Error Correcting Codes: Combinatorics, Algorithms and Applications (Spring 2010)

Lecture 10: Group Testing

February 05, 2010

Lecturer: Atri Rudra *Scribe: Devanshu Pandey*

1 An Overview

Robert Dorfman's paper in 1943 [**?**] can be considered to be the advent of what is called (Combinatorial) Group testing. This is the first in the series of approximately eight lectures that we will spend this semester exploring this very interesting application of Coding Theory. It must be noted that though this course covers Group testing as an application of Coding Theory, it took off long before Coding theory itself.

The original motivation arose during the Second World War when the United States Public Health service and the Selective Service embarked upon a large scale project. The objective was to weed out all syphilitic men called up for induction. [**?**] The naive way to do this would be to test each person individually, that is:

1. Draw sample from a given individual,
2. Perform required tests, then
3. Determine presence or absence of syphilis.

This method of one test per person will gives us a total of n steps for a total of n soldiers. Say we had more than $70 - 75\%$ of the soldier populatin infected. Only at such large numbers would the use of this method be reasonable. However, our goal is to achieve effective testing since it does not make sense to test $100,000$ people to get just 10 positives.

Here, we mention a property that we would like to use: *Property:* We can combine blood samples and test a combined sample together to chec if at least one soldier syphilis Simply put, say one has a very large number of items to test, and knows that only certain few will turn out positive, what is a nice and efficient way of testing? Note that it is important that we have an estimate of the number of possible positives. What we do not know is who among the group will have the infection.

2 Formalization of the problem

Input: The total number of soldiers: n, the number of infected soldiers: d. The input can also be described as a vector $X = (x_1, x_2, ..., x_n)$ where $x_i = 1$ if item i is infected else $x_i = 0$

Hamming Weight of x is defined as the number of 1s in x. Hence, $|x| \leq d$. This is a form of implicit input since we do not know the positions of 1s in the input. The only way to find out is to run the tests. Now, we will formalize the notion of a test.

Tests: A test $S \subseteq [n]$ is a Query/Test that returns:

$$Answer = \begin{cases} 1 & \text{if } \sum_{i \in S}^{n} x_i \geq 1; \\ 0 & \text{otherwise.} \end{cases}$$

Note that the addition operation used by the summation is the logical-OR ($\bigvee$)

Goal: Compute X and minimize the number of tests required to determine X.

This question boils down to one of *Combinatorial Searching*. Combinatorial searching in general can be explained as follows: Say you have a set of n variables and each of these can take on m possible values. So, finding possible solutions that match a certain constraint is a problem of combinatorial searching. The major problem with such questions is that the solution can grow exponentially in the size of the input. Here, we have no direct questions or answers. Any piece of information can only be obtained using an indirect query.

We present the definition of $t(d, n)$

Definition 2.1 ($t(d, n)$)**.** *Given a subset of n items with d defects, the minimum number of tests that one would have to make is defined as $t(d, n)$.*

Note that if we did this the naive way then $1 \leq t(d, n) \leq n$

3 Testing methods

Testing might be carried out in two ways.

1. Adaptive Testing is where we test a given subset of items, get their results and base our further tests on the outcome of the previous set of tests.

2. Non-Adaptive Testing on the otherhand is when all our tests are set even before we perform our first test. That is, all our tests are decided apriori.

Non-adaptive group testing is crucial for the Syphilis problem.

4 Representing the set of tests as a matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & . & . & . & a_1, i \\ - & - & - & - & - & - & - & - \\ & & & \chi_i & & & & \\ - & - & - & - & - & - & - & - \\ . & . & . & . & . & . & . & . \end{pmatrix}$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix}$$

$$R = \begin{pmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_n \end{pmatrix}$$

Here, $S \subseteq [n]$, $\chi_s \in 0, 1^n$ and $i \in S \Leftrightarrow \chi_s(i) = 1$ A is the $t \times n$ matrix of χ_i , X is our input vector transposed and R is the resultant. The relation to be established is $A \times X = R$. (Note that multiplication here is boolean-AND ($\bigcup$) and addition is boolean-OR ($\bigvee$)

Hence, for a say row 1, $r_1 =< x, \chi_t est1 >= \bigvee_{i \in test_i} x_i 1$

Our goal is to get to X from R with as small a t as possible. Note that a trivial solution would just be the $n \times n$ identity matrix.

In the next lecture we will se that $t(d, n) \geq \Omega(d * log(n))$ and will also show that this is the best we know about getting a lower bound for the number of Non-adaptive testing case.