Notes on group testing

AIMS Rwanda

June 18, 2022

1 Introduction

Let $A \in \{0,1\}^{M \times N}$ be a design matrix with M measurements (pools or slices) and N individuals. Each entry in A is either 1 or 0, i.e., $a_{ji} = 1$ if pool number j contains individual i and zero otherwise, for $j = 1, 2, \ldots M$ and $i = 1, 2, \ldots N$. Let $y \in \{0,1\}^M$ be the signs of the measurements

$$y_j = \begin{cases} 1, & \text{if} \quad y_j \geq \tau \\ 0, & \text{if} \quad y_j \leq \tau \end{cases}$$

for a predefined threshold τ and we want to recover x such that

$$Ax = y$$

where x is $N \times 1$ s-sparse vector.

The above setting is an instance of 1-bit Compressed Sensing problem where each measurement is represented by only a single bit, i.e., the measurement's sign, instead of a real value. The sparse vector x can be approximately recovered as follows:

$$\begin{aligned} \bar{x} \quad &= \quad \min_{x} \|x\|_1 \\ \text{s.t} \quad & \quad y = sign(Ax) \\ \text{and} \quad & \quad \|Ax\|_1 = c, \quad \text{for } c > 0, \end{aligned} \tag{1}$$

provided that the design matrix A satisfies the conditions required in the traditional CS, i.e., Restricted Isometry Property (RIP) or mutual coherence up to desired level of sparsity s that is known in advance. The 1-norm produces sparse solution, the first constraint guarantees consistency of sign between the measurements and corresponding solution, and the second constraint ensures that such solution will be non-trivial.

In the literature of 1-bit Compressed Sensing there are several algorithms for solving (1), however because of binary-valued nature of entries in the design matrix A, I found that the Binary Iterative Hard Threshold (BIHT) algorithm is robust and accurate. The only information required in advance is the level of sparsity k, which can be deduced approximately from an estimate of the disease's prevalence level p, i.e., $k \approx p * N$.

In BIHT, you initially set x_0 to be the zero vector and then iteratively perform the following program:

$$\begin{aligned} x \quad &= \quad x_{i-1} + \delta * A^T(y - sign(Ax)) \\ x_i \quad &= \quad \Gamma(x_i, k) \end{aligned} \tag{2}$$

where δ is constant for gradient step-size and Γ is hard thresholding operator, i.e., sets all but the greatest k absolute value entries of x to 0. The program continues for sufficient iteration count or approximation is consistent with y.