

Lecture 13

①

Finding Signals (Continued)

1. Statistical Approach

2. Similarity-Based Approach

Uses spliced alignment algorithm

⋮

(*) The Twenty Questions Game with Genes

Given an (unknown) set I of integers in the interval $[1, n]$, reconstruct the set I by asking the minimum number of queries of the form "does q given interval contain an integer from I ?"

[Note: In this formulation, interval $[1, n]$ corresponds to

cDNA, " I " corresponds to exon boundaries in cDNA, and the

↓
queries correspond to PCR reactions defined by a pair of primers.

original gene with cut-out introns

(complementary copy of an mRNA)

實驗的範圍

(2)

$$(|I| = k)$$

If the number of exon boundaries k is known. Then we can do something on finding the lower bounds.

Lemma The number of queries is at least $\log_2 \binom{n}{k}$.

Proof. By decision tree model, the decision assumes

sequential queries using one query at a time. Since

there are $\binom{n}{k}$ possible outcomes and the answer is either

yes or no, we conclude that we need at least h

queries to obtain the answer where $2^h \geq \binom{n}{k}$. This

concludes the proof. ■

(Note) $\log_2 \binom{n}{k} \approx k \log_2 n - k \log_2 k$.

$k=1$ ✓

$k=2$?

$$\log_2 \binom{n}{2} \leq h \leq 2 \log_2 n$$

↑
(Informatic Lower Bound)

Gap

(3)

(*) If a biologist tolerates an error Δ in the positions of exon boundaries, then the lower bound on the number of queries is $\approx k \log_2 \frac{n}{\Delta} - k \log_2 k$.
(Not easy to verify!).

In practical situation, we use about 30 primers and (typical cDNA).

3 rounds to find exon boundaries.

Hidden Graph Problem

A hidden graph G is known belonging to a given family $\mathcal{G}$ of labeled graphs on the set $N = [1, n]$.

The problem is to reconstruct G by asking queries as

few as possible, where a query is of the form

(*) "Does $S \subseteq N$ induce one edge of G ?" That is " $G[S]$ at least

contains an edge or $G[S]$ is not an empty graph (with no edges).

(*) Of course, we may ask $\binom{n}{2}$ pairs of vertices in $[1, n]$. But, it takes too many queries.

④

Four models

- (1) No restriction on the size of S and the answer ^{is either} ~~are~~ "Yes" or "No".
(not both)
- (2) No restriction on the size of S and the answer is the number of edges in $G[S]$.
- (3) $|S| \leq k$ and the answer is either Yes or No but not both.
- (4) $|S| \leq k$ and the answer is the size of $G[S]$.

- (1) General model
- (2) Quantity model
- (3) k -vertex general model
- (4) k -vertex quantity model

Algorithms

Step 1. Find a vertex of G .

Step 2. Find an edge of G .

Step 3. Find G . (The best idea so far is via finding a maximal matching of G .)

⑤

Algorithm of Step 1 (Find-one-vertex)

1. $S \leftarrow N$

$$Q(S) = \begin{cases} 0, & G[S] \text{ contains no edges,} \\ 1, & \text{otherwise.} \end{cases}$$

2. if $Q(S) = 0$ then

3. Return $\emptyset$.

4. end if

5. $A \leftarrow N$

6. while $|A| > 1$ do

7. Partition (arbitrarily) A into two balanced sets A_0 and A_1

8. if $Q(S \setminus A_0) = 1$ then

9. $S \leftarrow S \setminus A_0, A \leftarrow A_1$.

10. else

11. $A \leftarrow A_0$

12. end if

13. end while

14. Return the element in A .

(•) If the hidden graph has m edges, then there are $\binom{n}{2}^m$ possible solutions for the graph.

(•) As mentioned above, if we apply the so-called binary splitting idea to find these m edges, then we need apply at least $\log_2 \left(\binom{n}{2}^m \right)$ queries.

(*) In fact, we need more than that. Observe that we need to find one edge to start, and then the second one, and so on. This fact was proved by Chang et al. later in a paper [1].

Theorem [1] $\left\lceil \log_2 \left(\sum_{i=0}^m \binom{n}{2}^i \right) \right\rceil$ edge-detecting queries are required to identify a graph drawn from the family of all graphs of order n and size m .

[1] Huilan Chang, H.L. Fu and Chih-Huai Shih, Learning a hidden graph, Optimization Letters, 8(2014), no. 8, 2341-2348.

Proof. The main idea comes from the computation tree of search. From the leaves, we find the first edge. Then, delete the edge, $\dots$ (Omitted).

(*) After some effort, the following theorem was obtained.

Theorem [1]

For the family $\mathcal{G}$ of all graphs of size m on $[n]$, a hidden graph G can be reconstructed in $m \log_2 n + 10m + 3n$ edge-detecting queries.

(*) If we are dealing with hypergraphs with edge size larger than two, then it is getting more complicate.

Theorem [2] For an r -uniform hidden hypergraph ^{of size m} defined on $[n]$, we need $m r \log_2 n + (6e)^r m^{\frac{r+1}{2}}$ queries to reconstruct the graph.

[2] Learning a hidden uniform hypergraph (work with Huilan Chang and Chi-Huai Shih), *Optim. Letters*, 12 (2018), 55-62.

Lecture 14

Group Testing (An overview)

①

Date back: 1943, Blood testing during World War II
(Dorfman and Rosenblatt)

(First GT paper by Dorfman)

Group Testing Problem

Classical

Given a set N of n clones, each of which is either positive or negative but not both. The goal of group testing is to identify all positive ones (positives in short) by using as few number of "group tests" as possible. Here, a group test is an experiment on a subset of clones and the outcome of a group test on a subset $S \subseteq N$ is "1" if S contains at least one positive clone and "0" otherwise.

Example $N = \{1, 2, 3, 4, 5, 6\}$ and 3 is the only positive clone.
Then $t(\{1, 3, 3\}) = 1$ and $t(\{3, 4, 5\}) = 0$.

- Assumption on combinatorial group testing.

Let d be the number of positives in a set N of size n .
Then, we assume that $d \ll n$, i.e., d/n is very close to "0".

(Note) If d is too large comparing to n , then "group testing" is not working well in general.

(2)

Algorithms

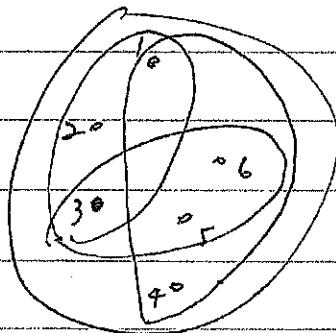
1. Adaptive : An adaptive algorithm conducts tests one by one (Sequential)
2. Non-adaptive : A non-adaptive algorithm specifies all tests simultaneously.
3. k-stage algorithm : All tests in the "first" stage are specified simultaneously.

• The non-adaptive algorithm is also known as pooling design which can be depicted by using a matrix M .

Let ^{the} number of clones be n and there are t tests.

The matrix M corresponding to a pooling design has the following properties :

- (a) M is a t by n matrix; and
- (b) $M(i,j) = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ test including clone "j"} \\ 0 & \text{otherwise.} \end{cases}$



$$\begin{array}{c}
 \begin{matrix} t_1 \\ t_2 \\ t_3 \end{matrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \\ \\ \end{bmatrix}
 \end{array}$$

outcome vector

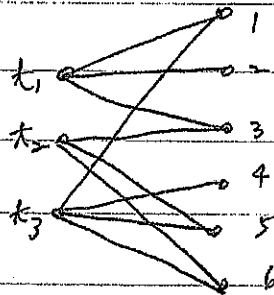
If (3) is positive, then the outcome vector is $(1, 1, 0)^t$.

③

How much do we know about $(0,1)$ -matrix?

(Fact 1) If M is a $t \times n$ $(0,1)$ -matrix, then we can find a labeled bipartite graph whose incidence matrix is M . Here $|A| = t$ and $|B| = n$. $G = (A, B)$

So, the above matrix is corresponding to the following bipartite graph



(Fact 2) If we let A be the set of elements and B is the collection of blocks, then we have a variety-block graph.

Again, the above example shows that there are six blocks $\{\{1,3\}, \{1\}, \{1,2\}, \{3\}, \{2,3\}, \{2,3\}\}$.

- A design can be described by using a $(0,1)$ -matrix, so it is a hypergraph.

(Fact 3) A $t \times n$ $(0,1)$ -matrix M can also be seen as a code where codewords are columns of M .

④

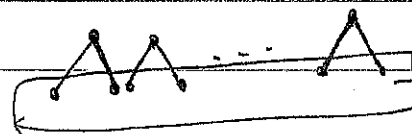
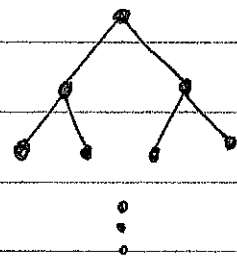
(Note) For practical use, we don't consider codewords from rows! (But, in G.T., each row is a test (pool).)

(*) The idea of finding positives

If we have d positives, then there are $\binom{n}{d}$ possible combinations to have d positives. In order to find these positives, we may use different algorithms. In general, adaptive algorithms take less tests than non-adaptive algorithm. But, the later one "save time".

Information lower bound

We can use the idea of computational tree ^{to} find how many tests are needed (at least).



$$2^h = \binom{n}{d} \Rightarrow h \approx \left\lceil \log_2 \binom{n}{d} \right\rceil.$$

(*) We need at least $\left\lceil \log_2 \binom{n}{d} \right\rceil$ tests!

⑤

Another point of view

Consider all possible outcomes if t tests are applied. Then, there are at most 2^t outcome vectors which are distinct. So, if we use t tests to find d positives out of n , then $\binom{n}{d} \leq 2^t$ and thus

$$t \geq \left\lceil \log_2 \binom{n}{d} \right\rceil \approx d \log_2 \frac{n}{d}.$$

(Note) This idea is good for both adaptive and non-adaptive algorithms.

(Fact 4) ^{checking the} With the idea of outcome vectors, we may extend the study of error-correcting codes to construct pooling designs which can also solve the group testing problem with the occurrence of errors in experiments.

In what follows, we shall focus on non-adaptive algorithms or k-stage algorithms due to the fact that apply them in computational molecular biology.

Definition (pool)

Each test is also said to be a pool. A positive pool is a test (corresponding to the pool) which is positive, i.e., some of the clones in the experiment is positive and a negative pool is defined correspondingly, i.e., every clone in this pool is negative.

(6)

Separable matrices

$$M = [M(i,j)]_{m \times n} \quad (t \text{ tests and } n \text{ clones})$$

- We shall index the columns of M by $1, 2, \dots, n$ and the rows of M by $t_1, t_2, \dots, t_m$.

- The clone (j^{th}) can be represented by a set C_j where $C_j = \{i \mid M(i,j) = 1\}$.

- The union of s columns $C_{j_1}, C_{j_2}, \dots, C_{j_s}$ is $\bigcup_{k=1}^s C_{j_k}$ which is the Boolean sum of these columns.

e.g.
$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \cup \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

$$\begin{aligned} &\downarrow \\ &U(\{j_1, j_2, \dots, j_s\}) \\ &= U(K) \end{aligned}$$

- The intersection of s' columns $C_{j_1}, C_{j_2}, \dots, C_{j_{s'}}$ is $\bigcap_{k=1}^{s'} C_{j_k}$.

e.g.
$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \cap \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Definition (d -separable).

A matrix is d -separable if for any two ^{distinct} sets of d columns D and D' , $U(D) \neq U(D')$.

Definition ($\bar{d}$ -separable)

A matrix is $\bar{d}$ -separable if for any two distinct sets of at most d columns D and D' , i.e., $|D| \leq d, |D'| \leq d, U(D) \neq U(D')$.

Definition (Disjunct matrices)

M is called d -disjunct if no column is contained in the union of any other d columns.

M is

M is either a

- (*) d -separating if ~~one of~~ d -separable, $\bar{d}$ -separable or d -disjunct matrix.

⑦

Lemma $\Rightarrow d$ -disjunct $\Rightarrow \bar{d}$ -separable $\Rightarrow d$ -separable.

(easy)

Proof. If M is not $\bar{d}$ -separable, then there D and D' with $1 \leq |D| \leq d$, $1 \leq |D'| \leq d$, and $U(D) = U(D')$. This implies that there exists a column from D which is contained in $U(D')$ and there $U(D'')$ where $D' \subseteq D''$ and $|D''| = d$. Hence M is not d -disjunct. $\square$

(Note: 直接否亦可。)

Proposition A d -separable matrix can be applied to find "exactly d positives." Proof. (See next page.)

Proposition $\bar{d}$ -separable $\Rightarrow$ to find at most d positives

(*) To locate the set of positives we need to check all the outcome vectors which is quite complicate.

(Decoding Algorithm) is needed.

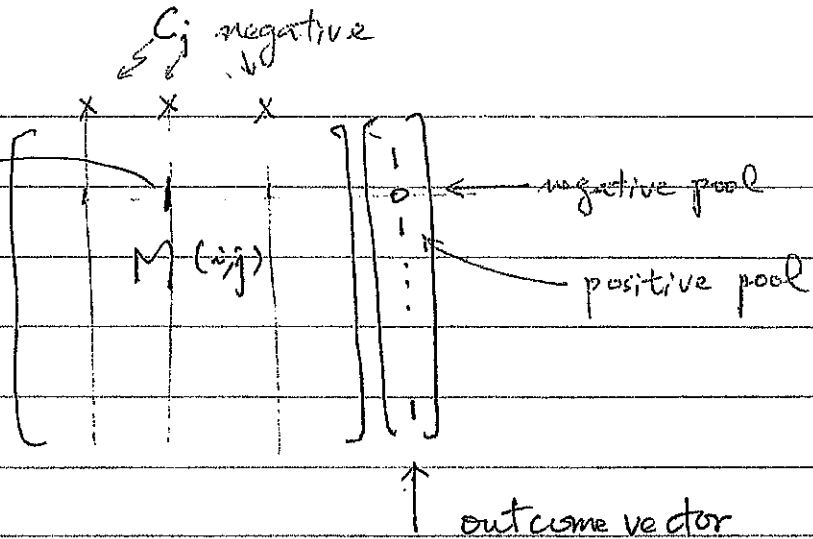
Here is an observation which is important!

↓ next page

②'

$$M(i,j)=1$$

$\equiv$ the i th row of M intersects the j th column.



- A test is ^(positive)negative if the corresponding component in the outcome vector is "0".
- negative (positive) test is also called negative (positive) pool.

观察: Negative pool 中包含的 clones 全部是 "Negative"!

结果: 先利用 "negative" pools 把一些 clones 消灭, 再从剩下的 clones 中找 d 个即可。

(*) Hitting Set Problem

Given a set X and a ^{collection $\mathcal{C}$ of} subsets of X , find a minimum-cardinality subset Y of X such that $Y \cap C \neq \emptyset \forall C \in \mathcal{C}$.

Let M_D be a $k_D \times n_D$ matrix obtained from M by keeping only the set T_D of the positive pools and the set N_D of n_D clones not appearing in any negative pool.

Outcome Vector: Can be recognized as a set, V .

Denote by $t_o^V(C) = |C \setminus V|$ where C is a column.

e.g.

		1	2	3	4	5	6	V
t_1		1	1	1	0	0	0	1
t_2		1	0	0	1	1	0	0
t_3		0	1	0	1	0	1	1
t_4		0	0	1	0	1	1	1

↑ ↑ ↑ ↑ ↑ ↑
 $C_1 C_2 C_3 C_4 C_5 C_6$

(•) Outcome vector plays the most important role in decoding.

$$V = C_2 \cup C_3$$

Note that $t_o^V(C) = |C \setminus V|$, $t_1^V = |C \cap V|$.

$t_o^V(C_1) = 1$	$t_1^V(C_1) = 1$
$t_o^V(C_2) = 0$	$t_1^V(C_2) = 2$
$t_o^V(C_3) = 0$	$t_1^V(C_3) = 2$
$t_o^V(C_4) = 1$	$t_1^V(C_4) = 1$
$t_o^V(C_5) = 1$	$t_1^V(C_5) = 1$
$t_o^V(C_6) = 0$	$t_1^V(C_6) = 2$

0-pool
包含这些 items!

↑
Negatives
 C_1, C_4, C_5

可能只是 $\begin{matrix} \textcircled{1} \textcircled{2} \\ \textcircled{2} \textcircled{5} \\ \textcircled{3} \textcircled{6} \end{matrix}$

如果是 2-disjunct 就不必留下 C_2, C_3, C_6 等待判断。

✓ positives

$C_1, C_2, C_3, C_4, C_5, C_6$

$$M: \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Fact 1. $V = U(C_p)$.

$$t_0^V(C_i)$$

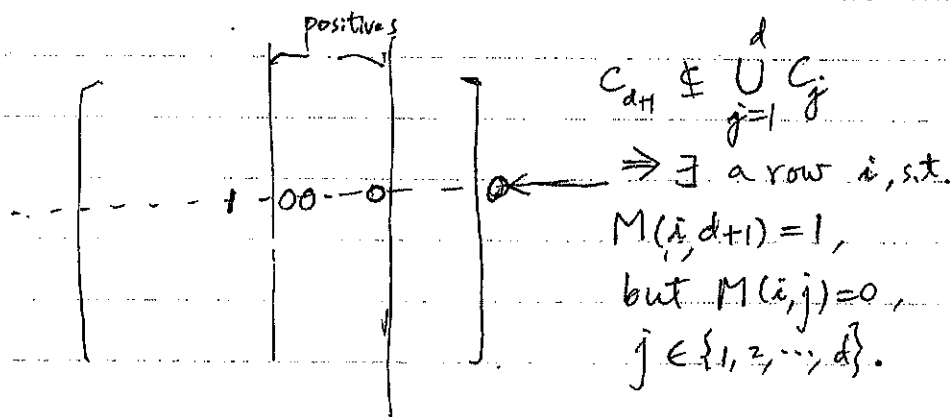
Fact 2 If a column C_i satisfying $|C_i \setminus V| > 0$, then C_i corresponds to a negative item.

Proof. All positive item corresponds to a subset of V . $\blacksquare$

Fact 3 If C_i is positive, then $|C_i \setminus V| = 0$.

Fact 4. If M is d -disjunct, then there exists a linear time decoding algorithm to find the set of d positives.

Proof. By evaluating $t_o^V(C)$ for each columns in M , we conclude the proof by using Fact 2 and Fact 3.



(*) We can use a d -disjunct matrix to identify up-to- d positive clones. ⑦ #4

Theorem If a set of n clones which contains at most d positive then a d -disjunct matrix can be applied to identify the positives. Moreover, the decoding algorithm is linear complexity of

Proof. The proof follows by evaluating $t_0^V(C)$ for each clone C . (V is the outcome vector.) We remark here that
 $t_0^V(C) = 0$ if C is positive and $t_0^V(C) \geq 1$ if C is negative.
 $(t_0^V(C) = |C \setminus V|.)$

写下一个 2-disjunct matrix

(证明) $V = \bigcup_{C \text{ is positive}} C$, 如果存在 C' 是 negative, 而 $|C' \setminus V| = 0$, 则 $C' \subseteq V$, 这 $\Rightarrow d$ -disjunct 的假设不符。

Algorithm (Clone Selection ($N, V, D(e)$))

```

1 for each clone  $C \in N$ , ( $D \leftarrow \emptyset$ )
2   compute  $t_0^V(C)$ 
3   if  $t_0^V(C) \leq e$ 
4     then  $D \leftarrow D \cup \{C\}$ 
5 return
```

(*) 上面定理由 Clone Selection ($N, V, D, 0$)。

$d+1$ -separable $\Rightarrow$ d -disjunct

Proof: Suppose that M is not d -disjunct. Then, there exist $d+1$ columns $C_1, C_2, \dots, C_{d+1}$, such that $C_{d+1} \subseteq \bigcup_{i=1}^d C_i$. This implies that $\bigcup_{i=1}^d C_i = C_{d+1} \cup (\bigcup_{i=1}^d C_i)$ and thus M is not $d+1$ -separable.

(*) In general, construct a d -disjunct matrix with a good ratio of d and n is very difficult.

(**) For each $d \leq n$, we are able to construct a d -disjunct matrix by using combinatorial design or by probabilistic method.

(*) The main tool using in probabilistic method is known as the Lovász Local Lemma.

(**) This lemma is very useful in proving the existence of certain events probabilistically.

在 ②* 中大致介绍这个引理, 希望它能在适当的时机被拿来应用。

Definition (Mutual independence)

An event A is said to be mutually independent of a set of events $\{B_i\}$ if for any subset β of events or their complements contained in $\{B_i\}$, we have $\Pr[A|\beta] = \Pr[A]$.

↑
conditional
probability

Definition (Dependency digraph) → If the cases are of symmetric form, then a "graph".

Let $A_1, A_2, \dots, A_n$ be events in a probability space. A directed graph $D = (V, A)$ with vertex set $V = \{1, 2, \dots, n\}$ is a dependency digraph for $A_1, A_2, \dots, A_n$ if for each i , A_i is mutually independent^{of} A_j , then $(i, j) \notin A$.

(*) For symmetric case, if A_i is mutually independent of A_j , then A_j is also independent of A_i .

(*) The maximum "out-degree" (or degree in a graph) will denoted by d .

Lovász Local Lemma (Symmetric Form)

Let $A_1, A_2, \dots, A_n$ be a set of events with $\Pr[A_i] \leq p$ for all $1 \leq i \leq n$ and $0 < p < 1$. If the dependency graph has maximum degree d and $ep(d+1) \leq 1$, then $\Pr\left[\bigcap_{i=1}^n \bar{A}_i\right] > 0$.
($e = 2.71828 \dots$, $\bar{A}_i$ denotes the complement of A_i .)

Proof. See attached from Wiki.

(*) Lovász Local Lemma can be carried out by using an algorithm: Algorithmic L.L.L.

(*) Since it is a random type of arguments, we need the aid of random variables.

(*) Here, we provide an explanation in next page.

(*) 主要的概念是不斷地選擇適當的 "evaluation" 來排除所有的 "Bad events", 如果完成, 就 "output" 它的結果, 例如 "p" 的選擇。

Lovász local lemma

In probability theory, if a large number of events are all independent of one another and each has probability less than 1, then there is a positive (possibly small) probability that none of the events will occur. The **Lovász local lemma** allows one to relax the independence condition slightly: As long as the events are "mostly" independent from one another and aren't individually too likely, then there will still be a positive probability that none of them occurs. It is most commonly used in the probabilistic method, in particular to give existence proofs.

There are several different versions of the lemma. The simplest and most frequently used is the symmetric version given below. A weaker version was proved in 1975 by László Lovász and Paul Erdős in the article *Problems and results on 3-chromatic hypergraphs and some related questions*. For other versions, see (Alon & Spencer 2000).

Contents

Statements of the lemma (symmetric version)

Asymmetric Lovász local lemma

Constructive versus non-constructive

Non-constructive proof

Example

Notes

References

Statements of the lemma (symmetric version)

Let $A_1, A_2, \dots, A_k$ be a sequence of events such that each event occurs with probability at most p and such that each event is independent of all the other events except for at most d of them.

Lemma I (Lovász and Erdős 1973; published 1975) If

$$4pd \leq 1$$

then there is a nonzero probability that none of the events occurs.

Lemma II (Lovász 1977; published by Joel Spencer^[1]) If

$$ep(d+1) \leq 1,$$

where $e = 2.718\dots$ is the base of natural logarithms, then there is a nonzero probability that none of the events occurs.

Lemma II today is usually referred to as "Lovász local lemma".

Lemma III (Shearer 1985^[2]) If

$$\begin{cases} p < \frac{(d-1)^{d-1}}{d^d} & d > 1 \\ p < \frac{1}{2} & d = 1 \end{cases}$$

then there is a nonzero probability that none of the events occurs.

The threshold in Lemma III is optimal and it implies that the bound

$$epd \leq 1$$

is also sufficient.

Asymmetric Lovász local lemma

A statement of the asymmetric version (which allows for events with different probability bounds) is as follows:

Lemma (asymmetric version). Let $\mathcal{A} = \{A_1, \dots, A_n\}$ be a finite set of events in the probability space Ω . For $A \in \mathcal{A}$ let $\Gamma(A)$ denote the neighbours of A in the dependency graph (In the dependency graph, event A is not adjacent to events which are mutually independent). If there exists an assignment of reals $x : \mathcal{A} \rightarrow [0, 1]$ to the events such that

$$\forall A \in \mathcal{A} : \Pr(A) \leq x(A) \prod_{B \in \Gamma(A)} (1 - x(B))$$

then the probability of avoiding all events in $\mathcal{A}$ is positive, in particular

$$\Pr(\overline{A_1} \wedge \dots \wedge \overline{A_n}) \geq \prod_{i \in \{1, \dots, n\}} (1 - x(A_i)).$$

The symmetric version follows immediately from the asymmetric version by setting

$$\forall A \in \mathcal{A} : x(A) = \frac{1}{d+1}$$

to get the sufficient condition

$$p \leq \frac{1}{d+1} \cdot \frac{1}{e}$$

since

$$\frac{1}{e} \leq \left(1 - \frac{1}{d+1}\right)^d.$$

Constructive versus non-constructive

Note that, as is often the case with probabilistic arguments, this theorem is nonconstructive and gives no method of determining an explicit element of the probability space in which no event occurs. However, algorithmic versions of the local lemma with stronger preconditions are also known (Beck 1991; Czumaj and Scheideler 2000). More recently, a constructive version of the local lemma was given by Robin Moser and Gábor Tardos requiring no stronger preconditions.

Non-constructive proof

We prove the asymmetric version of the lemma, from which the symmetric version can be derived. By using the principle of mathematical induction we prove that for all A in $\mathcal{A}$ and all subsets S of $\mathcal{A}$ that do not include A , $\Pr\left(A \mid \bigwedge_{B \in S} \overline{B}\right) \leq x(A)$. The induction here is applied on the size (cardinality) of the set S . For base case $S = \emptyset$ the statement obviously holds since $\Pr(A_i) \leq x(A_i)$. We need to show that the inequality holds for any subset of $\mathcal{A}$ of a certain cardinality given that it holds for all subsets of a lower cardinality.

Let $S_1 = S \cap \Gamma(A)$, $S_2 = S \setminus S_1$. We have from Bayes' theorem

$$\Pr\left(A \mid \bigwedge_{B \in S} \overline{B}\right) = \frac{\Pr\left(A \wedge_{B \in S_1} \overline{B} \mid \bigwedge_{B \in S_2} \overline{B}\right)}{\Pr\left(\bigwedge_{B \in S_1} \overline{B} \mid \bigwedge_{B \in S_2} \overline{B}\right)}.$$

We bound the numerator and denominator of the above expression separately. For this, let $S_1 = \{B_{j1}, B_{j2}, \dots, B_{jt}\}$. First, exploiting the fact that A does not depend upon any event in S_2 .

$$\text{Numerator} \leq \Pr\left(A \mid \bigwedge_{B \in S_2} \overline{B}\right) = \Pr(A) \leq x(A) \prod_{B \in \Gamma(A)} (1 - x(B)). \quad (1)$$

Expanding the denominator by using Bayes' theorem and then using the inductive assumption, we get

$$\begin{aligned} & \text{Denominator} \\ &= \Pr \left(\overline{B}_{j1} \mid \bigwedge_{t=2}^l \overline{B}_{jt} \wedge \bigwedge_{B \in S_2} \overline{B} \right) \cdot \Pr \left(\overline{B}_{j2} \mid \bigwedge_{t=3}^l \overline{B}_{jt} \wedge \bigwedge_{B \in S_2} \overline{B} \right) \cdots \Pr \left(\overline{B}_{jl} \mid \bigwedge_{B \in S_2} \overline{B} \right) \geq \prod_{B \in S_1} (1 - x(B)) \end{aligned} \quad (2)$$

The inductive assumption can be applied here since each event is conditioned on lesser number of other events, i.e. on a subset of cardinality less than $|S|$. From (1) and (2), we get

$$\Pr \left(A \mid \bigwedge_{B \in S} \overline{B} \right) \leq x(A) \prod_{B \in \Gamma(A) - S_1} (1 - x(B)) \leq x(A)$$

Since the value of x is always in $[0, 1]$. Note that we have essentially proved $\Pr \left(\overline{A} \mid \bigwedge_{B \in S} \overline{B} \right) \geq 1 - x(A)$. To get the desired probability, we write it in terms of conditional probabilities applying Bayes' theorem repeatedly. Hence,

$$\begin{aligned} & \Pr \left(\overline{A}_1 \wedge \cdots \wedge \overline{A}_n \right) \\ &= \Pr \left(\overline{A}_1 \mid \overline{A}_2 \wedge \cdots \wedge \overline{A}_n \right) \cdot \Pr \left(\overline{A}_2 \mid \overline{A}_3 \wedge \cdots \wedge \overline{A}_n \right) \cdots \Pr \left(\overline{A}_n \right) \\ &\geq \prod_{A \in \mathcal{A}} (1 - x(A)), \end{aligned}$$

which is what we had intended to prove.

Example

Suppose $11n$ points are placed around a circle and colored with n different colors in such a way that each color is applied to exactly 11 points. In any such coloring, there must be a set of n points containing one point of each color but not containing any pair of adjacent points.

To see this, imagine picking a point of each color randomly, with all points equally likely (i.e., having probability $1/11$) to be chosen. The $11n$ different events we want to avoid correspond to the $11n$ pairs of adjacent points on the circle. For each pair our odds of picking both points in that pair is at most $1/121$ (exactly $1/121$ if the two points are of different colors, otherwise 0), so we will take $p = 1/121$.

Whether a given pair (a, b) of points is chosen depends only on what happens in the colors of a and b , and not at all on whether any other collection of points in the other $n - 2$ colors are chosen. This implies the event " a and b are both chosen" is dependent only on those pairs of adjacent points which share a color either with a or with b .

There are 11 points on the circle sharing a color with a (including a itself), each of which is involved with 2 pairs. This means there are 21 pairs other than (a, b) which include the same color as a , and the same holds true for b . The worst that can happen is that these two sets are disjoint, so we can take $d = 42$ in the lemma. This gives

$$ep(d + 1) \approx 0.966 < 1.$$

By the local lemma, there is a positive probability that none of the bad events occur, meaning that our set contains no pair of adjacent points. This implies that a set satisfying our conditions must exist.

Notes

1. Spencer, J. (1977). "Asymptotic lower bounds for Ramsey functions". *Discrete Mathematics*. 20: 69–76. doi:10.1016/0012-365x(77)90044-9 ([https://doi.org/10.1016/0012-365x\(77\)90044-9](https://doi.org/10.1016/0012-365x(77)90044-9)).
2. Shearer, J (1985). "On a problem of Spencer". *Combinatorica*. 5 (3): 241–245. doi:10.1007/BF02579368 (<https://doi.org/10.1007/BF02579368>).

References

- Alon, Noga; Spencer, Joel H. (2000). *The probabilistic method* (2nd ed.). New York: Wiley-Interscience. ISBN 0-471-37046-0.
- Beck, J. (1991). "An algorithmic approach to the Lovász local lemma, I". *Random Structures and Algorithms*. 2 (4): 343–365. doi:10.1002/rsa.3240020402 (<https://doi.org/10.1002/rsa.3240020402>).
- Czumaj, Artur; Scheideler, Christian (2000). "Coloring nonuniform hypergraphs: A new algorithmic approach to the general Lovász local lemma". *Random Structures & Algorithms*. 17 (3–4): 213–237. doi:10.1002/1098-2418(200010/12)17:3/4<213::AID-RSA3>3.0.CO;2-Y ([https://doi.org/10.1002/1098-2418\(200010/12\)17:3/4<213::AID-RSA3>3.0.CO;2-Y](https://doi.org/10.1002/1098-2418(200010/12)17:3/4<213::AID-RSA3>3.0.CO;2-Y)).

Let $\mathcal{P}$ denote a set of random variables.

$\forall \mathcal{P} \in \mathcal{P}$, let $v_{\mathcal{P}}$ denote the current assignment (evaluation) of $\mathcal{P}$.

The unique minimal subset of random variables in $\mathcal{P}$ that determine the event A is denoted by $vbl(A)$.

If the event A is true under an evaluation $(v_{\mathcal{P}})_{\mathcal{P}}$, we say that $(v_{\mathcal{P}})_{\mathcal{P}}$ satisfies A , otherwise it avoids A .

Given a set of "bad" events $\mathcal{A}$ we wish to avoid

A collection of mutually independent random variables $\mathcal{P}$.

Algorithm

1. $\forall \mathcal{P} \in \mathcal{P}; v_{\mathcal{P}} \leftarrow$ a random evaluation of $\mathcal{P}$.
2. While $\exists A \in \mathcal{A}$ such that A is satisfied by $(v_{\mathcal{P}})_{\mathcal{P}}$,
 - pick an arbitrary satisfied event $A \in \mathcal{A}$.
 - $\forall \mathcal{P} \in vbl(A): v_{\mathcal{P}} \leftarrow$ a new random evaluation of $\mathcal{P}$.
3. return $(v_{\mathcal{P}})_{\mathcal{P}}$.

(8)

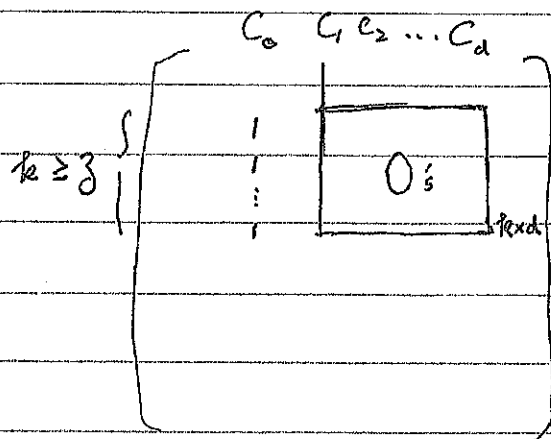
Definition $(d; z)$ -disjunct matrix.

A matrix is said to be $(d; z)$ -disjunct if for any $d+1$ columns $C_0, C_1, \dots, C_d$,

$$\left| C_0 \setminus \bigcup_{i=1}^d C_i \right| \geq z.$$

Note $(d; 1)$ -disjunct $\approx$ d -disjunct.

(*) There exist at least z rows in each of which C_0 has a "1"-entry and every $C_i, 1 \leq i \leq d$, has a "0"-entry.



Theorem A $(d; z)$ -disjunct matrix can be applied to identify up-to- d positives with at most e errors.

Proof. We remark that the errors occur on the conclusion of tests, i.e., the outcome vector (as a codeword) has at most e errors. (Due the fact that errors are possible, we shall have a set of possible outcomes, instead of just one outcome vector V). Now, assume the number of errors is e . For each negative clone $\bar{C}$, by definition, there exist at least $2e+1$ rows intersecting $\bar{C}$ but none of D (the set of positives). This implies that for each outcome vector V , $t_V(\bar{C}) \geq 2e+1$.

$$t_0^V(C) = |C \cap V|$$

⑨

On the other hand, for each positive clone C^+ , $t_0^V(C^+) \leq e$.
 ($t_0^V(C) = 0$ if no errors occurred.) So, the proof follows by checking $n(=|N|)$ clones by using the function $t_0^V(C)$ where V is the outcome vector (in $\mathbb{V}$). Note, there is exactly one outcome vector after the experiment is done. ■

Definition (Inhibitor)

A clone is an inhibitor if the clone neutralize positive clones, that is, the presence of an inhibitor in a pool dictates a negative outcome, regardless of the presence of positive clone in the pool.

(註) Inhibitor 出現時, 該 pool 是 陰性反應。

Lemma

⑨

Theorem (Error-tolerant inhibitor model)

A $(d+h; 2e+1)$ -disjunct matrix can be applied to identify all positive clones if there are at most d positives and h inhibitors

Proof. Observe that a clone C , which appears in at most e positive pools, i.e., $t_1^V(C) \leq e$, cannot be positive due to the $(d+h; 2e+1)$ -disjunctness property. (If C is positive, then $t_1^V(C) \geq 2e+1$ if there are no errors.) For the inhibitors, each inhibitor can occur in at most e positive pools (with e errors), i.e., $t_1^V(I) \leq e$.

Now, let O be the set of columns C such that $t_1^V(C) \leq e$. This implies that O contains all inhibitors but no positives. Hence, D , the set of positive clones is a subset of $N \setminus O$.

⑨'

Lemma (e errors and h inhibitors)

If M is an $(m; z)$ -disjunct matrix with $m \geq z$, then

- (a) $t_1^V(p) \geq m - h + z - e$ if p is positive;
- (b) $t_0^V(Q) \geq m - d + z - e$ if Q is negative; and
- (c) $t_1^V(I) \leq e$ if I is an inhibitor.

Here, V is the outcome vector.

Proof. Since $m \geq h$, M is also an $(z; z + m - h)$ -disjunct matrix. Hence $t_1^V(p) \geq m - h + z$ if there are no errors. Now, if e errors occur, then $t_1^V(p) \geq m - h + z - e$. This proves (a).

(Note that $t_1^V(S) = |S \cap V|$.) Moreover, ⁱⁿ an $(m; z)$ -disjunct matrix, the outcome vector must have at least z "1"s.

The proof of (b) is similar. Let's consider (c). All 1's are from errors



(10)

Let $C^- \in N \setminus O$. By disjointness property, we have $t_0^V(C^-) \geq (2e+1) - e = e+1$. (?) On the other hand, if C^+ is a positive clone in $N \setminus O$, then $t_0^V(C^+) \leq e$. (?) Therefore, by calculating $t_0^V(C)$ where $C \in N \setminus O$, we conclude the proof. ■

说明：在這個定理中我們只負責把 positives 找出來，基本上不知道那幾 γ clones 是 inhibitors.

说明'： $(d+h, \gamma)$ -disjointness 是用來把 Inhibitors 分出來；如果沒有 inhibitors (d, γ) -disjointness 就夠用了。

Group Testing on Complexes

In the complex model, we consider a set N of n clones and the positives are in an unknown family $P = \{P_i\}$ of subsets of N .
(known)

(To positive outcome 要由一些 clones 所組合而成，不再是單一 clone 而已。)

(*) Each member of P is called a ^{positive} complex; (each subset of N is a candidate of a member of P .)

(*) An experiment is applied to an arbitrary subset $S \subseteq N$ with two possible outcomes; a negative outcome implies S does not contain any $P_i \in P$, and a positive outcome implies otherwise.

(11)

Complex Model

- 我们必须假设当 X, X' 皆为 complexes 时, 彼此没有互相包含的图像, 亦即 $X \not\subseteq X'$ 且 $X' \not\subseteq X$.
- Or the model must be monotone on positives, i.e., if X is positive and $X \subseteq X'$, then X' is also positive. But, this is not quite true in applications.
- If not, and X contains a proper subset X^+ which is positive, then X can only appear in positive pools no matter it is positive or negative. Thus, X cannot be identified.
- Let H denote the given set of complexes. Then H can be viewed as a hypergraph with clones as vertices and complexes as edges. Accordingly, complex model is dealing with searching a hidden graph (subgraph) in a given hypergraph H . Clearly, P is induced by the set of positive edges.
- The formidable complexes of eukaryotic DNA transcription and RNA translation could involve hundred of molecules.
- Application: protein-to-protein interaction (Many) (Lappe and Holm, 2004).
- (*) (Creating a protein-protein interaction network for

- A complex contains at most r clones! (Assumption)

(Generalized cover-free family)
Definition $(d, r; z]$ -disjunct Stinson and Wei, 1999

A matrix is $(d, r; z]$ -disjunct if for any $r+d$ columns $C_1, C_2, \dots, C_{r+d}$,

$$\left| \bigcap_{i=1}^r C_i \setminus \bigcup_{i=r+1}^{r+d} C_i \right| \geq z.$$

$$\begin{array}{c} \underbrace{C_1 \dots C_r}_{r \text{ columns}} \quad \underbrace{C_{r+1}, \dots, C_{r+d}}_{d \text{ columns}} \\ \geq z \rightarrow \begin{bmatrix} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \end{array}$$

Note that if $r=1$, then we use $(d; z]$ -disjunct for short.

→ Using a $(d, r; z]$ -disjunct matrix

Proposition For a negative complex X , there are at least z rows containing X but no positive complexes.

For a complex X ,

Proof $t_0^V(X) \stackrel{\text{def}}{=} |\cap X \setminus V|$ and $t_1^V(X) \stackrel{\text{def}}{=} |(\cap X) \cap V|$.

- We can use $(d, r; ze+1)$ -disjunct matrix to identify up-to- d positives with at most e errors.

(P3)

Threshold Group Testing [Damaschke, 2005]

Let l and u be nonnegative integers $l < u$, called the lower and upper threshold. A group test (threshold model) for a subset S of clones is positive if S contains at least u positives, and negative if at most l positives are present. Furthermore, if the number of positives in S is between l and u , the outcome can give an arbitrary answer, either positive or negative.

$g = u - l - 1$ is called the gap.

Note: Use $(d, n; z)$ -disjunct matrices to identify the positives with at most e errors. ($z \geq 2e+1$)

We have ^{to} stop somewhere! Thank you for your patience. ^^