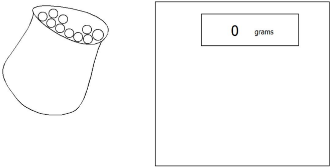
Making Do with Less: An Introduction to Compressed Sensing 1

Kurt Bryan

July 3, 2023

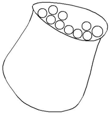
Ingredients



Given

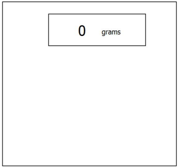
- A bag with 100 marbles, nominal mass 10 grams each. But one marble may be bad (off mass). It's not visually obvious.

Ingredients

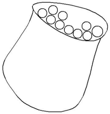


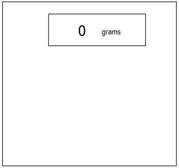
Given

- A bag with 100 marbles, nominal mass 10 grams each. But one marble may be bad (off mass). It's not visually obvious.
- An electronic balance.



Ingredients



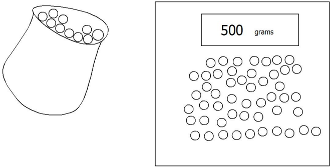


Given

- A bag with 100 marbles, nominal mass 10 grams each. But one marble may be bad (off mass). It's not visually obvious.
- An electronic balance.

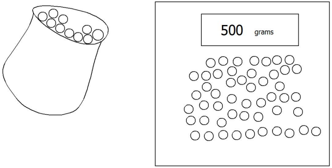
How can we find the bad marble with the fewest weighings?

Divide and Conquer



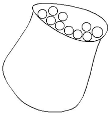
- Put half the marbles on the scale.

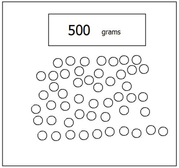
Divide and Conquer



- Put half the marbles on the scale.
- Decide if the bad marble is in that subset.

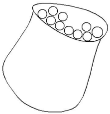
Divide and Conquer

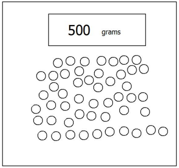




- Put half the marbles on the scale.
- Decide if the bad marble is in that subset.
- Repeat with the subset that (might) contain the bad marble.

Divide and Conquer





- Put half the marbles on the scale.
- Decide if the bad marble is in that subset.
- Repeat with the subset that (might) contain the bad marble.

We find the bad marble in about $\log_2(100) \approx 7$ weighings, a big improvement over the expected 50.5 weighings for a sequential approach.

Two or More Bad Marbles

- If there are $k \geq 2$ bad marbles (of unknown weights) the problem gets a lot harder.

Two or More Bad Marbles

- If there are $k \geq 2$ bad marbles (of unknown weights) the problem gets a lot harder.
- The sequential approach takes an expected $101k/(k+1)$ weighings if there are k bad marbles.

Two or More Bad Marbles

- If there are $k \geq 2$ bad marbles (of unknown weights) the problem gets a lot harder.
- The sequential approach takes an expected $101k/(k+1)$ weighings if there are k bad marbles.
- And if we don't know k, we don't know when we're done.

Two or More Bad Marbles

- If there are $k \geq 2$ bad marbles (of unknown weights) the problem gets a lot harder.
- The sequential approach takes an expected $101k/(k+1)$ weighings if there are k bad marbles.
- And if we don't know k, we don't know when we're done.
- To do better we need to weigh subsets—but how should be choose them?

Two or More Bad Marbles

- If there are $k \geq 2$ bad marbles (of unknown weights) the problem gets a lot harder.
- The sequential approach takes an expected $101k/(k+1)$ weighings if there are k bad marbles.
- And if we don't know k, we don't know when we're done.
- To do better we need to weigh subsets—but how should be choose them?
- Randomly!

Random Subsets

- Number the marbles 1 to 100.

Random Subsets

- Number the marbles 1 to 100.
- Choose a random subset of the marbles, of about size 50, and weigh them.

Random Subsets

- Number the marbles 1 to 100.
- Choose a random subset of the marbles, of about size 50, and weigh them.
- Repeat the last step a total of n times, about $n = 25$.

Random Subsets

- Number the marbles 1 to 100.
- Choose a random subset of the marbles, of about size 50, and weigh them.
- Repeat the last step a total of n times, about $n = 25$.

Claim: If there are only a few bad marbles (say $k \leq 5$) and we weigh $n = 25$ subsets, we almost certainly have enough information to identify the bad marbles.

A Smaller Example

Suppose there are only 10 marbles, nominal mass 10 grams each. Let x_i be the deviation of the ith marble from nominal. We expect $x_i = 0$ for most i.

A Smaller Example

Suppose there are only 10 marbles, nominal mass 10 grams each. Let x_i be the deviation of the ith marble from nominal. We expect $x_i = 0$ for most i.

- For example, suppose $x_3 = -0.3, x_9 = 0.44$, and all other $x_i = 0$ (but we don't know this).

A Smaller Example

Suppose there are only 10 marbles, nominal mass 10 grams each. Let x_i be the deviation of the ith marble from nominal. We expect $x_i = 0$ for most i.

- For example, suppose $x_3 = -0.3, x_9 = 0.44$, and all other $x_i = 0$ (but we don't know this).
- Choose a random subset of the marbles, say marbles $1, 2, 3, 5, 7, 9$. Weigh the subset. In this case we obtain 60.14 grams. We can conclude that

$$x_1 + x_2 + x_3 + x_5 + x_7 + x_9 = 0.14.$$

A Smaller Example

Suppose there are only 10 marbles, nominal mass 10 grams each. Let x_i be the deviation of the ith marble from nominal. We expect $x_i = 0$ for most i.

- For example, suppose $x_3 = -0.3, x_9 = 0.44$, and all other $x_i = 0$ (but we don't know this).
- Choose a random subset of the marbles, say marbles $1, 2, 3, 5, 7, 9$. Weigh the subset. In this case we obtain 60.14 grams. We can conclude that

$$x_1 + x_2 + x_3 + x_5 + x_7 + x_9 = 0.14.$$

- Repeat with a few more randomly chosen subsets.

A Small Example

Suppose we choose four more subsets, say $\{5, 6, 7, 9\}$, $\{1, 3, 4, 5, 7, 8\}$, $\{3, 6, 8, 9\}$, and $\{3, 5, 6, 7, 8, 10\}$. The respective weighings yield $40.44, 59.7, 40.14, 59.7$.

A Small Example

Suppose we choose four more subsets, say $\{5, 6, 7, 9\}$, $\{1, 3, 4, 5, 7, 8\}$, $\{3, 6, 8, 9\}$, and $\{3, 5, 6, 7, 8, 10\}$. The respective weighings yield $40.44, 59.7, 40.14, 59.7$. We can conclude that

$$\begin{aligned}
x_1 + x_2 + x_3 + x_5 + x_7 + x_9 &= 0.14 \\
x_5 + x_6 + x_7 + x_9 &= 0.44 \\
x_1 + x_3 + x_4 + x_5 + x_7 + x_8 &= -0.30 \\
x_3 + x_6 + x_8 + x_9 &= 0.14 \\
x_3 + x_5 + x_6 + x_7 + x_8 + x_{10} &= -0.30.
\end{aligned}$$

This is a system of 5 linear equations in 10 unknowns. There are many infinitely many solutions!

Brute Force For Finding Sparse Solutions

We could try to find a 1-sparse solution (we assume there's only one bad marble). In our example we might try setting all $x_i = 0$ for $2 \leq i \leq 10$, then trying to solve the equations

$$
\begin{aligned}
x_1 + x_2 + x_3 + x_5 + x_7 + x_9 &= 0.14 \\
x_5 + x_6 + x_7 + x_9 &= 0.44 \\
x_1 + x_3 + x_4 + x_5 + x_7 + x_8 &= -0.30 \\
x_3 + x_6 + x_8 + x_9 &= 0.14 \\
x_3 + x_5 + x_6 + x_7 + x_8 + x_{10} &= -0.30.
\end{aligned}
$$

This doesn't work. So try letting only x_2 be nonzero, which doesn't work. Then individually try x_3, x_4, etc.

Brute Force For Finding Sparse Solutions

If we can't find a 1-sparse solution, try a 2-sparse solution. Let only x_1 and x_2 be nonzero and try to solve

$$\begin{aligned}
x_1 + x_2 + x_3 + x_5 + x_7 + x_9 &= 0.14\\
x_5 + x_6 + x_7 + x_9 &= 0.44\\
x_1 + x_3 + x_4 + x_5 + x_7 + x_8 &= -0.30\\
x_3 + x_6 + x_8 + x_9 &= 0.14\\
x_3 + x_5 + x_6 + x_7 + x_8 + x_{10} &= -0.30.
\end{aligned}$$

If this doesn't work, try x_1 and x_3, x_1 and x_4, etc.

Brute Force For Finding Sparse Solutions

We might generally try to:

Brute Force For Finding Sparse Solutions

We might generally try to:

- Look for a 1-sparse solution, where only a single x_i is nonzero. Try each of $x_1, x_2, \ldots$

Brute Force For Finding Sparse Solutions

We might generally try to:

- Look for a 1-sparse solution, where only a single x_i is nonzero. Try each of $x_1, x_2, \ldots$
- If that doesn't work, look for 2-sparse solutions, e.g., only x_1 and x_2 nonzero, x_1 and x_3, etc.

Brute Force For Finding Sparse Solutions

We might generally try to:

- Look for a 1-sparse solution, where only a single x_i is nonzero. Try each of $x_1, x_2, \ldots$
- If that doesn't work, look for 2-sparse solutions, e.g., only x_1 and x_2 nonzero, x_1 and x_3, etc.
- If that doesn't work, look for 3-sparse solutions, then 4-sparse, etc.

Brute Force For Finding Sparse Solutions

We might generally try to:

- Look for a 1-sparse solution, where only a single x_i is nonzero. Try each of $x_1, x_2, \ldots$
- If that doesn't work, look for 2-sparse solutions, e.g., only x_1 and x_2 nonzero, x_1 and x_3, etc.
- If that doesn't work, look for 3-sparse solutions, then 4-sparse, etc.
- This is guaranteed to yield the sparsest possible solution.

Brute Force For Finding Sparse Solutions

We might generally try to:

- Look for a 1-sparse solution, where only a single x_i is nonzero. Try each of $x_1, x_2, \ldots$
- If that doesn't work, look for 2-sparse solutions, e.g., only x_1 and x_2 nonzero, x_1 and x_3, etc.
- If that doesn't work, look for 3-sparse solutions, then 4-sparse, etc.
- This is guaranteed to yield the sparsest possible solution.

Unfortunately this completely intractable for larger problems.

Brute Force Example

With 10 marbles, trying all 1-sparse solutions requires us to consider 10 cases.

Brute Force Example

With 10 marbles, trying all 1-sparse solutions requires us to consider 10 cases.

Trying all 2-sparse solutions requires us to consider 45 cases.

Brute Force Example

With 10 marbles, trying all 1-sparse solutions requires us to consider 10 cases.

Trying all 2-sparse solutions requires us to consider 45 cases.

Trying all 3-sparse solutions requires us to consider 120 cases.

Brute Force Example

With 10 marbles, trying all 1-sparse solutions requires us to consider 10 cases.

Trying all 2-sparse solutions requires us to consider 45 cases.

Trying all 3-sparse solutions requires us to consider 120 cases.

With 100 marbles we have to try $100, 4950, 161700, \ldots$ cases. Things quickly get out of hand.

Brute Force Example

With 10 marbles, trying all 1-sparse solutions requires us to consider 10 cases.

Trying all 2-sparse solutions requires us to consider 45 cases.

Trying all 3-sparse solutions requires us to consider 120 cases.

With 100 marbles we have to try $100, 4950, 161700, \ldots$ cases. Things quickly get out of hand.

And the real problems of interest have millions (or more) variables!

Alternate Approaches

We need a computationally feasible method for finding sparse solutions to an underdetermined system of linear equations that may involve many variables.

Alternate Approaches

We need a computationally feasible method for finding sparse solutions to an underdetermined system of linear equations that may involve many variables.

There are several approaches. Two of the most common are

Alternate Approaches

We need a computationally feasible method for finding sparse solutions to an underdetermined system of linear equations that may involve many variables.

There are several approaches. Two of the most common are

- **Basis Pursuit**, which poses the problem in terms of optimization.

Alternate Approaches

We need a computationally feasible method for finding sparse solutions to an underdetermined system of linear equations that may involve many variables.

There are several approaches. Two of the most common are

- **Basis Pursuit**, which poses the problem in terms of optimization.
- **Orthogonal Matching Pursuit**, a greedy algorithm that uses the notion of "correlation" to build up a sparse solution one component at a time.

Traditional Minimum Norm Solutions

One traditional method to nail down a solution (out of infinitely many) to an underdetermined set of linear equations for $x_1, \ldots, x_n$ is to minimize the quantity

$$x_1^2 + x_2^2 + \cdots + x_n^2$$

subject to the constraints that the variables satisfy the linear equations.

Traditional Minimum Norm Solutions

One traditional method to nail down a solution (out of infinitely many) to an underdetermined set of linear equations for $x_1, \ldots, x_n$ is to minimize the quantity

$$x_1^2 + x_2^2 + \cdots + x_n^2$$

subject to the constraints that the variables satisfy the linear equations.

But this does not typically generate sparse solutions.

Example

We want to find a sparse solution to $x_1 + 2x_2 = 8$. There are infinitely many solutions, but the sparsest are the 1-sparse solutions $(x_1 = 8, x_2 = 0)$ and $(x_1 = 0, x_2 = 4)$.

Example

We want to find a sparse solution to $x_1 + 2x_2 = 8$. There are infinitely many solutions, but the sparsest are the 1-sparse solutions $(x_1 = 8, x_2 = 0)$ and $(x_1 = 0, x_2 = 4)$.

Minimizing $x_1^2 + x_2^2$ subject to the constraint $x_1 + 2x_2 = 8$ produces the solution $x_1 = 8/5, x_2 = 16/5$.

Example

We want to find a sparse solution to $x_1 + 2x_2 = 8$. There are infinitely many solutions, but the sparsest are the 1-sparse solutions $(x_1 = 8, x_2 = 0)$ and $(x_1 = 0, x_2 = 4)$.

Minimizing $x_1^2 + x_2^2$ subject to the constraint $x_1 + 2x_2 = 8$ produces the solution $x_1 = 8/5, x_2 = 16/5$.

Both x_1 and x_2 are nonzero; this solution is not the sparsest possible.

Basis Pursuit

In the 1980's (earlier?) it was discovered that minimizing the function

$$|x_1| + |x_2| + |x_3| + \cdots + |x_n|$$

subject to the condition that the variables $x_1, \ldots, x_n$ satisfy a set of linear equations often produces sparse solutions to the set of equations.

Basis Pursuit

In the 1980's (earlier?) it was discovered that minimizing the function

$$|x_1| + |x_2| + |x_3| + \cdots + |x_n|$$

subject to the condition that the variables $x_1, \ldots, x_n$ satisfy a set of linear equations often produces sparse solutions to the set of equations.

But this expression is not differentiable, so this looks hard.

Basis Pursuit

In the 1980's (earlier?) it was discovered that minimizing the function

$$|x_1| + |x_2| + |x_3| + \cdots + |x_n|$$

subject to the condition that the variables $x_1, \ldots, x_n$ satisfy a set of linear equations often produces sparse solutions to the set of equations.

But this expression is not differentiable, so this looks hard.

It turns out that this is a standard and easy-to-solve problem in the field of "linear programming."

Example

We want to find a sparse solution to $x_1 + 2x_2 = 8$. There are infinitely many solutions, but the sparsest are the 1-sparse solutions $(x_1 = 8, x_2 = 0)$ and $(x_1 = 0, x_2 = 4)$.

Example

We want to find a sparse solution to $x_1 + 2x_2 = 8$. There are infinitely many solutions, but the sparsest are the 1-sparse solutions $(x_1 = 8, x_2 = 0)$ and $(x_1 = 0, x_2 = 4)$.

The second solution $x_1 = 0, x_2 = 4$ also happens to be the pair (x_1, x_2) satisfying $x_1 + 2x_2 = 8$ that minimizes $|x_1| + |x_2|$.

Example

We want to find a sparse solution to $x_1 + 2x_2 = 8$. There are infinitely many solutions, but the sparsest are the 1-sparse solutions $(x_1 = 8, x_2 = 0)$ and $(x_1 = 0, x_2 = 4)$.

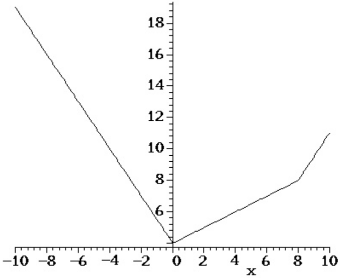
The second solution $x_1 = 0, x_2 = 4$ also happens to be the pair (x_1, x_2) satisfying $x_1 + 2x_2 = 8$ that minimizes $|x_1| + |x_2|$.

You can see this by:

1. Solving $x_1 + 2x_2 = 8$ for $x_2 = 4 - x_1/2$, then
2. Plotting $|x_1| + |x_2| = |x_1| + |4 - x_1/2|$.

Example

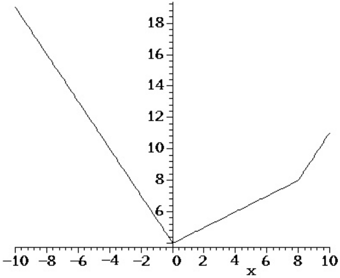
A plot of $|x_1| + |4 - x_1/2|$:



The minimum is at $x_1 = 0$ (corresponds to $x_2 = 4$).

Example

A plot of $|x_1| + |4 - x_1/2|$:



The minimum is at $x_1 = 0$ (corresponds to $x_2 = 4$).

Why does this work?

Distance

The familiar distance formula from the origin to a point (x_1, x_2) in the plane is

$$d_2(x_1, x_2) = \sqrt{x_1^2 + x_2^2}$$

the Euclidean or "ℓ^2 norm".

Distance

The familiar distance formula from the origin to a point (x_1, x_2) in the plane is

$$d_2(x_1, x_2) = \sqrt{x_1^2 + x_2^2}$$

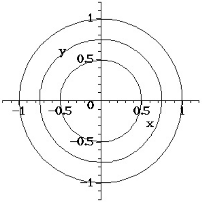
the Euclidean or "ℓ^2 norm".

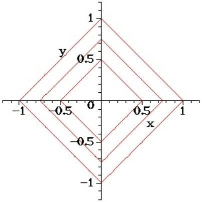
An alternate way to measure distance is the "ℓ^1 norm":

$$d_1(x_1, x_2) = |x_1| + |x_2|.$$

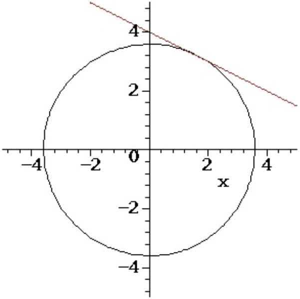
Distance

Sets of the form $d_2(x_1, x_2) = r$ are circles of radius r centered at the origin, while sets of the form $d_1(x_1, x_2) = r$ are "diamonds".



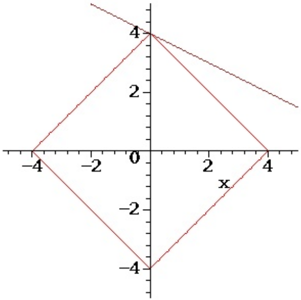


Geometric Intuition



The solution to $x_1 + 2x_2 = 8$ that is closest to the origin in the usual ℓ^2 sense (minimizes $d_2(x_1, x_2)$) is not sparse—both x_1 and x_2 are nonzero.

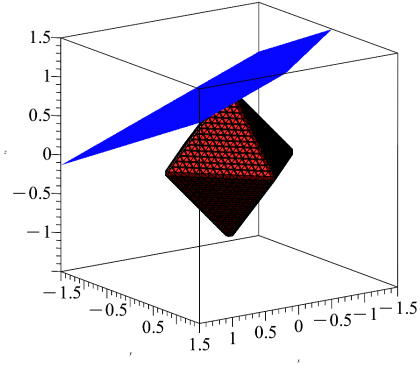
The ℓ^1 Ball



The solution to $x_1 + 2x_2 = 8$ that is closest to the origin in the ℓ^1 sense (minimizes $d_1(x_1, x_2)$) is 1-sparse.

The ℓ^1 Ball

Similar observations holds in three and higher dimensions:



A Small Example

In the present 10-marble case, we need to minimize the quantity

$$f(x_1, \ldots, x_{10}) = |x_1| + |x_2| + \cdots + |x_{10}|$$

subject to the constraints

$$\begin{aligned}
x_1 + x_2 + x_3 + x_5 + x_7 + x_9 &= 0.14 \\
x_5 + x_6 + x_7 + x_9 &= 0.44 \\
x_1 + x_3 + x_4 + x_5 + x_7 + x_8 &= -0.30 \\
x_3 + x_6 + x_8 + x_9 &= 0.14 \\
x_3 + x_5 + x_6 + x_7 + x_8 + x_{10} &= -0.30.
\end{aligned}$$

Back to 100 Marbles

Minimizing $|x_1| + |x_2| + \cdots + |x_{10}|$ subject to the constraint equations works perfectly here and recovers the 2-sparse solution $x_3 = -0.3, x_0 = 0.44$.

Back to 100 Marbles

Minimizing $|x_1| + |x_2| + \cdots + |x_{10}|$ subject to the constraint equations works perfectly here and recovers the 2-sparse solution $x_3 = -0.3, x_0 = 0.44$.

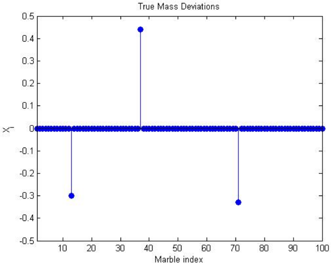
Minimizing $x_1^2 + x_2^2 + \cdots x_{10}^2$ subject to the constraint equations produces a solution (approximate)

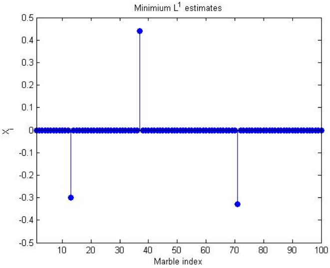
$$x_1 = -0.033, x_2 = -0.008, x_3 = -0.139, x_4 = -0.025, x_5 = 0.015,$$
$$x_6 = 0.120, x_7 = 0.015, x_8 = -0.132, x_9 = 0.291, x_{10} = -0.178.$$

It's not sparse.

Back to 100 Marbles

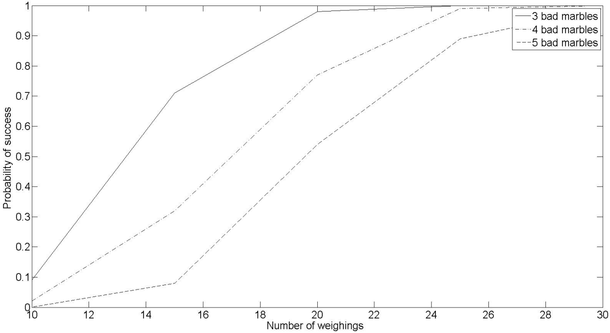
In the 100 marble example with three bad marbles and 20 random subgroups weighed, the result with this approach is also (almost always) exact!





Probability of Success

The more weighings we do (for any fixed number of defective marbles) the better chance of success this approach has.



Quick Summary of Compressed Sensing

We have an underdetermined system of linear system of equations with n unknowns $(x_1, \ldots, x_n)$ and m equations, but $m < n$. However...

Quick Summary of Compressed Sensing

We have an underdetermined system of linear system of equations with n unknowns $(x_1, \ldots, x_n)$ and m equations, but $m < n$. However...

The solution we seek is sparse (most of the variables x_i are zero.)

Quick Summary of Compressed Sensing

We have an underdetermined system of linear system of equations with n unknowns $(x_1, \ldots, x_n)$ and m equations, but $m < n$. However...

The solution we seek is sparse (most of the variables x_i are zero.)

We can often recover a sparse (correct) solution by minimizing the quantity

$$|x_1| + |x_2| + \cdots |x_n|$$

subject to the constraints dictated by the linear equations.

Quick Summary of Compressed Sensing

We have an underdetermined system of linear system of equations with n unknowns $(x_1, \ldots, x_n)$ and m equations, but $m < n$. However...

The solution we seek is sparse (most of the variables x_i are zero.)

We can often recover a sparse (correct) solution by minimizing the quantity

$$|x_1| + |x_2| + \cdots |x_n|$$

subject to the constraints dictated by the linear equations.

Alternatively, we can use "matching pursuit" algorithms (more on this later).

History

History

- That main idea behind compressed sensing—that many problems in applied math/stats/CS involve linear systems of equations but in which the solution is sparse—dates back to seismology work in the 1970s.

History

- That main idea behind compressed sensing—that many problems in applied math/stats/CS involve linear systems of equations but in which the solution is sparse—dates back to seismology work in the 1970s.
- In fact, it's an updated version of "group testing" that dates back to World War 2.

History

- That main idea behind compressed sensing—that many problems in applied math/stats/CS involve linear systems of equations but in which the solution is sparse—dates back to seismology work in the 1970s.
- In fact, it's an updated version of "group testing" that dates back to World War 2.
- Similar ideas appeared in the applied math and statistics communities in the 1980s.

History

- That main idea behind compressed sensing—that many problems in applied math/stats/CS involve linear systems of equations but in which the solution is sparse—dates back to seismology work in the 1970s.
- In fact, it's an updated version of "group testing" that dates back to World War 2.
- Similar ideas appeared in the applied math and statistics communities in the 1980s.
- Several people (Candes, Tao, Donoho) put the subject on a firmer and more insightful mathematical foundation.

History

- That main idea behind compressed sensing—that many problems in applied math/stats/CS involve linear systems of equations but in which the solution is sparse—dates back to seismology work in the 1970s.
- In fact, it's an updated version of "group testing" that dates back to World War 2.
- Similar ideas appeared in the applied math and statistics communities in the 1980s.
- Several people (Candes, Tao, Donoho) put the subject on a firmer and more insightful mathematical foundation.
- CS has a been a hot topic since then.

Future Issues

When will this approach work? In particular,

Future Issues

When will this approach work? In particular,

- Under what conditions will a sparse solution be unique? (Answer: There are a couple different quantitative conditions (the "restricted isometry property", or "incoherence") that the linear system can have that assures a sparse solution is unique.)

Future Issues

When will this approach work? In particular,

- Under what conditions will a sparse solution be unique? (Answer: There are a couple different quantitative conditions (the "restricted isometry property", or "incoherence") that the linear system can have that assures a sparse solution is unique.)
- If a unique sparse solution exists, how do we find it? (Answer: There are a variety of efficient algorithms for finding sparse solutions, or approximately sparse, if there is no truly sparse solution).

Future Issues

When will this approach work? In particular,

- Under what conditions will a sparse solution be unique? (Answer: There are a couple different quantitative conditions (the "restricted isometry property", or "incoherence") that the linear system can have that assures a sparse solution is unique.)
- If a unique sparse solution exists, how do we find it? (Answer: There are a variety of efficient algorithms for finding sparse solutions, or approximately sparse, if there is no truly sparse solution).
- What are some other applications of finding sparse solutions? Examples will include remote sensing of radio sources, machine learning algorithms, signal and image compression, and the "one pixel camera."

Issues

Surprisingly, successful application of the technique usually requires a strong dose of randomness!