Adaptive Estimation in Weighted Group Testing

Jayadev Acharya
EECS, MIT
jayadev@csail.mit.edu

Clément L. Canonne
CS, Columbia University
ccanonne@cs.columbia.edu

Gautam Kamath
EECS, MIT
g@csail.mit.edu

***Abstract*—We consider a generalization of the problem of estimating the support size of a hidden subset S of a universe U from samples. This framework falls under the *group testing* [1] and the *conditional sampling* models [2, 3]. In group testing, for a query set, we are told if it intersects with the set S. We propose a generalization of this problem, where each element has a non-negative weight, and the objective is to estimate the total weight of the universe. In contrast to the regular group testing, we consider stronger access models, where each query outputs an element (with an appropriate probability), and reveals its weight. We show that in this natural generalization of the problem can be solved with only polylogarithmically many queries, and also discuss some lower bounds for the problem.**

Keywords: Group testing, Conditional sampling, Distribution support estimation

I. Introduction

Introduced by Dorfman [1] for the purpose of identifying infected individuals by efficient testing of blood samples, *group testing* has found applications in a number of fields, including biology, experimental design and optimization. At a high level, the objective of (combinatorial) group testing is to identify an unknown subset S of *defective items* from a universe U of n elements (where $|S| = o(n)$), by making as few queries as possible. In the usual setting, a query (also referred to as *test*) consists of a subset $T \subseteq U$, and receives a positive answer if the intersection $T \cap S$ is non-empty – that is, one is told whether the set queried contains at least one defective item.

While $|U|$ queries trivially suffice to determine S, [1] shows that when $|S|$ is small, one can achieve this goal with far fewer queries. Following this work, a long line of research has delved into this problem, tightening the bounds and analyzing some of its variants (see [4–7] for surveys on group testing and various related topics).

Broadly speaking, group testing can be divided into two categories, namely *adaptive* and *non-adaptive* testing. In the former, one is allowed to choose queries based on the previous observations and answers; while the latter enforces that the queries be fixed in advance [5, 8]. Although non-adaptive group testing requires more queries, it allows the protocol to be performed in a distributed fashion. This is a great advantage in situations such as DNA sequencing, where each test can take by itself a significant amount of time. In particular, it is known that for adaptive testing the optimal number of queries scales as $\Theta\left(|S| \log \frac{n}{|S|}\right)$, versus the optimal $\Theta\left(|S|^2 \log n\right)$ for the non-adaptive case [5].

Relation with previous work: Besides the vanilla version of group testing, many variants have been investigated over the last few decades. These include noisy group testing [9, 10], testing with more structured types of queries [11, 12], as well as various types of algorithms ranging from the classic combinatorial to some inspired by "compressive sensing-type" problems.

Most of these variants are still concerned with determining the exact set S. In contrast, in this work we are interested in estimating some function of the set S: specifically, in our model each element of the universe is assigned an unknown non-negative weight. The objective is then to *approximate* the sum W of the weights of all defective elements. A special case of this problem is when all weights are either 0 or 1, and the goal is to determine the support size of non-zero weight elements, i.e. the number of defective items, as in e.g. [13].

We observe that our setup is reminiscent of those of *survey sampling* (see e.g. [14] or [15] for an introduction to this broad field) and *priority sampling* (as defined and studied in [16, 17]). Indeed, in both settings, one is similarly given the task of estimating the total weight of a subset, and has (a type of) sampling access to its elements. We note however two major differences: firstly, in these settings, the task amounts to *designing* a good sampling procedure, either by preprocessing the weights of all elements of the universe or by acting directly on the sampling process. Secondly, and quite crucially, they assume that the subset itself is known beforehand – that is, the set S is provided as input to the estimation procedure, which is then required to estimate its total weight. For these reasons, the techniques used in these areas do not directly apply to our problem.

Motivation: As a concrete example of a scenario where such a question would arise, imagine being in charge of an Internet hub, handling and monitoring on a daily basis the requests and network traffic of millions of clients. Among these, a small portion may be corrupted – their communications originating from a computer virus. Your present concern is to estimate which overall fraction of the traffic data is controlled by these "infected clients", e.g. to decide whether it is time to shutdown this portion of the network. Detecting the trace of the virus in any particular data transmission is not difficult – it usually is given away by a pattern or signature, itself easy to spot. However, analyzing every single such transmission is clearly impossible, as this would require going through huge amounts of data in real-time.

Fortunately, our results show this task is not as hopeless as

it may seem: indeed, by choosing to inspect any particular data transmission at random, one effectively samples them according to their volume (i.e., "weight"); moreover, as the total length of a data flux is usually provided in the headers alongside the data itself, by doing so we obtain the weight of the sampled communication at the same time. This network estimation scenario does therefore fall under our weighted group testing model, and can benefit from the techniques we develop.

Organization: In Section II, we formally define the setting of "weighted group testing" that we shall work in. In Section V, we describe and analyze an (adaptive) algorithm for this generalization of group testing. This algorithm is able to output a good estimate of $W = \sum_{i \in U} w_i$ with only $\operatorname{polylog} n$[1] "labelled conditional queries" (an extension of the basic queries described above). This result is to be compared to the naive approach of Section IV, which yields a $O(|S| \log \frac{n}{|S|})$-query protocol. In particular, the former becomes particularly interesting in the "not-too-sparse" regime, where $|S| = n^\gamma$ for some small constant γ: we discuss this tradeoff and show how to combine both results in Section VI, under a mild assumption on the weights w_i. Finally, in Section VII we draw connections with other work, allowing us to obtain lower bounds for the weighted group testing problem in both the non-adaptive and adaptive settings.

II. Problem Statement

Consider a setting with n individuals, represented by $[n] = \{1, \dots, n\}$, where the i-th individual has a corresponding weight $w_i \geq 0$. For a subset $S \subseteq [n]$, let $W_S = \sum_{i \in S} w_i$. The experiment is as follows. Each test takes as input a set $T \subseteq [n]$ and returns:

- an element $i \in T$ with probability w_i / W_T (or uniformly in T if $W_T = 0$), along with
- its weight w_i.

The objective is to estimate $W \stackrel{\text{def}}{=} \sum_{i=1}^n w_i$. A special case of the problem is estimating the support size of a ground set $S \subseteq [n]$. In this case $w_i = 1$ for $i \in S$, and $w_i = 0$ otherwise.

More specifically, the problem we address is as follows. Given a parameter $\varepsilon > 0$, the experimenter needs to come up with a protocol that allows her to obtain an estimate $\hat{W}$ which, with probability $2/3$, satisfies $(1-\varepsilon)W \leq \hat{W} \leq (1+\varepsilon)W$. Furthermore, she must do so while performing as few tests as possible. (Note that in this setting, the tests are allowed to be *adaptive*, that is to be chosen taking into account the outcome of previous tests.)

III. Preliminaries

Let $D(T) = W_T / W$ be the probability of observing an element from T when we sample from the entire universe $[n]$. Hereafter, we write D_T for the conditional distribution induced by a probability distribution D on a set T, noting that D_T is only defined whenever $D(T) > 0$. In the setting above, we say that D is *induced* by the weights w_i if $D(i) = \frac{w_i}{W}$ for all $i \in [n]$. Given an (unknown) probability distribution D over $[n]$ (induced by a set of weights $(w_i)_{i \in [n]}$), a *labelled conditional query* to D consists of a subset $T \subseteq [n]$, and is answered as follows:

- if $D(T) > 0$, then a random element x is sampled from D_T independently of any previous query;
- otherwise (if $D(T) = 0$, i.e. $W_T = 0$), a uniformly distributed element x in T is drawn (again independently from previous queries).

The pair (x, w_x) is then returned.

Note that this is in contrast with the weaker usual access model of group testing, where the only type of queries allowed is, given a subset $T \subseteq [n]$, the question *"is $D(T)$ non-zero?"*. In particular, given labelled conditional queries one can determine any specific weight w_i by querying the set $\{i\}$, getting as answer the pair (i, w_i).

Finally, we shall use the following Chernoff bound.

Lemma 1: [18] Suppose $X_1, \dots, X_n$ are independent random variables taking values in $[0,1]$, and set $X = X_1 + \dots + X_n$. Let $\mu = \mathbb{E}[X]$. Then for $0 \leq \varepsilon \leq 1$,

$$\mathbb{P}\left[X \in ((1-\varepsilon)\mu, (1+\varepsilon)\mu)\right] \geq 1 - 2 \cdot \exp\left(-\frac{\varepsilon^2 \mu}{3}\right).$$

IV. Warmup: an easy $O(|S| \log n)$-query algorithm for exact computation of W

Suppose only a subset S of the elements have a non-zero weight. In this case, it is not difficult to see that, using known machinery and results from group testing [5], one can identify with $O(|S| \log \frac{n}{|S|})$ queries the *exact* hidden subset of elements S. From there, by repeatedly making conditional queries on $\{x\}$ for each $x \in S$, it is possible to retrieve each non-zero w_i and compute the *exact* value of W, at the price of a (negligible) additional $|S|$ queries.

However appealing and simple this approach may be, it quickly becomes impractical as $|S|$ grows – in particular, whenever $|S| = n^{\Omega(1)}$, and in particular $S = [n]$, when all the elements have non-zero weights. As we shall see momentarily, in this case (which arguably occurs in many situations) it is possible to obtain a good approximation of W using far fewer queries: namely, only $\operatorname{polylog} n$.

V. An $\tilde{O}(\log^3 n / \varepsilon^2)$-query algorithm

In this section, we describe an approach that leverages the ability to perform *conditional queries* on the distribution induced by the weights, i.e. $D(i) = \frac{w_i}{W}$. At a high-level, our algorithm derives from the following observation: if we can obtain, for *any* $i \in S$, a (good enough) approximation $\hat{p}_i$ of $D(i)$, then the value $\widehat{W} \stackrel{\text{def}}{=} \frac{w_i}{\hat{p}_i}$ will in turn be an accurate estimate of W.

In the spirit of [3][2], it turns out one can indeed obtain (modulo some technical details) such a good $\hat{p}_i$ for a "nice" i, by making $\tilde{O}(\log^3 n / \varepsilon^2)$ (adaptive) conditional queries to D.

[1] Hereafter, $\operatorname{polylog} n$ denotes a polynomial in $\log n$.

[2] See e.g. the proofs of Theorem 4 and Theorem 15, which adopt a related "binary descent" approach.

At a very high level, the idea is, given any $i \in [n]$, to perform a "biased-towards-the-heaviest binary descent" by splitting the current domain T (originally $T = [n]$) in two halves T_1 and T_2 and estimating the ratio $D(T_1)/D(T_2)$; before recursing on T_1, until the point where T is a singleton $\{i\}$. Provided all estimates were accurate enough, it then suffices to multiply them to obtain a multiplicative approximation of $D(i)/D([n]) = D(i)$.

The caveat here is that estimating the ratio may only be efficient if T_1 and T_2 have comparable weights, or at least that $D(T_1)/D(T_2) = \Omega(1)$; hence the need for, at each stage, a suitable partitioning of T to ensure this happens with high probability. Specifically, taking a constant number of samples from the conditional distribution of D on T we can ensure (by the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality [19, 20]) that one of the following happens:

(a) we find an element $x \in T$ such that $D(x)/D(T) = \Omega(1)$; or
(b) we get a partition T_1 and T_2 of T such that $D(T_1)/D(T_2) = \Omega(1)$ and $|T_1| < |S|/2$.

In the first case, we are done, as estimating this ratio $D(x)/D(T)$ and backtracking the recursion yields a good estimate of $D(x)$, as wished; while in the second we can efficiently get an approximation of $D(T_1)/D(T_2)$ before recursing on T_1 (which has at most half as many elements as R, making sure the recursion will end in at most $\log n$ steps). We give the algorithm in Algorithm 1, proving the following theorem:

Theorem 2 (Main upper bound): There exists an algorithm which, given labelled conditional query access to a hidden subset $S \subseteq [n]$ as defined in Section II, satisfies the following. On input $\varepsilon \in (0, 1]$, it makes $\tilde{O}(\frac{\log^3 n}{\varepsilon^2})$ adaptive queries, and outputs an estimate $\widehat{W}$ which, with probability at least $2/3$, is within a factor $(1 + \varepsilon)$ of W_S.

Proof: We first argue correctness of Algorithm 1, before establishing its query complexity.

Correctness: Observe that, by the choice of T, the recursion ends after at most $\log n$ stages, as the size of R drops by a factor at least 2 at each iteration. By the DKW inequality, the choice of constant in the $O(\cdot)$ notation and a union bound over all $\log n$ stages, we get that for each choice of R in the execution of the algorithm, $\sup_{i \le j} |\hat{D}(\{i, \dots, j\}) - D_R(\{i, \dots, j\})| \le 1/10$, except with probability at most $1/20$. We hereafter condition on this.

This in particular implies that $D_R(T)$ is within $\pm 1/10$ of $\hat{D}(T)$; since our preliminary check ensured that $\hat{D}(c) < 1/10$, this in turn guarantees that $D_R(T) \in [9/20, 12/20]$. But then, a direct application of the Chernoff bounds (along with our choice of q) yields that, with probability at least $1 - \delta$, the estimate ρ_T we compute satisfies $D_R(T)/\rho_T \in [1 - \varepsilon', 1 + \varepsilon']$. Similarly, if a "heavy" element $x \in R$ is found (and an estimate ρ_x is computed), we have that $D_R(T)/\rho_x \in [1 - \varepsilon', 1 + \varepsilon']$ with probability at least $1 - \delta$. Overall, by a union bound and our choice of δ, all estimates computed during the course of the algorithm are correct except with probability at most $1/20$. Again, we condition on this event.

Algorithm 1 BINARY-DESCENT-ESTIMATION

Set $\varepsilon' = \frac{\varepsilon}{2 \log n}$, $\delta = \frac{1}{20 \log n}$, $q = O\big(\frac{1}{\varepsilon'^2} \log \frac{1}{\delta}\big)$
Initialize $R \leftarrow [n]$, $\hat{W} \leftarrow 1$
while $|R| > 1$ **do**
  Obtain $O(\log(1/\delta))$ queries from $R = \{a, \dots, b\}$, let $\hat{D}$ be the empirical probability distribution they define.
  if $\exists x \in R$ s.t. $\hat{D}(x) > 1/10$ **then**
    Make q fresh queries from R, getting q elements.
    Compute ρ_x, the estimate of $D_R(x)$ defined as the frequency of x among them.
    Set $\hat{W} \leftarrow \hat{W} \cdot \rho_x^{-1}$, $R \leftarrow \{x\}$ and exit the loop.
  else
    Let c be the min. element s.t. $\hat{D}(\{a, \dots, c\}) \ge 1/2$.
    Let $T = \{a, \dots, c\}$ if $|\{a, \dots, c\}| \le |S|/2$ and $T = \{c + 1, \dots, b\}$ otherwise.
    Make q fresh queries from R, getting q elements.
    Compute ρ_T, the estimate of $D_R(T)$ defined as the fraction of elements from T among them.
    Set $\hat{W} \leftarrow \hat{W} \cdot \rho_T^{-1}$ and $R \leftarrow T$.
  end if
end while
return $\hat{W} \leftarrow \hat{W} \cdot w_i$, where $R = \{i\}$.

Putting it all together, we get that, with probability at least $18/20 = 9/10$ (and writing $[n] = R_1 \supseteq R_2 \cdots \supseteq R_k = \{i\}$ for the sets computed during the execution of Algorithm 1) the estimate $\hat{W}$ we output satisfies the following.

$$\begin{aligned}
\hat{W} = w_i \cdot \prod_{\ell=1}^{k} \rho_{R_\ell}^{-1} &\in w_i \cdot [(1-\varepsilon')^k, (1+\varepsilon')^k] \prod_{\ell=2}^{k} \frac{1}{D_{R_{\ell-1}}(R_\ell)} \\
&\in w_i \cdot [(1-\varepsilon')^k, (1+\varepsilon')^k] \prod_{\ell=2}^{k} \frac{D(R_\ell)}{D(R_{\ell-1})} \\
&\in [(1-\varepsilon')^k, (1+\varepsilon')^k] \frac{w_i}{D(\{i\})} \\
&= [(1-\varepsilon')^k, (1+\varepsilon')^k] W \\
&\in [(1-\varepsilon')^{\log n}, (1+\varepsilon')^{\log n}] W \\
&\subseteq [1-\varepsilon, 1+\varepsilon] W,
\end{aligned}$$

where the last inclusion comes from our choice of $\varepsilon' = \frac{\varepsilon}{2 \log n}$. Therefore, with probability at least $2/3$, the algorithm outputs a value which is within a multiplicative $(1 \pm \varepsilon)$ of W, as desired.

Query complexity: It is straightforward to see that the query complexity at each of the (at most) $\log n$ stages is dominated by the q queries to R. Thus, the total number of queries is at most $O(q \cdot \log n) = O(\frac{\log n}{\varepsilon'^2} \log \log n)$, giving the claimed $\tilde{O}(\frac{\log^3 n}{\varepsilon^2})$. ∎

A. Boosting the success probability

The algorithm above has a probability of success 2/3. It is possible to achieve any probability of error $p_e > 0$ with

$\tilde{O}(\frac{\log^3 n}{\varepsilon^2}\log\frac{1}{p_e})$ using the following classic trick. Repeat Algorithm 1 $O(\log\frac{1}{p_e})$ times and output the median of all W's thus generated. It can then be shown that the median does not fall in the specified interval with a probability that drops exponentially with the number of experiments.

VI. Best of both worlds: choosing the best algorithm

The two algorithms we described, respectively in Section IV and Section V, yield different guarantees; and it is not clear *a priori* which one to choose, as which of the two is the best option depends on the – unknown – size of the hidden set S.

However, it is possible to leverage one of the results of Acharya, Canonne, and Kamath [21] to efficiently perform this choice as long as we are promised the non-zero weights are not too small, and always opt for the best of our two approaches. The idea is to first compute a crude estimate of $|S|$, up to say a factor two; and according to this estimate choose between the $O(|S|\log\frac{n}{|S|})$- and the $\tilde{O}(\log^3 n/\varepsilon^2)$-query algorithm. Of course, for this idea to be any good one must be able to perform this crude approximation with a very small number of queries: fortunately, this is the case, as shown in [21]:

Theorem 3 (Theorem 1.2 of [21], restated): There exists an (adaptive) algorithm which, given conditional access to an unknown distribution D on U which has minimum non-zero probability $1/n$, makes $\tilde{O}(\log\log n)$ queries to the oracle and outputs a value $\tilde{\omega}$ such that the following holds. With probability at least $2/3$, $\tilde{\omega}\in[\frac{1}{2}\cdot\omega, 2\cdot\omega]$, where $\omega=|\operatorname{supp} D|$.

Loosely speaking, their algorithm works by performing a "doubly exponential search" to find a – very – coarse candidate size, and then improve this estimate by a binary search on the smaller, narrowed-down range. The crux is to efficiently determine at each step whether the current guess $\tilde{\omega}$ of the size is precise enough: this is done by making a query on a random set $R_{\tilde{\omega}}$ of cardinality roughly $n/\tilde{\omega}$. If $\tilde{\omega}$ is good enough, this set is likely to intersect S on $\Theta(1)$ many points; however, if $\tilde{\omega}\ll|S|$, then $R_{\tilde{\omega}}$ will not intersect S at all, and the search goes on with the next candidate value.

Now, by applying the above theorem as a black box, we can *(a)* get a 2-estimate ω of $|S|$, and *(b)* choose which of the two algorithm apply (depending on whether $\omega\log\frac{n}{\omega}\gg\frac{\log^3 n}{\varepsilon^2}$). This gives us the following overall result:

Theorem 4: Provided all non-zero w_i's are guaranteed to be at least $1/n$, one can solve the estimation problem of Section II with $\tilde{O}(\min(|S|\log\frac{n}{|S|},\frac{\log^3 n}{\varepsilon^2}))$ adaptive labelled conditional queries.

VII. Lower bounds and discussion

In this section, we give lower bounds on the query complexity of estimating W_S.

We get these lower bounds for the special case of estimating the support size, i.e. when $w_i=1$ for all $i\in S$. Without loss of generality, suppose n is a power of 2. The construction on which these lower bounds are based is as follows.

- Let t be picked uniformly from $\{1,2,\dots,\log n\}$,
- S is a random set of size 2^t from all such sets,
- Set $w_i=1$ for $i\in S$, and 0 otherwise.

Before discussing lower bounds on the precise, adaptive, version of our problem, we consider their counterpart for the weaker *non-adaptive* query model.

A. Non-adaptive lower bounds

Suppose, that instead of choosing a new query set at each stage that is dependent on the output of the previous stage, we are asked to design queries independently without this knowledge. Even though this is a weaker model for specifying the query sets, our querying itself is more powerful than those considered in group testing and support estimation problems, namely we are provided not only whether the query set intersects with S but also provided a random element from the intersection. This model was considered in [13]; however, the authors there only provide upper bounds for this particular problem, leaving lower bounds as an open question.

This problem was partially resolved in [21, Theorem 4.1], which states that in the construction above, estimating the support size up to a factor $\log n$ requires $\Omega\Big(\frac{\log n}{\log\log n}\Big)$ queries. By adapting their argument for a smaller (constant) value of the approximation factor, the same techniques yield the following:

Theorem 5: Any non-adaptive algorithm for estimating W_S up to a factor 2 requires at least $\Omega(\log n)$ queries.

Indeed, in the full version of [21] it is shown that, for the construction described above, the outcomes for values of $|S|$ differing by a factor of 2 are *indistinguishable*; because of the setting of the weight we consider, $W_S=|S|$, which immediately implies that W_S cannot be estimated either.

B. Adaptive lower bound

It is also possible to adapt the algorithms to obtain bounds on the number of adaptive queries to estimate W. The construction is the same as for the non-adaptive setting. In this case, the following is a consequence of Lemma 7.3.7 of [2].

Theorem 6: Any algorithm to estimate W up to a factor 2 requires $\Omega(\sqrt{\log\log n})$ queries.

VIII. Conclusion

We consider generalizations of the group testing problem to estimate the total *weight* of the population. We show that even in this setting one can design efficient tests, by giving a procedure that only performs a poly-logarithmic number of queries; and further also provide a non-trivial lower bound. An interesting open question that remains is to determine the precise query complexity, either by designing more efficient algorithms or establishing stronger lower bounds. Some algorithms very recently proposed in [22] might provide faster algorithms for this set-up. We believe that, in practical situations, the extra cost of implementing our more complex query access can be offset by the significant savings in the number of queries required, just as in group testing.

REFERENCES

[1] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 12 1943. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177731363

[2] S. Chakraborty, E. Fischer, Y. Goldhirsh, and A. Matsliah, "On the power of conditional samples in distribution testing," in *Proceedings of ITCS*. New York, NY, USA: ACM, 2013, pp. 561–580. [Online]. Available: http://doi.acm.org/10.1145/2422436.2422497

[3] C. L. Canonne, D. Ron, and R. A. Servedio, "Testing probability distributions using conditional samples," *CoRR*, no. abs/1211.2664, Nov. 2012.

[4] D. Balding, W. Bruno, D. Torney, and E. Knill, "A comparative survey of non-adaptive pooling designs," in *Genetic Mapping and DNA Sequencing*, ser. The IMA Volumes in Mathematics and its Applications, T. Speed and M. Waterman, Eds. Springer New York, 1996, vol. 81, pp. 133–154. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-0751-1_8

[5] D. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*, ser. Applied Mathematics. World Scientific, 2000. [Online]. Available: http://books.google.com/books?id=KW5-CyUUOggC

[6] H. Q. Ngo and D.-Z. Du, "A Survey on Combinatorial Group Testing Algorithms with Applications to DNA Library Screening," in *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2000.

[7] H.-B. Chen and F. K. Hwang, "A survey on nonadaptive group testing algorithms through the angle of decoding," *Journal of Combinatorial Optimization*, vol. 15, no. 1, pp. 49–59, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10878-007-9083-3

[8] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing: Explicit bounds and novel algorithms," *CoRR*, vol. abs/1202.0206, 2012. [Online]. Available: http://arxiv.org/abs/1202.0206

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

[10] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi, "GROTESQUE: noisy group testing (quick and efficient)," *CoRR*, vol. abs/1307.2811, 2013. [Online]. Available: http://arxiv.org/abs/1307.2811

[11] F. K. Hwang, "Three versions of a group testing game," *SIAM Journal on Algebraic Discrete Methods*, vol. 5, no. 2, pp. 145–153, 1984. [Online]. Available: http://dx.doi.org/10.1137/0605016

[12] T. Gerzen, "On a group testing problem: Characterization of graphs with 2-complexity c2 and maximum number of edges," *Discrete Appl. Math.*, vol. 159, no. 17, pp. 2058–2068, October 2011. [Online]. Available: http://dx.doi.org/10.1016/j.dam.2011.06.026

[13] D. Ron and G. Tsur, "The power of an example: Hidden set size approximation using group queries and conditional sampling," *CoRR*, no. abs/1404.5568, 2014.

[14] G. Kalton, *Introduction to Survey Sampling*, ser. Quantitative Applications in the Social Sciences Series. SAGE PUBN Incorporated, 1983. [Online]. Available: http://books.google.com/books?id=P03woAEACAAJ

[15] C. Särndal, B. Swensson, and J. Wretman, *Model Assisted Survey Sampling*, ser. Springer series in statistics. Springer-Verlag, 1992. [Online]. Available: http://books.google.com/books?id=MWCzngEACAAJ

[16] N. Alon, N. Duffield, C. Lund, and M. Thorup, "Estimating arbitrary subset sums with few probes," in *Proceedings of the Twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '05. New York, NY, USA: ACM, 2005, pp. 317–325. [Online]. Available: http://doi.acm.org/10.1145/1065167.1065209

[17] N. Duffield, C. Lund, and M. Thorup, "Priority sampling for estimation of arbitrary subset sums," *Journal of the ACM*, vol. 54, no. 6, December 2007. [Online]. Available: http://doi.acm.org/10.1145/1314690.1314696

[18] M. Mitzenmacher and E. Upfal, *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge Univ. Press, 2005.

[19] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 642–669, 09 1956. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177728174

[20] P. Massart, "The tight constant in the Dvoretzky–Kiefer–Wolfowitz inequality," *The Annals of Probability*, vol. 18, no. 3, pp. 1269–1283, 07 1990. [Online]. Available: http://dx.doi.org/10.1214/aop/1176990746

[21] J. Acharya, C. L. Canonne, and G. Kamath, "A chasm between identity and equivalence testing with conditional queries," *CoRR*, vol. abs/1411.7346, 2014. [Online]. Available: http://arxiv.org/abs/1411.7346

[22] M. Falahatgar, A. Jafarpour, A. Orlitsky, V. Pichapathi, and A. T. Suresh, "Faster algorithms for testing under conditional sampling," *CoRR*, vol. abs/1504.04103, 2015. [Online]. Available: http://arxiv.org/abs/1504.04103