

Non-adaptive Algorithms

Definition A group testing algorithm is non-adaptive if all tests must be specified without knowing the outcomes

of other tests.

Based on this nature, a non-adaptive algorithm can be represented by an $m \times n$ (0,1)-matrix if its tests are

specified for a group testing on n items. Such a

matrix is also known as a pooling design. The example

we presented in 1' is a 9×12 (0,1)-matrix which can

be considered as a pooling design using 9 tests for 12 items.

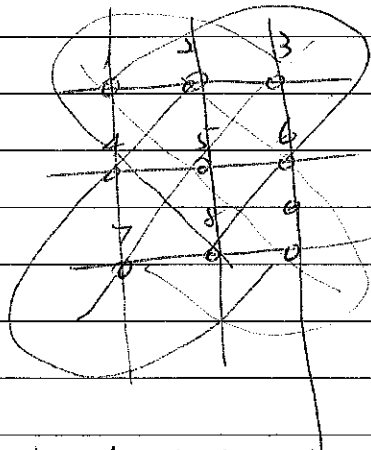
In general, let M be as follows:

$$M: \begin{matrix} & C_1, C_2 & \dots & C_n \\ \begin{matrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{matrix} & \left[\begin{array}{c} M(i,j) = 1 \text{ if the } j\text{th} \\ \text{item is involved in} \\ \text{the } i\text{th test; and} \\ M(i,j) = 0 \text{ otherwise} \end{array} \right. & & \end{matrix}$$

(*) The j th item will be represented as a set C_j where $C_j = \{i \mid \text{the } i\text{th test the } j\text{th item is involved in the}\}$.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
r_1	1	0	0	1	0	0	1	0	0	1	0	0
r_2	1	0	0	0	1	0	0	1	0	0	1	0
r_3	1	0	0	0	0	1	0	0	1	0	0	1
$-r_4$	0	1	0	1	0	0	0	0	1	0	1	0
$-r_5$	0	1	0	0	1	0	1	0	0	0	0	1
r_6	0	1	0	0	0	1	0	1	0	1	0	0
r_7	0	0	1	1	0	0	0	1	0	0	0	1
r_8	0	0	1	0	1	0	0	0	1	1	0	0
r_9	0	0	1	0	0	1	1	0	0	0	1	0

9x12



(o) Each test (row) is also known as a pool.

(o) The whole collection of tests is a pooling design.

(*) In fact, a pooling design can be obtained by a combinatorial design.

(o) $(X, B), B \subseteq 2^X$.

eg.

$$M: \begin{matrix} & C_1 & C_2 & C_3 & C_4 & C_5 \\ \begin{matrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$

$$C_1 = \{3, 4\}, C_2 = \{1, 3\}, C_3 = \{1, 2, 4\}, C_4 = \{3\}, C_5 = \{1, 2, 3, 4\}$$

(*) Since the items are represented by sets, combining items

together can be "seen" as the union of sets.

Definition (d -separable)

M is called d -separable if the unions of d columns are all distinct. If the unions of at most d columns are all distinct, then the matrix is called \bar{d} -separable.

Definition (d -disjunct)

M is called d -disjunct if any column does not contain in a union of d (or less) other columns. Equivalently, given any $d+1$ distinct columns of M , $C_1, C_2, \dots, C_d, C_{d+1}$, $C_{d+1} \not\subseteq \bigcup_{i=1}^d C_i$.

Fact 1. d -disjunct $\Rightarrow \bar{d}$ -separable $\Rightarrow d$ -separable

Fact 2. $\bar{d+1}$ -separable $\Rightarrow d$ -disjunct

Proof. Consider C_1, C_2, \dots, C_{d+1} . By definition $\bigcup_{i=1}^d C_i \neq \bigcup_{j=1}^{d+1} C_j$.

This implies that $C_{d+1} \not\subseteq \bigcup_{i=1}^d C_i$. ■

Following a non-adaptive algorithm, the m tests will be done and an outcome vector $(y_1, y_2, \dots, y_m)^t$ is obtained. We are aiming at determining the d positive items by using the knowledge on the collection of outcome vectors.

(*) If there are d positive items in a set of n items, then there are $\binom{n}{d}$ different possible combinations.

(*) In order to tell the correct answer, we need at least $\binom{n}{d}$ distinct outcome vectors. Therefore, the minimal requirement of decoding "successfully" is: $2^m \geq \binom{n}{d}$, i.e., $m \geq \lceil \log_2 \binom{n}{d} \rceil$.

Again, this is the informatic lower bound.

(*) If $2^m \geq \binom{n}{d}$, then it's possible to find a d -separable matrix which is an $m \times n$ (0,1)-matrix. So, far, for general n , finding such a matrix with $m \approx \log_2 \binom{n}{d}$ is still far from solved.

Solving a (d, n) -problem by using non-adaptive algorithm
(M : pooling design)

can be modeled as follows:

$$M : \binom{N}{d} \xrightarrow{1-1} \{0, 1\}^m \quad (|N| = n)$$

(*) To determine d positives, we need to find the 1-1 correspondence between this vector $(x_1, x_2, \dots, x_m)^t$ and the outcome vector.

Since there are $\binom{n}{d}$ distinct vectors to check, the decoding algorithm takes $O(n^d)$ times to find the answer in the worst case. (This situation comes from the assumption that M is d -separable.)

(*) If M is d -disjunct, then the decoding is much simpler, it takes $O(m)$ to find all the positive items.

Proof. Let the tests with "0" outcome be the "0-pool", denoted by $\bar{t}(M)$. Assume that C_1, C_2, \dots, C_d are the positive items.

Then, for each negative item C_{d+1} , $C_{d+1} \not\subseteq \bigcup_{i=1}^d C_i$. Hence, there exists an $x \in C_{d+1}$ such that $x \notin \bigcup_{i=1}^d C_i$. If x is in the i th

test, t_i , then $t_i \in \bar{t}(M)$. Hence, C_{d+1} can be found by using this test t_i .

In fact, all items involved in the i th test are all negative!

Since checking all the 0 -pools in $\bar{t}(M)$ can find all negative items, all positive items can be found subsequently. ■

A quick review about the 9×12 pooling design mentioned in previous lecture is in fact a 2 -disjunct matrix and therefore we can use it to find $d \leq 2$ positive items with fairly quick decoding algorithm. Similar idea from combinatorial design (projective plane of order 3) can be applied to find other disjunct matrices. Here we present the first systematic construction.

Theorem (A. J. Macula, DM 1996)

Let M be indexed by $\binom{N}{d}$ rows from $\binom{N}{d}$ and $\binom{N}{k}$ columns from $\binom{N}{k}$ where $N = [1, n]$ and $d \leq k$. Furthermore, let $M(i, j) = 1$ if the i th d -subset of N is contained in the j th k -subset of N and 0 otherwise. Then, M is a d -disjunct matrix.

Let $C_1, C_2, \dots, C_d, C_{d+1}$ be any $d+1$ columns.

Proof. It suffices to find one row, say the i th row, such

that $M(i, d+1) = 1$ but $M(i, j) = 0$ for $j = 1, 2, \dots, d$.

Since C_1, C_2, \dots, C_{d+1} are distinct k subsets of N , $C_{d+1} \setminus C_j \neq \emptyset$ for $j = 1, 2, \dots, d$. Let $x_j \in C_{d+1} \setminus C_j$ and $D = \{x_1, x_2, \dots, x_d\}$ (a multi-set).

Now, clearly $D \subseteq C_{d+1}$ but $D \not\subseteq C_j$ for each $j = 1, 2, \dots, d$. Let

\tilde{D} be a d -subset of N such that $D \subseteq \tilde{D} \subseteq C_{d+1}$ and let \tilde{D}

occur in the i th row. Since C_1, C_2, \dots, C_d can not contain \tilde{D} ,

the proof follows. ▀

(*) We remark here that a good pooling design should have
(for (d, n) -problem)

two good properties:

(1) the # of items
the # of tests is large; and

(2) fast decoding algorithm.

(**) The above properties are not the only requirements for

group testings with extra "challenges", say the occurrence

of errors, inhibitors, ..., etc.