Proc. 36th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2022.
[19] O. Gebhard, M. Hahn-Klimroth, D. Kaaser, and P. Loick. Distributed reconstruction of noisy pooled data. *Proc. 42nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2022.
[20] V. Grebinski and G. Kucherov. Optimal reconstruction of graphs under the additive model. *Algorithmica*, 28(1):104–124, 2000.
[21] M. Hahn-Klimroth and N. Müller. Near optimal efficient decoding from pooled data. *Proc. 35th Conference on Learning Theory (COLT)*, 2022.
[22] S. Janson. On concentration of probability. *Combinatorics, Probability and Computing*, 11:2002, 1999.
[23] S. Janson, T. Łuczak, and A. Rucinski. *Random Graphs*. John Wiley & Sons, Inc., Feb. 2000. doi: 10.1002/9781118032718.
[24] K. Joag-Dev and F. Proschan. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983. ISSN 00905364.
[25] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson. Sparse graph codes for non-adaptive quantitative group testing. *IEEE Information Theory Workshop (ITW)*, pages 1–5, 2019.
[26] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson. Non-adaptive quantitative group testing using irregular sparse graph codes. *Proc. 2019 IEEE Allerton*, pages 608–614, 2019.
[27] F. Kong, L. Yuan, Y. F. Zheng, and W. Chen. Automatic liquid handling for life science: A critical review of the current state of the art. *Journal of Laboratory Automation*, 17(3):169–185, 2012.
[28] W. Liang and J. Zou. Neural group testing to accelerate deep learning. *IEEE International Symposium on Information Theory ISIT*, pages 958–963, 2021.
[29] J. P. Martins, R. Santos, and R. Sousa. Testing the maximum by the mean in quantitative group tests. *New Advances in Statistical Modeling and Applications*, pages 55–63, 2014.
[30] J. Noonan and A. Zhigljavsky. Random and quasi-random designs in group testing. *arXiv:2101.06130*, 2021.
[31] J. Scarlett and V. Cevher. Phase transitions in the pooled data problem. *Proc. 30th NEURIPS*, pages 376–384, 2017.
[32] P. Sham, J. S. Bader, I. Craig, M. O'Donovan, and M. Owen. DNA pooling: a tool for large-scale association studies. *Nature Reviews Genetics*, 3:862–871, 2002.
[33] N. Tan, W. Tan, and J. Scarlett. Performance bounds for group testing with doubly-regular designs. *arXiv:2201.03745*, 2022.
[34] C. Wang, Q. Zhao, and C. Chuah. Group testing under sum observations for heavy hitter detection. *Information Theory and Applications Workshop, ITA*, pages

149–153, 2015.

[35] H. Zhou, A. Jiang, and J. Bruck. Nonuniform codes for correcting asymmetric errors in data storage. *IEEE transactions on information theory*, 59(5):2988–3002, 2013.

[36] Y. Zhou, U. Porwal, C. Zhang, H. Q. Ngo, X. Nguyen, C. Ré, and V. Govindaraju. Parallel feature selection inspired by group testing. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014.

maximilian.hahnklimroth@tu-dortmund.de, TU Dortmund University, Germany

dominik.kaaser@tuhh.de, TU Hamburg, Germany

malin.rau@uni-hamburg.de, Universität Hamburg, Germany