Group Testing:
From Syphilis to Sparse Fourier Transforms

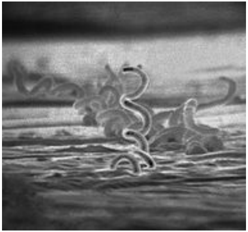
Mark Iwen

Michigan State University

November 6, 2013

History of Group Testing

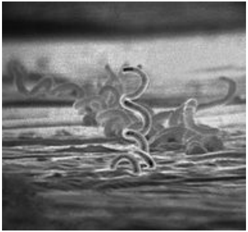
- Syphilis Testing [Dorfman 1943]



- Mix many recruits' blood samples together and test the mixture!

History of Group Testing

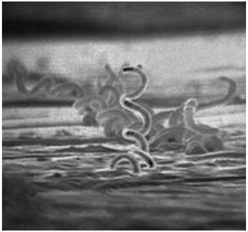
- Syphilis Testing [Dorfman 1943]



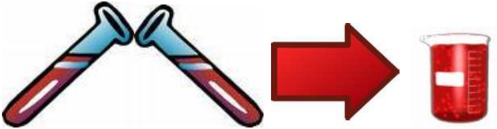
- Mix many recruits' blood samples together and test the mixture!

History of Group Testing

- Syphilis Testing [Dorfman 1943]



- Mix many recruits' blood samples together and test the mixture!



Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK

Group Testing Example

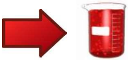
- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK

Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK

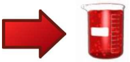
Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK



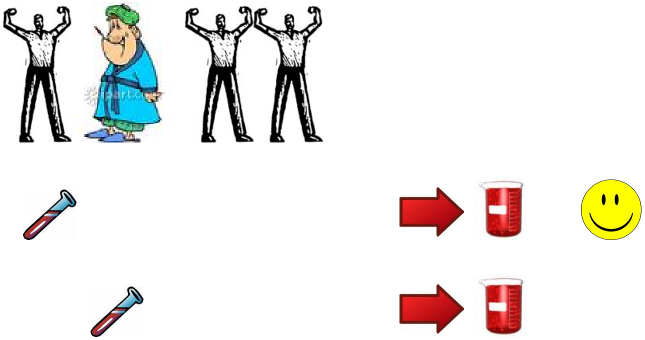
Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK



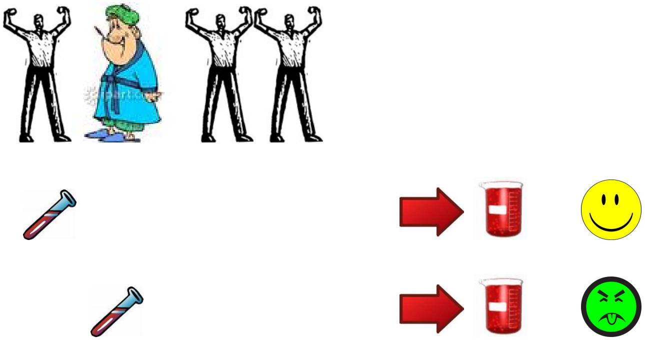
Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK



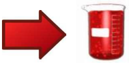
Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK



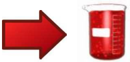
Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK



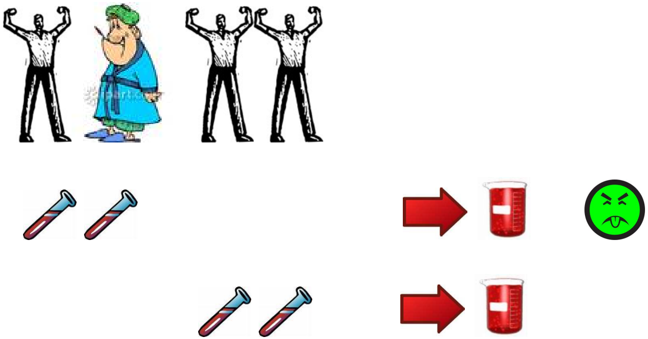
Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK



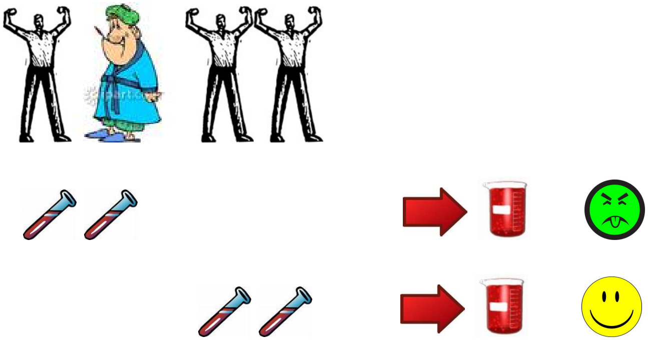
Group Testing Example

- Line up our recruits.
- Let's see how testing turns out IF WE KNOW WHO IS SICK



Group Testing Example

- Line up our recruits.
- Let’s see how testing turns out IF WE KNOW WHO IS SICK



Using Group Testing to Find One Sick Person

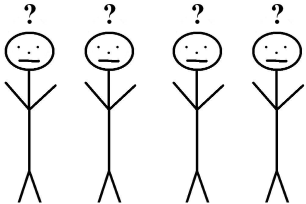
- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.

Using Group Testing to Find One Sick Person

- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.

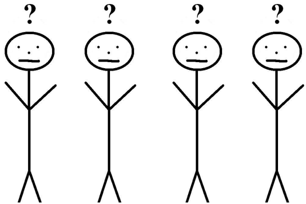
Using Group Testing to Find One Sick Person

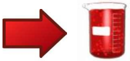
- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.



Using Group Testing to Find One Sick Person

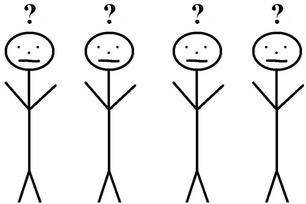
- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.





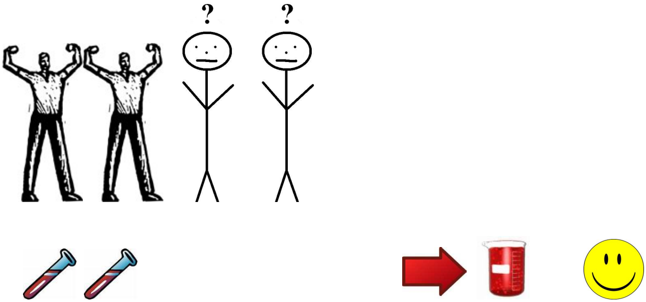
Using Group Testing to Find One Sick Person

- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.



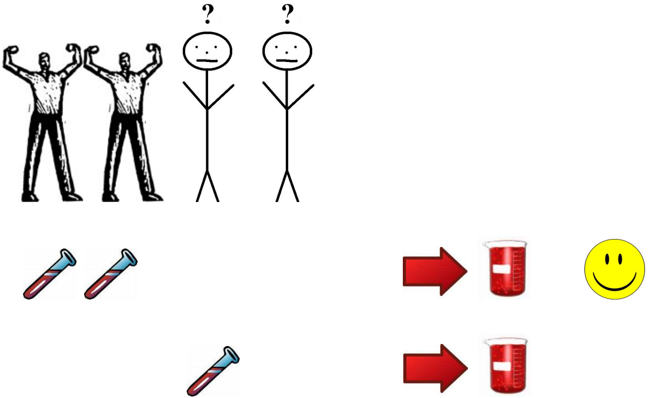
Using Group Testing to Find One Sick Person

- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.



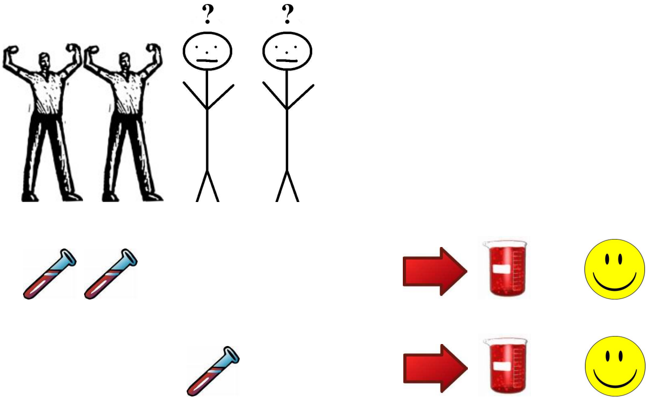
Using Group Testing to Find One Sick Person

- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.



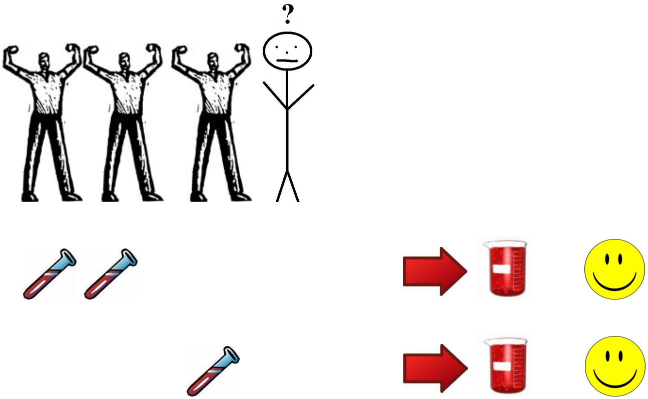
Using Group Testing to Find One Sick Person

- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.



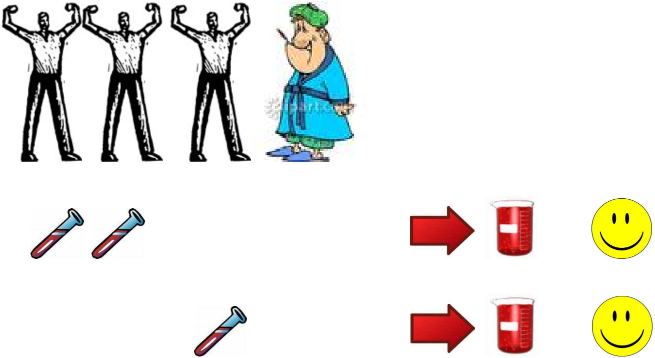
Using Group Testing to Find One Sick Person

- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.



Using Group Testing to Find One Sick Person

- Line up our recruits.
- We will use TWO TESTS to find the ONE sick person.
- Since we know there is one sick person, it must be the last one!



Find One Sick Sample Hidden in Three Healthy Ones

1. Line up the four samples.
2. Mix tests from the first two samples together and test them.
3. IF this first test is , THEN these first two samples are healthy. OTHERWISE, if the test is , the last two samples are healthy.

   WE SHOULD NOW ONLY HAVE TWO UNKNOWN SAMPLES!

4. Pick one of the two remaining unknown samples and test it.
5. IF this test is , THEN the sample we **didn't** test is sick. OTHERWISE, if the test is , the sample we **did** test is sick.

Find One Sick Sample Hidden in Three Healthy Ones

1. Line up the four samples.
2. Mix tests from the first two samples together and test them.
3. IF this first test is , THEN these first two samples are healthy. OTHERWISE, if the test is , the last two samples are healthy.

   WE SHOULD NOW ONLY HAVE TWO UNKNOWN SAMPLES!

4. Pick one of the two remaining unknown samples and test it.
5. IF this test is , THEN the sample we **didn't** test is sick. OTHERWISE, if the test is , the sample we **did** test is sick.

Find One Sick Sample Hidden in Three Healthy Ones

1. Line up the four samples.
2. Mix tests from the first two samples together and test them.
3. IF this first test is , THEN these first two samples are healthy. OTHERWISE, if the test is , the last two samples are healthy.

WE SHOULD NOW ONLY HAVE TWO UNKNOWN SAMPLES!

4. Pick one of the two remaining unknown samples and test it.
5. IF this test is , THEN the sample we **didn't** test is sick. OTHERWISE, if the test is , the sample we **did** test is sick.

Find One Sick Sample Hidden in Three Healthy Ones

1. Line up the four samples.
2. Mix tests from the first two samples together and test them.
3. IF this first test is , THEN these first two samples are healthy. OTHERWISE, if the test is , the last two samples are healthy.

WE SHOULD NOW ONLY HAVE TWO UNKNOWN SAMPLES!

4. Pick one of the two remaining unknown samples and test it.
5. IF this test is , THEN the sample we **didn't** test is sick. OTHERWISE, if the test is , the sample we **did** test is sick.

Find One Sick Sample Hidden in Three Healthy Ones

1. Line up the four samples.
2. Mix tests from the first two samples together and test them.
3. IF this first test is 🙂, THEN these first two samples are healthy. OTHERWISE, if the test is 🤢, the last two samples are healthy.

WE SHOULD NOW ONLY HAVE TWO UNKNOWN SAMPLES!

4. Pick one of the two remaining unknown samples and test it.
5. IF this test is , THEN the sample we **didn't** test is sick. OTHERWISE, if the test is , the sample we **did** test is sick.

Find One Sick Sample Hidden in Three Healthy Ones

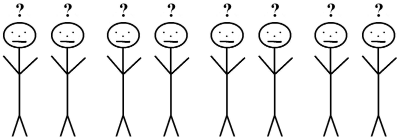
1. Line up the four samples.
2. Mix tests from the first two samples together and test them.
3. IF this first test is , THEN these first two samples are healthy. OTHERWISE, if the test is , the last two samples are healthy.

WE SHOULD NOW ONLY HAVE TWO UNKNOWN SAMPLES!

4. Pick one of the two remaining unknown samples and test it.
5. IF this test is , THEN the sample we **didn't** test is sick. OTHERWISE, if the test is , the sample we **did** test is sick.

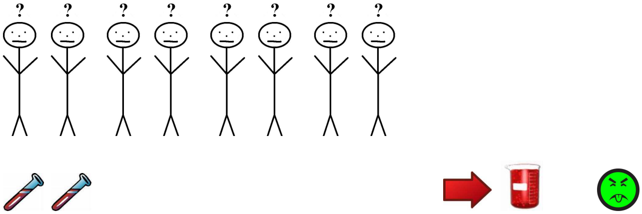
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



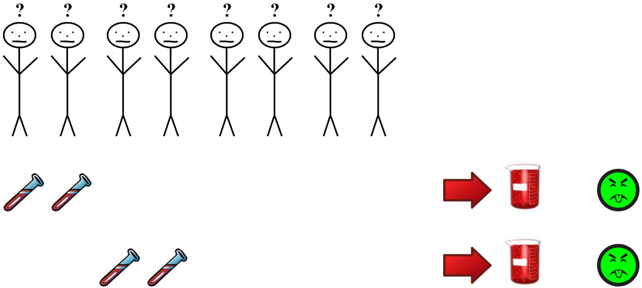
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



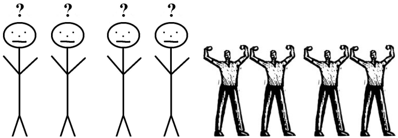
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



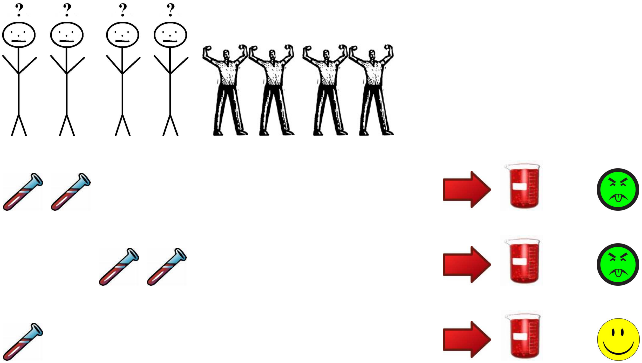
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



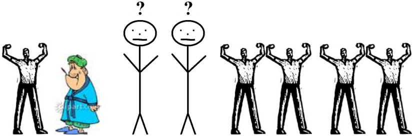
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



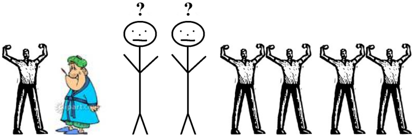
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



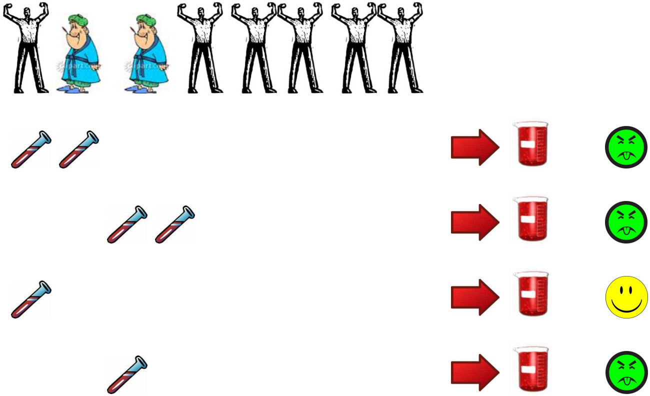
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



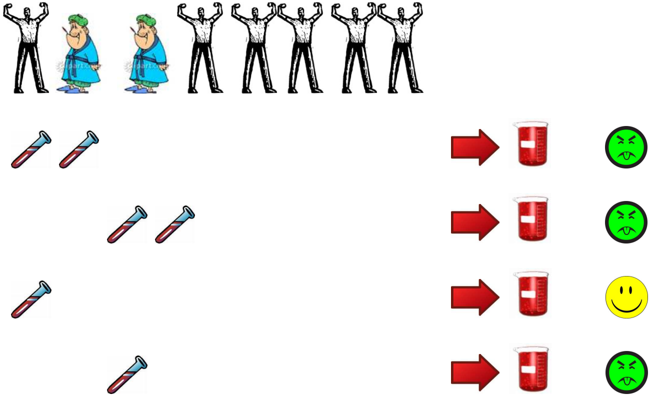
We Can Also Find More Hidden Sick Samples...

- What if we know we have TWO sick recruits?



We Can Also Find More Hidden Sick Samples...

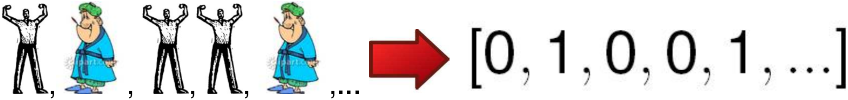
- What if we know we have TWO sick recruits?



- Can you generalize this further?

Group Testing - Overview

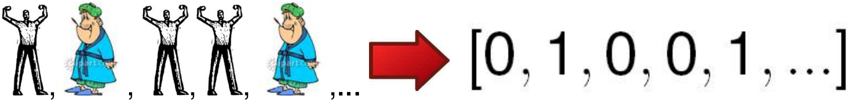
- Encode the problem in a binary array



- Find the nonzero entries by testing subsets of the array
  - Boolean $K \times N$ measurement matrix $\mathcal{M}$
  - Boolean array $\mathbf{a} \in \{0,1\}^N$ containing k ones
  - All arithmetic Boolean (+ = OR, * = AND)
  - Identify the location of k ones using $\mathbf{y} = \mathcal{M}\mathbf{a}$ measurements
  - How small can we make K and still recover $\mathbf{a}$ using $\mathbf{y}$?

Group Testing - Overview

- Encode the problem in a binary array

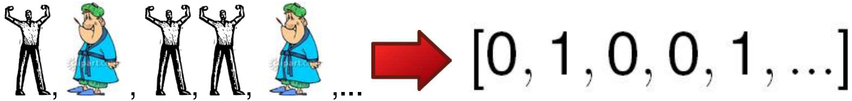


- Find the nonzero entries by testing subsets of the array

- Boolean $K \times N$ measurement matrix $\mathcal{M}$
- Boolean array $\mathbf{a} \in \{0, 1\}^N$ containing k ones
- All arithmetic Boolean (+ = OR, * = AND)
- Identify the location of k ones using $\mathbf{y} = \mathcal{M}\mathbf{a}$ measurements
- How small can we make K and still recover $\mathbf{a}$ using $\mathbf{y}$?

Group Testing - Overview

- Encode the problem in a binary array

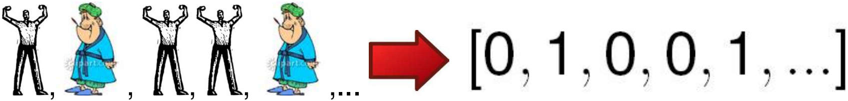


- Find the nonzero entries by testing subsets of the array

- Boolean $K \times N$ measurement matrix $\mathcal{M}$
- Boolean array $\mathbf{a} \in \{0,1\}^N$ containing k ones
- All arithmetic Boolean (+ = OR, * = AND)
- Identify the location of k ones using $\mathbf{y} = \mathcal{M}\mathbf{a}$ measurements
- How small can we make K and still recover $\mathbf{a}$ using $\mathbf{y}$?

Group Testing - Overview

- Encode the problem in a binary array



- Find the nonzero entries by testing subsets of the array

  - Boolean $K \times N$ measurement matrix $\mathcal{M}$
  - Boolean array $\mathbf{a} \in \{0, 1\}^N$ containing k ones
  - All arithmetic Boolean (+ = OR, * = AND)
  - Identify the location of k ones using $\mathbf{y} = \mathcal{M}\mathbf{a}$ measurements
  - How small can we make K and still recover $\mathbf{a}$ using $\mathbf{y}$?

Group Testing - Overview

- Encode the problem in a binary array
- Find the nonzero entries by testing subsets of the array

- Boolean $K \times N$ measurement matrix $\mathcal{M}$
- Boolean array $\mathbf{a} \in \{0, 1\}^N$ containing k ones
- All arithmetic Boolean (+ = OR, * = AND)
- Identify the location of k ones using $\mathbf{y} = \mathcal{M}\mathbf{a}$ measurements
- How small can we make K and still recover $\mathbf{a}$ using $\mathbf{y}$?

Group Testing - Overview

- Encode the problem in a binary array
- Find the nonzero entries by testing subsets of the array

- Boolean $K \times N$ measurement matrix $\mathcal{M}$
- Boolean array $\mathbf{a} \in \{0, 1\}^N$ containing k ones
- All arithmetic Boolean (+ = OR, * = AND)
- Identify the location of k ones using $\mathbf{y} = \mathcal{M}\mathbf{a}$ measurements
- How small can we make K and still recover $\mathbf{a}$ using $\mathbf{y}$?

Group Testing - Overview

- Encode the problem in a binary array
- Find the nonzero entries by testing subsets of the array

- Boolean $K \times N$ measurement matrix $\mathcal{M}$
- Boolean array $\mathbf{a} \in \{0,1\}^N$ containing k ones
- All arithmetic Boolean (+ = OR, * = AND)
- Identify the location of k ones using $\mathbf{y} = \mathcal{M}\mathbf{a}$ measurements
- How small can we make K and still recover $\mathbf{a}$ using $\mathbf{y}$?

Adaptive Group Testing

- What if we can adaptively sample $\mathbf{a} \in \{0, 1\}^N$ several times, how many tests do we need to find its k (or fewer) nonzero entries?

- ANSWER: We can use at most $\log(N)$ matrices with at most $2k + 1$ rows each! The total number of inner products is only $O(k \log N)$!

Adaptive Group Testing

- What if we can adaptively sample $\mathbf{a} \in \{0,1\}^N$ several times, how many tests do we need to find its k (or fewer) nonzero entries?

- ANSWER: We can use at most $\log(N)$ matrices with at most $2k+1$ rows each! The total number of inner products is only $O(k \log N)$!

Group Testing - Nonadaptive Example

- $\mathcal{M}$ is 5×30, **a** contains 1 nonzero entry.

$$\begin{matrix} 0^{\text{th}} \text{ bit} \\ 1^{\text{st}} \text{ bit} \\ 2^{\text{nd}} \text{ bit} \\ 3^{\text{rd}} \text{ bit} \\ 4^{\text{th}} \text{ bit} \end{matrix} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix}$$

Group Testing - Nonadaptive Example

- $\mathcal{M}$ is 5×30, **a** contains 1 nonzero entry.

$$\begin{matrix} 0^{\text{th}} \text{ bit} \\ 1^{\text{st}} \text{ bit} \\ 2^{\text{nd}} \text{ bit} \\ 3^{\text{rd}} \text{ bit} \\ 4^{\text{th}} \text{ bit} \end{matrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix}$$

Group Testing - Nonadaptive Example

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} \Leftarrow 0^{\text{th}} \text{ bit } = 1 \\ \Leftarrow 1^{\text{st}} \text{ bit } = 1 \\ \Leftarrow 2^{\text{nd}} \text{ bit } = 0 \\ \Leftarrow 3^{\text{rd}} \text{ bit } = 0 \\ \Leftarrow 4^{\text{th}} \text{ bit } = 0 \end{matrix}$$

- Recovery is simple: The result is the position of 1 in binary.
- QUIZ: Can we do better if we let our measurement matrix contains arbitrarily large integers?

Group Testing - Nonadaptive Example

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} \Leftarrow 0^{\text{th}} \text{ bit } = 1 \\ \Leftarrow 1^{\text{st}} \text{ bit } = 1 \\ \Leftarrow 2^{\text{nd}} \text{ bit } = 0 \\ \Leftarrow 3^{\text{rd}} \text{ bit } = 0 \\ \Leftarrow 4^{\text{th}} \text{ bit } = 0 \end{matrix}$$

- Recovery is simple: The result is the position of 1 in binary.
- QUIZ: Can we do better if we let our measurement matrix contains arbitrarily large integers?

Group Testing - Nonadaptive Example

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} \Leftarrow 0^{\text{th}} \text{ bit } = 1 \\ \Leftarrow 1^{\text{st}} \text{ bit } = 1 \\ \Leftarrow 2^{\text{nd}} \text{ bit } = 0 \\ \Leftarrow 3^{\text{rd}} \text{ bit } = 0 \\ \Leftarrow 4^{\text{th}} \text{ bit } = 0 \end{matrix}$$

- Recovery is simple: The result is the position of 1 in binary.
- QUIZ: Can we do better if we let our measurement matrix contains arbitrarily large integers?

Group Testing - Nonadaptive Example

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \dots \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} = 3$$

- Recovery is simple: The result is the position of 1 in binary.
- QUIZ: Can we do better if we let our measurement matrix contains arbitrarily large integers?
- YES!!!

Nonadaptive Group Testing: >1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k-strongly selective* if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

Nonadaptive Group Testing: >1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k-strongly selective* if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

0	1	1	0	0	1	0
0	1	0	1	0	1	0
0	0	1	0	0	1	0
1	0	0	1	0	0	1
1	0	0	0	1	0	1

Nonadaptive Group Testing: >1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k-strongly selective* if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

0	1	1	0	0	1	0
0	1	0	1	0	1	0
0	0	1	0	0	1	0
1	0	0	1	0	0	1
1	0	0	0	1	0	1

Nonadaptive Group Testing: > 1 Sick Person

Measurement Matrix Construction

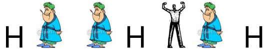
A binary matrix $\mathcal{M}$ is *k-strongly selective* if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

0	1	1	0	0	1	0
0	1	0	1	0	1	0
0	0	1	0	0	1	0
1	0	0	1	0	0	1
1	0	0	0	1	0	1

Nonadaptive Group Testing: $>$ 1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k-strongly selective* if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.



0	1	1	0	0	1	0
0	1	0	1	0	1	0
0	0	1	0	0	1	0
1	0	0	1	0	0	1
1	0	0	0	1	0	1

Nonadaptive Group Testing: $>$ 1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k-strongly selective* if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

0	1	1	0	0	1	0
0	1	0	1	0	1	0
0	0	1	0	0	1	0
$*$	$*$	$*$	$*$	$*$	$*$	$*$
1	0	0	0	1	0	1

Nonadaptive Group Testing: >1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k*-strongly selective if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

H			H	H		H
0	1	1	0	0	1	0
0	1	0	1	0	1	0
0	0	1	0	0	1	0
$*$	$*$	$*$	$*$	$*$	$*$	$*$
1	0	0	0	1	0	1

Nonadaptive Group Testing: >1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k*-strongly selective if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

- Simple Recovery: For each k-strongly selective test that evaluates to a 0 (i.e., All Healthy)...

Nonadaptive Group Testing: >1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k*-strongly selective if for any column, $\mathbf{x}$, and subset of columns containing at most k elements, X, there exists a row in $\mathcal{M}$ with a 1 in column $\mathbf{x}$ and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

- Simple Recovery: For each k-strongly selective test that evaluates to a 0 (i.e., All Healthy)...
- Mark all individuals tested in that test as Healthy.

Nonadaptive Group Testing: $>$ 1 Sick Person

Measurement Matrix Construction

A binary matrix $\mathcal{M}$ is *k*-strongly selective if for any column, **x**, and subset of columns containing at most *k* elements, *X*, there exists a row in $\mathcal{M}$ with a 1 in column **x** and zeros in all of the other $X - \{\mathbf{x}\}$ columns.

- Simple Recovery: For each *k*-strongly selective test that evaluates to a 0 (i.e., All Healthy)...
- Mark all individuals tested in that test as Healthy.
- If there are at most *k* sick individuals, we will find them all!

Nonadaptive Group Testing: $>$ 1 Sick Person

Theorem 1

Let $\mathbf{a} \in \{0,1\}^N$ be a binary vector containing k nonzero entries. Furthermore, let $\mathcal{M}$ be a k-strongly selective binary matrix. Then, the positions of all k nonzero entries in $\mathbf{a}$ can be recovered using only the result of $\mathcal{M}\mathbf{a}$.

Theorem 2

There exist explicitly constructible $\left(\min\{k^2 \cdot \log N, N\}\right) \times N$ k-strongly selective binary matrices. And, they are optimal in the number of rows.[a]

[a] See Porat and Rothschild's paper "Explicit Non-Adaptive Combinatorial Group Testing Schemes".

Nonadaptive Group Testing: >1 Sick Person

Theorem 1

Let $\mathbf{a} \in \{0,1\}^N$ be a binary vector containing k nonzero entries. Furthermore, let $\mathcal{M}$ be a k-strongly selective binary matrix. Then, the positions of all k nonzero entries in $\mathbf{a}$ can be recovered using only the result of $\mathcal{M}\mathbf{a}$.

Theorem 2

There exist explicitly constructible $\left(\min\{k^2 \cdot \log N, N\}\right) \times N$ k-strongly selective binary matrices. And, they are optimal in the number of rows.[a]

[a] See Porat and Rothschild's paper "Explicit Non-Adaptive Combinatorial Group Testing Schemes".

Group Testing - Another example

Error Detection

Suppose we want to transmit a binary vector $\mathbf{a} \in \{0,1\}^N$ through a noisy environment. How can we tell if we received the real message?

- Used for DVD, CD, and other media devices in your house!
- Basic Methods: Parity and Checksums
- A bit stronger: Use a strongly selective matrix!
  - Transmit (or read) both $\mathbf{a}$ and $\mathcal{M}\mathbf{a}$
  - The receiver gets (or reads) $\mathbf{a}' = \mathbf{a} + \epsilon$
  - Check to see if $\mathcal{M}\mathbf{a} = \mathcal{M}\mathbf{a}'$

Group Testing - Another example

Error Detection

Suppose we want to transmit a binary vector $\mathbf{a} \in \{0,1\}^N$ through a noisy environment. How can we tell if we received the real message?

- Used for DVD, CD, and other media devices in your house!
- Basic Methods: Parity and Checksums
- A bit stronger: Use a strongly selective matrix!
  - Transmit (or read) both $\mathbf{a}$ and $\mathcal{M}\mathbf{a}$
  - The receiver gets (or reads) $\mathbf{a}' = \mathbf{a} + \epsilon$
  - Check to see if $\mathcal{M}\mathbf{a} = \mathcal{M}\mathbf{a}'$

Group Testing - Another example

Error Detection

Suppose we want to transmit a binary vector $\mathbf{a} \in \{0,1\}^N$ through a noisy environment. How can we tell if we received the real message?

- Used for DVD, CD, and other media devices in your house!
- Basic Methods: Parity and Checksums
- A bit stronger: Use a strongly selective matrix!
  - Transmit (or read) both $\mathbf{a}$ and $\mathcal{M}\mathbf{a}$
  - The receiver gets (or reads) $\mathbf{a}' = \mathbf{a} + \epsilon$
  - Check to see if $\mathcal{M}\mathbf{a} = \mathcal{M}\mathbf{a}'$

Group Testing - Another example

Error Detection

Suppose we want to transmit a binary vector $\mathbf{a} \in \{0, 1\}^N$ through a noisy environment. How can we tell if we received the real message?

- Used for DVD, CD, and other media devices in your house!
- Basic Methods: Parity and Checksums
- A bit stronger: Use a strongly selective matrix!
  - Transmit (or read) both $\mathbf{a}$ and $\mathcal{M}\mathbf{a}$
  - The receiver gets (or reads) $\mathbf{a}' = \mathbf{a} + \epsilon$
  - Check to see if $\mathcal{M}\mathbf{a} = \mathcal{M}\mathbf{a}'$

Group Testing - Another example

Error Detection

Suppose we want to transmit a binary vector $\mathbf{a} \in \{0,1\}^N$ through a noisy environment. How can we tell if we received the real message?

- Used for DVD, CD, and other media devices in your house!
- Basic Methods: Parity and Checksums
- A bit stronger: Use a strongly selective matrix!
  - Transmit (or read) both $\mathbf{a}$ and $\mathcal{M}\mathbf{a}$
  - The receiver gets (or reads) $\mathbf{a}' = \mathbf{a} + \epsilon$
  - Check to see if $\mathcal{M}\mathbf{a} = \mathcal{M}\mathbf{a}'$

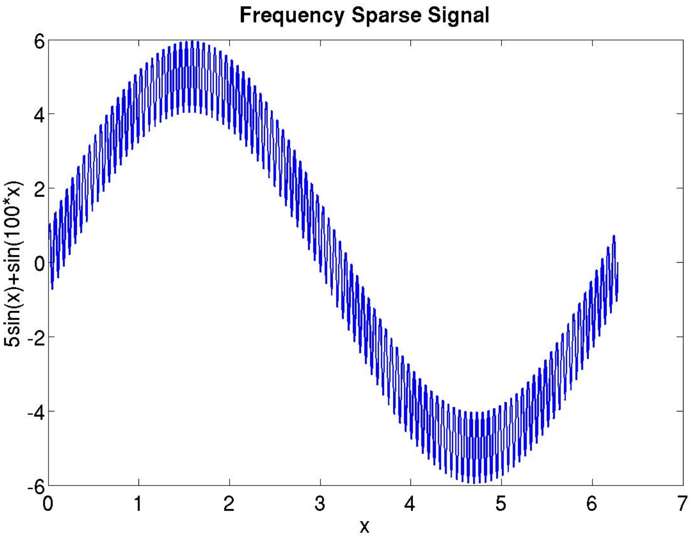
Group Testing - Another example

Error Detection

Suppose we want to transmit a binary vector $\mathbf{a} \in \{0,1\}^N$ through a noisy environment. How can we tell if we received the real message?

- Used for DVD, CD, and other media devices in your house!
- Basic Methods: Parity and Checksums
- A bit stronger: Use a strongly selective matrix!
  - Transmit (or read) both $\mathbf{a}$ and $\mathcal{M}\mathbf{a}$
  - The receiver gets (or reads) $\mathbf{a}' = \mathbf{a} + \epsilon$
  - Check to see if $\mathcal{M}\mathbf{a} = \mathcal{M}\mathbf{a}'$

The Goal: Sparse Signal Recovery



Where Do Fourier Sparse Signals Appear?

Motivated by

Applications involving wideband signals that are locally frequency sparse in time [see work by Baranuik, Duarte, Romberg, Tropp, ...].

- Frequency hopping modulation schemes [Lamarr et al., 1941]
- The inverse: Medical Imaging, ...

Where Do Fourier Sparse Signals Appear?

Motivated by

Applications involving wideband signals that are locally frequency sparse in time [see work by Baranuik, Duarte, Romberg, Tropp, ...].

- Frequency hopping modulation schemes [Lamarr et al., 1941]
- The inverse: Medical Imaging, ...

Where Do Fourier Sparse Signals Appear?

Motivated by

Applications involving wideband signals that are locally frequency sparse in time [see work by Baranuik, Duarte, Romberg, Tropp, ...].

- Frequency hopping modulation schemes [Lamarr et al., 1941]
- The inverse: Medical Imaging, ...

Inherent Sparsity Example: Angiography [Lustig et al., 2007]

Problem Setup

Recover $f : [0, 2\pi] \mapsto \mathbb{C}$ consisting of k trigonometric terms

$$f(x) = \sum_{j=1}^{k} C_j \cdot e^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left[1 - \frac{N}{2}, \frac{N}{2}\right]$$

- Computationally efficient recovery...
- Use as few samples from f as possible.
- And, simple sampling patterns...
- We prefer strong recovery guarantees...

Problem Setup

Recover $f : [0, 2\pi] \mapsto \mathbb{C}$ consisting of k trigonometric terms

$$f(x) = \sum_{j=1}^{k} C_j \cdot e^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \dots, \omega_k\} \subset \left[1 - \frac{N}{2}, \frac{N}{2}\right]$$

- Computationally efficient recovery...
- Use as few samples from f as possible.
- And, simple sampling patterns...
- We prefer strong recovery guarantees...

Standard Solution: Trigonometric Interpolation

Recover

$$f(x) = \sum_{j=1}^{k} C_j \cdot e^{x \cdot \omega_j \cdot \mathbb{i}}, \ \Omega = \{\omega_1, \ldots, \omega_k\} \subset \left[1 - \frac{N}{2}, \frac{N}{2}\right]$$

- Take N equally spaced samples

$$f(0), f\left(2\pi/N\right), \ldots, f\left(2\pi(N-1)/N\right)$$

- Take an FFT of the samples in $O(N \cdot \log N)$ time.
- Locate k non-zero Fourier coefficients.

Doesn't Take Sparsity Into Account...

Standard Solution: Trigonometric Interpolation

Recover

$$f(x) = \sum_{j=1}^{k} C_j \cdot e^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left[1 - \frac{N}{2}, \frac{N}{2}\right]$$

- Take N equally spaced samples

$$f(0), f\left(2\pi/N\right), \dots, f\left(2\pi(N-1)/N\right)$$

- Take an FFT of the samples in $O(N \cdot \log N)$ time.
- Locate k non-zero Fourier coefficients.

Doesn't Take Sparsity Into Account...

Standard Solution: Trigonometric Interpolation

Recover

$$f(x) = \sum_{j=1}^{k} C_j \cdot e^{x \cdot \omega_j \cdot \mathbb{i}}, \; \Omega = \{\omega_1, \dots, \omega_k\} \subset \left[1 - \frac{N}{2}, \frac{N}{2}\right]$$

- Take N equally spaced samples

$$f(0), f\left(2\pi/N\right), \dots, f\left(2\pi(N-1)/N\right)$$

- Take an FFT of the samples in $O(N \cdot \log N)$ time.
- Locate k non-zero Fourier coefficients.

Doesn't Take Sparsity Into Account...

Group Testing - Example

- $\mathcal{M}$ is 5×6, $\vec{\mathbf{a}}$ contains 1 nonzero entry.

$$\begin{matrix} \equiv 0 \bmod 2 \\ \equiv 1 \bmod 2 \\ \equiv 0 \bmod 3 \\ \equiv 1 \bmod 3 \\ \equiv 2 \bmod 3 \end{matrix} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE TEST!

Group Testing - Example

- $\mathcal{M}$ is 5×6, $\vec{\mathbf{a}}$ contains 1 nonzero entry.

$$\begin{matrix} \equiv 0 \bmod 2 \\ \equiv 1 \bmod 2 \\ \equiv 0 \bmod 3 \\ \equiv 1 \bmod 3 \\ \equiv 2 \bmod 3 \end{matrix} \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE TEST!

Group Testing - Example

- $\mathcal{M}$ is 5×6, $\vec{\mathbf{a}}$ contains 1 nonzero entry.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix} \begin{matrix} \Leftarrow \text{ Index } \equiv 0 \bmod 2 \\ \\ \\ \\ \Leftarrow \text{ Index } \equiv 2 \bmod 3 \end{matrix}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE TEST!

Group Testing - Example

- $\mathcal{M}$ is 5×6, $\vec{\mathbf{a}}$ contains 1 nonzero entry.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix} \begin{matrix} \Leftarrow \text{ Index } \equiv 0 \bmod 2 \\ \\ \\ \\ \Leftarrow \text{ Index } \equiv 2 \bmod 3 \end{matrix}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE TEST!

Group Testing - Example

- $\mathcal{M}$ is 5×6, $\vec{\mathbf{a}}$ contains 1 nonzero entry.

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix} \begin{array}{l} \Leftarrow \text{ Index } \equiv 0 \bmod 2 \\ \\ \\ \\ \Leftarrow \text{ Index } \equiv 2 \bmod 3 \end{array}$$

- Reconstruct entry index via Chinese Remainder Theorem
- Two estimates of the entry's value

SAVED ONE TEST!

Group Testing - Fourier Example

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Group Testing - Fourier Example

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \mathcal{F}_{6\times 6}\mathcal{F}^{-1}_{6\times 6} \cdot \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Group Testing - Fourier Example

$$\left(\left(\begin{array}{cccccc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right) \mathcal{F}_{6\times 6} \right) \cdot \left(\mathcal{F}_{6\times 6}^{-1} \left(\begin{array}{c} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{array} \right) \right) = \left(\begin{array}{c} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{array} \right)$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Group Testing - Fourier Example

$$\begin{pmatrix} \sqrt{\frac{3}{2}} & 0 & 0 & \sqrt{\frac{3}{2}} & 0 & 0 \\ \sqrt{\frac{3}{2}} & 0 & 0 & -\sqrt{\frac{3}{2}} & 0 & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \end{pmatrix} \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Group Testing - Fourier Example

$$\begin{pmatrix} \sqrt{\frac{3}{2}} & 0 & 0 & \sqrt{\frac{3}{2}} & 0 & 0 \\ \sqrt{\frac{3}{2}} & 0 & 0 & -\sqrt{\frac{3}{2}} & 0 & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \\ * & 0 & * & 0 & * & 0 \end{pmatrix} \cdot \left(\mathcal{F}_{6\times 6}^{-1} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Group Testing - Fourier Example

$$\left(\begin{array}{c} \sqrt{3} \cdot \mathcal{F}_{2\times 2} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\ \sqrt{2} \cdot \mathcal{F}_{3\times 3} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \right) \cdot \left(\mathcal{F}_{6\times 6}^{-1} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Group Testing - Fourier Example

$$\left(\begin{array}{c} \sqrt{3} \cdot \mathcal{F}_{2\times 2} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\ \sqrt{2} \cdot \mathcal{F}_{3\times 3} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \right) \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Group Testing - Fourier Example

$$\begin{pmatrix} \sqrt{3} \cdot \mathcal{F}_{2\times 2} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\ \sqrt{2} \cdot \mathcal{F}_{3\times 3} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{pmatrix} \cdot \left(\mathcal{F}^{-1}_{6\times 6} \begin{pmatrix} 0 \\ 0 \\ 3.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right) = \begin{pmatrix} 3.5 \\ 0 \\ 0 \\ 0 \\ 3.5 \end{pmatrix}$$

- We only utilize 4 samples
- Computed Efficiently using 2 FFTs
- Reconstruct frequency index via Chinese Remainder Theorem
- Two estimates of nonzero Fourier coefficient

SAVED TWO SAMPLES!

Robustness: Nonlinear Approximation Guarantees

Theorem [I. '10]

Suppose $f : [0, 2\pi] \to \mathbb{C}$ has $\hat{f} \in \ell_1$ and $\hat{f}_k^{\mathrm{opt}}$ supported in $[-N/2, N/2]$. Then, we can deterministically approximate f by a k-term trigonometric polynomial, a, so that

$$\|f - a\|_2 \leq \left\|f - f_k^{\mathrm{opt}}\right\|_2 + \frac{\left\|\hat{f} - \hat{f}_k^{\mathrm{opt}}\right\|_1}{\sqrt{k}} + \epsilon_N$$

in $O\left(k^2 \cdot \log^4 N\right)$ time. Number of f samples used is $O\left(k^2 \cdot \log^4 N\right)$.

Monte Carlo Sparse Fourier Transform

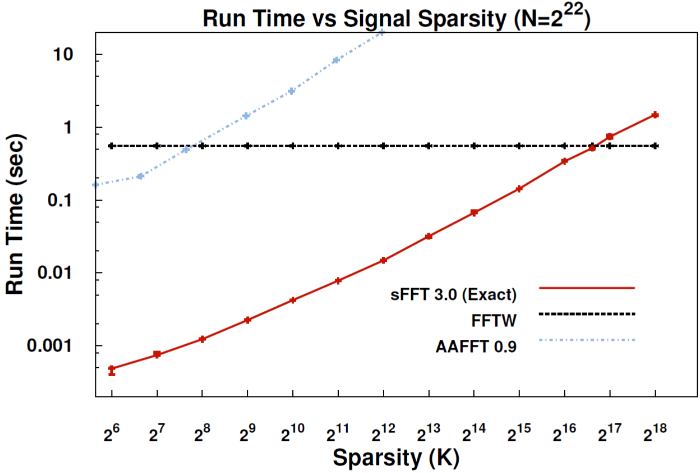
Theorem [I. '10]

Suppose $f : [0, 2\pi] \to \mathbb{C}$ has $\hat{f} \in \ell_1$ and $\hat{f}_k^{\text{opt}}$ supported in $[-N/2, N/2]$. If we run DSFT using $O\left(\log\left(\frac{N}{1-\sigma}\right)\right)$ randomly selected p_{q+j}-primes, then with probability at least σ we will approximate f by a k-term trigonometric polynomial, a, having

$$\|f - a\|_2 \leq \left\|f - f_k^{\text{opt}}\right\|_2 + \frac{\left\|\hat{f} - \hat{f}_k^{\text{opt}}\right\|_1}{\sqrt{k}} + \epsilon_N$$

in $O\left(k \cdot \log^4 N\right)$ time. Number of f samples used is $O\left(k \cdot \log^4 N\right)$.

Code of Hassanieh, Indyk, Katabi, and Price



Thank You!

Questions?