BIBDs and Group Testing

Lucia Moura
School of Electrical Engineering and Computer Science
University of Ottawa
lucia@eecs.uottawa.ca

Winter 2017

Review BIBDs

Definition (Balanced Incomplete Block Design)

Let v, k and λ be positive integers such that $v > k \geq 2$. A (v, k, λ)-BIBD is a design $(V, \mathcal{B})$ such that

1. $|V| = v$,
2. each block contains exactly k points, and
3. every pair of distinct points is contained in exactly λ blocks.

We learned that in a (v, k, λ)-BIBD
every point appears in $r = \frac{\lambda(v-1)}{k-1}$ blocks, and there are
$b = \frac{vr}{k} = \frac{\lambda(v^2-v)}{k^2-k}$ blocks.
So a (v, k, λ)-BIBD can be written (v, b, r, k, λ)-BIBD where r and b are determined by the other 3 parameters.

Example: a $(v = 7, b = 14, r = 6, k = 3, \lambda = 2)$-BIBD

$v = 7$ points arranged in blocks of size $k = 3$ such that every pair of points appear in $\lambda = 2$ blocks

124	126
235	237
346	341
457	452
561	563
672	674
713	715

every point appears in $r = \frac{\lambda(v-1)}{k-1} = \frac{6}{2}$ blocks, and there are $b = \frac{vr}{k} = \frac{7 \times 6}{3} = \lambda\frac{(v^2-v)}{k^2-k}$ blocks.

Necessary conditions for the existence of a (v, k, λ)-BIBD

If there exist a (v, k, λ)-BIBD then

$$\begin{array}{rcl} k-1 & \mid & \lambda(v-1) \\ k(k-1) & \mid & \lambda v(v-1) \\ v & \leq & b \qquad \text{(Fisher's inequality)} \end{array}$$

Note: symbol | means "divides".

These necessary conditions are not always sufficient.

Incidence matrices

Definition

The incidence matrix M of a (v, b, r, k, λ)-BIBD $(V, \mathcal{B})$ is a $v \times b$ 01-matrix, with rows indexed by the points and columns indexed by the blocks and such that $M_{i,j} = 1$ if $i \in B_j$ and $M_{i,j} = 0$ if $i \notin B_j$

For the previous $(7, 14, 6, 3, 2)$-design, we have M:

	123456789abcbde
1	10001011010001
2	11000101101000
3	01100010110100
4	10110000011010
5	01011000001101
6	00101101000110
7	00010110100011

Incidence matrices

Definition

Let M be the incidence matrix of a BIBD. The dual design is the design corresponding to the transpose of the incidence matrix M^T.

In general this design this is not a BIBD, unless $v = b$.

	1234567
1	1101000
2	0110100
3	0011010
4	0001101
5	1000110
6	0100011
7	1010001
8	1100010
9	0110001
a	1011000
b	0101100
c	0010110
d	0001011
e	1000101

Group Testing

A blood test application:

- test a large number of blood samples for a rare disease
- because tests are expensive we combine several samples in a group before testing
- a NEGATIVE result means none of the samples in the group is positive
- a POSITIVE result means at least one of the samples in the group is positive

Group testing aims at identifying the positive samples with a small number of tests, making it more efficient than testing the samples individually.

Adaptive and non-adaptive group testing

Adaptive GT: go doing tests and using the results of previous tests to choose new tests.
Advantage:

- efficiency; avoiding unnecessary tests.

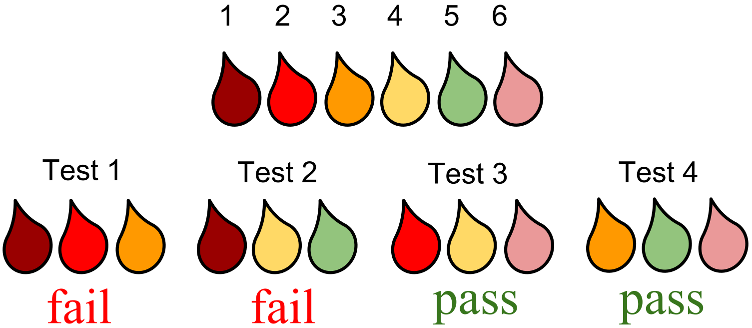
Classical method: binary splitting $O(d \log n)$ for n items and d defectives (positive blood samples)

Non adaptive GT: all the tests are decided ahead of time.
Advantages:

- potentially less overhead (tests can be organized in a convenient manner)
- tests can be performed in parallel (important if the time to set up a test is long).

Method: Test schedule can be created using combinatorial designs.

Non adaptive group testing: example



Group Testing: applications

- Biological applications, DNA applications (see CGT book by Du and Hwang 1999).
- Batch verification of digital signatures using signature aggregation and CGT (Zaverucha and Stinson 2010).
- Locating modifications on signed documents (Bardini Idalino et al. 2015)

Nonadaptive group testing schedule

Definition

Let X be a set of m elements called *samples* and $\mathcal{A}$ a set of n subsets of X called tests. A *non-adaptive group testing algorithm* (m, n)-NAGTA has threshold d if the results of tests uniquely identifies any group of up to d defective (positive) items.

Example: $X = \{1, 2, 3, 4, 5, 6\}$ and

$$\mathcal{A} = \{\{1, 2, 3\}, \{1, 4, 5\}, \{2, 4, 6\}, \{3, 5, 6\}\}$$

This is a $(6, 4)$-NAGTA. Let's determine its threshold d.

Consider the $(6, 4)$-NAGTA with $X = \{1, 2, 3, 4, 5, 6\}$ and

$\mathcal{A} = \{A_1 = \{1, 2, 3\}, A_2 = \{1, 4, 5\}, A_3 = \{2, 4, 6\}, A_4 = \{3, 5, 6\}\}.$

Suppose the set of defectives is $U \subseteq (X)$.
Let's consider the tuple of results $R(U) = (r_1, r_2, r_3, r_4)$ where $r_i = 1$ if $A_i \cap U \neq \emptyset$ (set A_i contains a defective item) and $r_i = 0$, otherwise.

For example: $U = \{1, 4\}$ $R(U) = (1110)$.
Note that another set of cardinality 2, $U' = \{2, 4\}$ yields $R(U') = R(U)$.
From this we conclude the threshold $d < 2$.
Is the threshold $d = 1$?

Is the threshold $d = 1$ for this example?

$(6,4)$-NAGTA with $X = \{1,2,3,4,5,6\}$ and

$\mathcal{A} = \{A_1 = \{1,2,3\}, A_2 = \{1,4,5\}, A_3 = \{2,4,6\}, A_4 = \{3,5,6\}\}$.

Sets U of cardinality at most $d = 1$ and their results:

U	$R(U)$
$\emptyset$	(0000)
$\{1\}$	(1100)
$\{2\}$	(1010)
$\{3\}$	(1001)
$\{4\}$	(0110)
$\{5\}$	(0101)
$\{6\}$	(0011)

Since all result vectors are distinct, the result vector will uniquely identify the set U of defectives, if $|U| \leq 2$.
So the threshold of this $(6,4)$-NAGTA is $d = 1$.

A construction of NAGTAs using BIBDs

Construction

Let $(Y, \mathcal{B})$ be a $(v, b, r, k, 1)$-BIBD, and let $(X, \mathcal{A})$ be the dual incidence structure (i.e. the design whose transpose is the incidence matrix of $(Y, \mathcal{B})$). We use $(X, \mathcal{A})$ as a (b, v)-NAGTA. We will show $d = k - 1$

Example: $(Y, \mathcal{B})$ is the $(9, 3, 1)$-BIBD we have seen before:

$$\mathcal{B} = \{123, 456, 789, 147, 258, 369, 159, 267, 348, 168, 249, 357\}$$

$(X, \mathcal{A})$ is a $(12, 9)$-NAGTA with

$$A_1 = \{1, 4, 7, 10\}, A_2 = \{1, 5, 8, 11\}, A_3 = \{1, 6, 9, 12\}$$
$$A_4 = \{2, 4, 9, 11\}, A_5 = \{2, 5, 7, 12\}, A_6 = \{2, 6, 8, 10\}$$
$$A_7 = \{3, 4, 8, 12\}, A_8 = \{3, 5, 9, 10\}, A_9 = \{3, 6, 7, 11\}$$

Let's look at incidence matrix of the $(9, 3, 1)$-BIBD:

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}
1	1	0	0	1	0	0	1	0	0	1	0	0
2	1	0	0	0	1	0	0	1	0	0	1	0
3	1	0	0	0	0	1	0	0	1	0	0	1
4	0	1	0	1	0	0	0	0	1	0	1	0
5	0	1	0	0	1	0	1	0	0	0	0	1
6	0	1	0	0	0	1	0	1	0	1	0	0
7	0	0	1	1	0	0	0	1	0	0	0	1
8	0	0	1	0	1	0	0	0	1	1	0	0
9	0	0	1	0	0	1	1	0	0	0	1	0

This gives a $(12, 6)$-NAGTA with $d = 2$.

(b, v)-NAGTAs with threshold $k - 1$ from $(v, k, 1)$-BIBDs

Theorem

If there exists a $(v, b, r, k, 1)$-BIBD, then there exists a (b, v)-NAGTA with threshold $k - 1$.

Proof. Consider the given construction and let U with $|U| \leq k - 1$ and let $r(U)$ be the result vector. First, note that $U \subseteq X \setminus \bigcup_{\{A_i \in \mathcal{A} : r_i = 0\}} A$.
We claim that since the NAGTA is the dual of a BIBD with $\lambda = 1$, $U = X \setminus \bigcup_{\{A_i \in \mathcal{A} : r_i = 0\}} A$.
This is true, otherwise there exists $x \notin U$ such that $x \notin \bigcup_{\{A_i \in \mathcal{A} : r_i = 0\}} A$. In other words $x \notin U$ and $r_i = 1$ for every A_i such that $x \in A_i$.
There are k such sets and each of them must contain an element of U. But since $\lambda = 1$ each of these elements that occur together with x in a block must be all different. So $|U| \geq k$, contradicting our assumption. $\square$

Algorithm to identify U from the (b, v)-NAGTA results

Algorithm $\mathrm{IDENTIFY}$ (R(U))

```
U ← ∅;
for i ← 1 to b do
    M[i] ← 1;
for j ← 1 to v do
    if r_j = 0 then
        for each x ∈ A_j do M[x] ← 0;
for i ← 1 to b do
    if M[i] = 1 then U ← U ∪ {i};
if |U| ≤ k − 1 then return U
  else return("the positive subset has size at least k");
```

References

- T. Bardini Idalino, L. Moura, R.F. Custodio and D. Panario, Locating modifications in signed data for partial data integrity, *Information Processing Letters* **115** (2015), 731–737.
- D-Z Du and F.K. Hwang, "Combinatorial group testing and its applications", 2nd ed, 1999.
- D. R. Stinson, "Combinatorial Designs: Constructions and Analysis", 2004.
- G. M. Zaverucha and D. R. Stinson. Group testing and batch verification. *Lecture Notes in Computer Science* **5973** (2010), 140-157. (ICITS 2010 Proceedings.)