Near-Optimal Compressive Binary Search

Matthew L. Malloy
Electrical and Computer Engineering
University of Wisconsin-Madison
Email: mmalloy@wisc.edu

Robert D. Nowak
Electrical and Computer Engineering
University of Wisconsin-Madison
Email: nowak@ece.wisc.edu

Abstract—**We propose a simple modification to the recently proposed *compressive binary search* [1]. The modification removes an unnecessary and suboptimal factor of $\log \log n$ from the SNR requirement, making the procedure optimal (up to a small constant). Simulations show that the new procedure performs significantly better in practice as well. We also contrast this problem with the well known *noisy binary search* problem.**

I. Introduction

The recently proposed *compressive binary search* (CBS) algorithm [1] aims to determine the location of a single non-zero entry in a vector $\boldsymbol{x} \in \mathbb{R}^n$ using m adaptive linear projections of the form

$$y_i = \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle + z_i \qquad i = 1, ..., m$$

where $\boldsymbol{a}_i \in \mathbb{R}^n$ are sensing vectors with $||\boldsymbol{a}_i|| = 1$ and z_i are i.i.d. $\mathcal{N}(0,1)$. The CBS algorithm proposed in [1] finds the non-zero entry with vanishing probability of error as n gets large provided $\mu \geq C\sqrt{(n/m)\log \log_2 n}$ where μ is the amplitude of the non-zero entry. A bit more precisely, Theorem 1 in [1] states that $\mathbb{P}_e \leq \delta$ provided

$$\mu \geq \sqrt{\frac{8n}{m}\left(\log \frac{1}{2\delta} + \log \log_2 n\right)} \tag{1}$$

where $\mathbb{P}_e$ is the probability the procedure fails to return an index corresponding to the non-zero entry. The dependence on δ is to be expected, but the authors of [1] rightly question whether the $\log \log_2 n$ term is needed. The main contribution of this paper is a simple modification of the algorithm proposed in [1] which eliminates this unnecessary and suboptimal dependence on n.

II. Main Result

The CBS algorithm proposed in [1] operates as follows. The algorithm consists of $s_0 = \log_2 n$ steps (as in [1] we assume n is dyadic). At each step of the algorithm, indexed by s, measurements are taken on progressively smaller dyadic subintervals of $\{1, \ldots, n\}$. The sensing vectors are discrete Haar wavelets, and the sign of each measurement gives an indication of whether the non-zero entry is in the right or left half-interval of the wavelet's support. The CBS algorithm is outlined in Fig. 1.

```
Compressive Binary Search (CBS)
number of steps: $s_0 := \log_2 n$
measurements per step: $m_s$ , with $\sum_{s=1}^{s_0} m_s \leq m$
initial support: $J_0^{(1)} := \{1, \ldots, n\}$
for $s = 1, \ldots, s_0$
1) split: $J_1^{(s)}$ and $J_2^{(s)}$, left and right subinterval of $J_0^{(s)}$
2) sensing vector: $\boldsymbol{u}^{(s)} = 2^{-(s_0-s+1)/2}$ on $J_1^{(s)}$,
    $\boldsymbol{u}^{(s)} = -2^{-(s_0-s+1)/2}$ on $J_2^{(s)}$, and 0 otherwise
3) measure: $y_i^{(s)} = \langle \boldsymbol{u}^{(s)}, \boldsymbol{x} \rangle + z_i^{(s)}$,
    $z_i^{(s)} \overset{iid}{\sim} \mathcal{N}(0,1), \; i = 1, \ldots, m_s$
4) update support: $J_0^{(s+1)} = J_1^{(s)}$ if $\sum_{i=1}^{m_s} y_i^{(s)} > 0$
    and $J_0^{(s+1)} = J_2^{(s)}$ otherwise
end
output: $J_0^{(s_0+1)}$ (a single index)
```

Fig. 1. Compressive Binary Search algorithm.

Since the sensing vectors have unit norm, the magnitude of the non-zero entries of the sensing vectors grow as the algorithm proceeds from coarse-to-fine wavelets. Consequently, the SNR of the measurements grows exponentially as the procedure progresses. In [1], the authors divide the m measurements among the steps as follows. For $s = 1, \ldots, s_0$

$$m_s := \tilde{m}_s + 1, \qquad \tilde{m}_s = \left\lfloor (m - s_0)\, 2^{-s} \right\rfloor$$

It is easy to check that $\sum_{s=1}^{s_0} m_s \leq m$. This allocation roughly equalizes the SNR at each step. If μ satisfies (1), then the CBS algorithm is guaranteed to have a probability of error of at most δ/s_0 at each step (and the union bound yields an overall probability of error of at most δ).

Here, instead, we allocate the m measurements as follows:

$$m_s := \tilde{m}_s + 1, \qquad \tilde{m}_s = \left\lfloor (m - s_0)\; s\; 2^{-(s+1)} \right\rfloor .$$

Again, it is easily verified that $\sum_{s=1}^{s_0} m_s \leq m$. With this allocation, the probability of error is at most a constant times 2^{-s} at each step; i.e., the probability of error decreases exponentially over the steps, rather than remaining constant as above. This simple modification is enough to eliminate the $\log \log_2 n$ term in the bound in [1].

Theorem 1. *If $m \geq 2\log_2 n$ and*

$$m_s := \tilde{m}_s + 1, \qquad \tilde{m}_s = \left\lfloor (m - s_0)\; s\; 2^{-(s+1)} \right\rfloor \tag{2}$$

then $\sum_{s=1}^{s_0} m_s \leq m$ and the CBS algorithm succeeds with $\mathbb{P}_e \leq \delta$ provided magnitude of the non-zero entry satisfies

$$\mu \geq \sqrt{\frac{16n}{m}\log\left(\frac{1}{2\delta} + 1\right)}. \tag{3}$$

Proof: The total measurement budget satisfies

$$\sum_{s=1}^{s_0} m_s = s_0 + \sum_{s=1}^{s_0} \tilde{m}_s \leq s_0 + (m - s_0) \sum_{s=1}^{s_0}\ s\ 2^{-(s+1)} \leq m,$$

since $\sum_{s=1}^{s_0} s2^{-(s+1)} \leq 1$. By the union bound and a Gaussian tail bound, the total error probability satisfies

$$\mathbb{P}_e \quad \leq \quad \frac{1}{2} \sum_{s=1}^{s_0} \exp\left(- \frac{m_s \mu^2 2^s}{4n} \right).$$

Since $m_s \geq (m - s_0)s2^{-(s+1)}$ and $m \geq 2 \log_2 n = 2s_0$, we conclude $m_s \geq \frac{m}{2} s2^{-(s+1)}$ and thus

$$\mathbb{P}_e \quad \leq \quad \frac{1}{2} \sum_{s=1}^{s_0} \exp\left(- \frac{ms\mu^2}{16n} \right).$$

Letting $\mu \geq \sqrt{\frac{16n}{m} \log\left(\frac{1}{2\delta} + 1\right)}$ yields

$$\mathbb{P}_e \quad \leq \quad \frac{1}{2} \sum_{s=1}^{s_0} \left(\frac{1}{2\delta} + 1 \right)^{-s} \quad \leq \quad \delta \ .$$

■

Remark 1. The authors of [1] also show that no procedure can succeed at the compressive binary search problem with probability greater than $1/2$ if $\mu \leq \sqrt{n/m}$. Using the new allocation of measurements, the CBS algorithm succeeds with probability greater than $1/2$ if $\mu \geq \sqrt{16 \log(2) n/m}$, within a small constant factor of the lower bound.

Remark 2. The CBS problem is closely related to the so-called *noisy binary search* problem [2]–[5]. Noisy binary search addresses a version of the classic binary search problem with binary noise. If the SNR is equal in each step of CBS (as in [1]), then it is equivalent to the "naive" noisy binary search algorithm discussed in [5], which also suffers from the suboptimal $\log \log_2 n$ factor. More sophisticated algorithms such as Horstein's algorithm [2]–[4] and binary search with backtracking [5] are optimal to within constants. The CBS problem, however, is different from noisy binary search in that more localized measurements (or "queries") are more reliable. Because of this unique feature, the optimal algorithms for noisy binary search do not yield optimal solutions for the CBS problem. Instead, all that is needed to eliminate the $\log \log_2 n$ factor is a measurement allocation that properly exploits the fact that more localized measurements have a larger SNR.

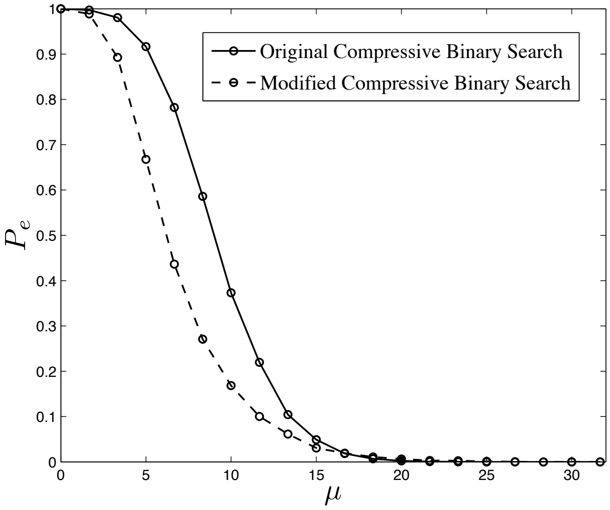

Fig. 2. Numerical Simulation. Empirical performance of CBS (both original and modified) as a function of μ for $n = 4096$ and $m = 256$. 10,000 trials.

References

[1] M. A. Davenport and E. Arias-Castro, "Compressive binary search," *ArXiv e-prints*, Feb. 2012. [Online]. Available: http://arxiv.org/abs/1202.0937

[2] M. Horstein, "Sequential decoding using noiseless feedback," *IEEE Trans. Info. Theory*, vol. 9, no. 3, pp. 136–143, 1963.

[3] M. V. Burnashev and K. S. Zigangirov, "An interval estimation problem for controlled observations," *Problems in Information Transmission*, vol. 10, pp. 223–231, 1974.

[4] R. Nowak, "The geometry of generalized binary search," *Information Theory, IEEE Transactions on*, vol. 57, no. 12, pp. 7893 –7906, dec. 2011.

[5] R. Karp and R. Kleinberg, "Noisy binary search and its applications," in *ACM-SIAM Symposium on Discrete Algorithms (2007)*, pp. 881–890.