# Making Do with Less: An Introduction to Compressed Sensing 3

Kurt Bryan

July 7, 2023

# Imperfect Solutions

Consider a linear system $\mathbf{Ax} = \mathbf{b}$.

# Imperfect Solutions

Consider a linear system $\mathbf{Ax} = \mathbf{b}$.

For a vector $\mathbf{w}$ the quantity $\|\mathbf{Aw} - \mathbf{b}\|$ is called the *residual error* and provides one way to measure the extent to which $\mathbf{w}$ is a solution.

# Imperfect Solutions

Consider a linear system $\mathbf{Ax} = \mathbf{b}$.

For a vector $\mathbf{w}$ the quantity $\|\mathbf{Aw} - \mathbf{b}\|$ is called the *residual error* and provides one way to measure the extent to which $\mathbf{w}$ is a solution.

If $\|\mathbf{Aw} - \mathbf{b}\| = 0$ then $\mathbf{x} = \mathbf{w}$ is a solution to $\mathbf{Ax} = \mathbf{b}$.

# Imperfect Solutions

Consider a linear system $\mathbf{Ax} = \mathbf{b}$.

For a vector $\mathbf{w}$ the quantity $\|\mathbf{Aw} - \mathbf{b}\|$ is called the *residual error* and provides one way to measure the extent to which $\mathbf{w}$ is a solution.

If $\|\mathbf{Aw} - \mathbf{b}\| = 0$ then $\mathbf{x} = \mathbf{w}$ is a solution to $\mathbf{Ax} = \mathbf{b}$.

If we have two approximate solutions to $\mathbf{Ax} = \mathbf{b}$, the solution with smaller residual is generally preferred.

# Example: Finding a 1-Sparse Solution

Consider the linear system $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

# Example: Finding a 1-Sparse Solution

Consider the linear system $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

Conditions for uniqueness of 1-sparse solutions are met ($\mu(\mathbf{A}) \approx 0.95 < 1$, so no two columns are parallel).

# Example: Finding a 1-Sparse Solution

Consider the linear system $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

Conditions for uniqueness of 1-sparse solutions are met ($\mu(\mathbf{A}) \approx 0.95 < 1$, so no two columns are parallel).

We're looking for a single nonzero $x_k$ so that

$$x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ -1 \end{bmatrix} + x_3 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

# Example: Finding a 1-Sparse Solution

Consider the linear system $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

Conditions for uniqueness of 1-sparse solutions are met ($\mu(\mathbf{A}) \approx 0.95 < 1$, so no two columns are parallel).

We're looking for a single nonzero $x_k$ so that

$$x_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ -1 \end{bmatrix} + x_3 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}.$$

The true solution is $\mathbf{x}^* = \langle 0, 0, 3, 0 \rangle$.

# Example: Finding a 1-Sparse Solution

If a 1-sparse solution exists then $\mathbf{b}$ is parallel to some column $\mathbf{a}_k$ of $\mathbf{A}$. Then $\mu(\mathbf{a}_k, \mathbf{b}) = 1$.

## Example: Finding a 1-Sparse Solution

If a 1-sparse solution exists then $\mathbf{b}$ is parallel to some column $\mathbf{a}_k$ of $\mathbf{A}$. Then $\mu(\mathbf{a}_k, \mathbf{b}) = 1$.

Also since no two columns of $\mathbf{A}$ are parallel, $\mu(\mathbf{a}_j, \mathbf{b}) < 1$ for $j \neq k$.

# Example: Finding a 1-Sparse Solution

If a 1-sparse solution exists then $\mathbf{b}$ is parallel to some column $\mathbf{a}_k$ of $\mathbf{A}$. Then $\mu(\mathbf{a}_k, \mathbf{b}) = 1$.

Also since no two columns of $\mathbf{A}$ are parallel, $\mu(\mathbf{a}_j, \mathbf{b}) < 1$ for $j \neq k$.

**Solution Strategy**: Compute $\mu(\mathbf{a}_j, \mathbf{b})$ for each $j = 1, \ldots, n$, choose the $j = k$ so that $\mu(\mathbf{a}_k, \mathbf{b}) = 1$, then find $x_k$ so that $x_k \mathbf{a}_k = \mathbf{b}$.

# Example: Finding a 1-Sparse Solution

With

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}.$$

and $\mathbf{b} = \langle 3, 3 \rangle$ we find coherences $0.707, 0.707, 1.00, 0.949$.

# Example: Finding a 1-Sparse Solution

With

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}.$$

and $\mathbf{b} = \langle 3, 3 \rangle$ we find coherences $0.707, 0.707, 1.00, 0.949$.

Then $\mathbf{b}$ is a multiple of $\mathbf{a}_3$, so solve $x_3 \mathbf{a}_3 = \mathbf{b}$ to find $x_3 = 3$.

# Example: Finding a 1-Sparse Solution

With

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}.$$

and $\mathbf{b} = \langle 3, 3 \rangle$ we find coherences $0.707, 0.707, 1.00, 0.949$.

Then $\mathbf{b}$ is a multiple of $\mathbf{a}_3$, so solve $x_3\mathbf{a}_3 = \mathbf{b}$ to find $x_3 = 3$.

The recovered 1-sparse solution is then

$$\mathbf{x} = 3\mathbf{e}_3 = \langle 0, 0, 3, 0 \rangle$$

where $\mathbf{e}_k$ means the $k$th standard basis vector.

# Example: Finding a 1-Sparse Solution

But suppose the data is a bit off, say $\mathbf{b} = \langle 3.1, 3 \rangle$, and we want the best possible 1-sparse solution.

# Example: Finding a 1-Sparse Solution

But suppose the data is a bit off, say $\mathbf{b} = \langle 3.1, 3 \rangle$, and we want the best possible 1-sparse solution.

Which column of

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}.$$

should we use?

# Example: Finding a 1-Sparse Solution

But suppose the data is a bit off, say $\mathbf{b} = \langle 3.1, 3\rangle$, and we want the best possible 1-sparse solution.

Which column of

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}.$$

should we use?

It seems natural to use the column $\mathbf{a}_3$ (again) since it is "most parallel" to $\mathbf{b}$.

# Example: Finding a 1-Sparse Solution

But suppose the data is a bit off, say $\mathbf{b} = \langle 3.1, 3 \rangle$, and we want the best possible 1-sparse solution.

Which column of

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix}.$$

should we use?

It seems natural to use the column $\mathbf{a}_3$ (again) since it is "most parallel" to $\mathbf{b}$.

We can put this reasoning on a firmer footing.

# Example: Finding a 1-Sparse Solution with Imperfect Data

Suppose no 1-sparse solution to $\mathbf{Ax} = \mathbf{b}$ exists. That is, in

$$x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$$

no $\mathbf{a}_j$ is parallel to $\mathbf{b}$.

# Example: Finding a 1-Sparse Solution with Imperfect Data

Suppose no 1-sparse solution to $\mathbf{Ax} = \mathbf{b}$ exists. That is, in

$$x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$$

no $\mathbf{a}_j$ is parallel to $\mathbf{b}$.

We can still construct a "best" 1-sparse solution.

# Example: Finding a 1-Sparse Solution with Imperfect Data

Suppose no 1-sparse solution to $\mathbf{Ax} = \mathbf{b}$ exists. That is, in

$$x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$$

no $\mathbf{a}_j$ is parallel to $\mathbf{b}$.

We can still construct a "best" 1-sparse solution.

Suppose we decide to use a solution with support $S = \{k\}$. That is, $\mathbf{x} = \langle 0, 0, \ldots, 0, x_k, 0, \ldots, 0\rangle$. This means we need

$$x_k\mathbf{a}_k = \mathbf{b}.$$

## Example: Finding a 1-Sparse Solution with Imperfect Data

Suppose no 1-sparse solution to $\mathbf{Ax} = \mathbf{b}$ exists. That is, in

$$x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$$

no $\mathbf{a}_j$ is parallel to $\mathbf{b}$.

We can still construct a "best" 1-sparse solution.

Suppose we decide to use a solution with support $S = \{k\}$. That is, $\mathbf{x} = \langle 0, 0, \ldots, 0, x_k, 0, \ldots, 0 \rangle$. This means we need

$$x_k\mathbf{a}_k = \mathbf{b}.$$

We will choose $x_k$ to minimize the squared residual $\|x_k\mathbf{a}_k - \mathbf{b}\|^2$.

## Example: Finding a 1-Sparse Solution with Imperfect Data

To minimize $\|x_k\mathbf{a}_k - \mathbf{b}\|^2$ with respect to $x_k$ write

# Example: Finding a 1-Sparse Solution with Imperfect Data

To minimize $\|x_k \mathbf{a}_k - \mathbf{b}\|^2$ with respect to $x_k$ write

$$\begin{aligned}\|x_k \mathbf{a}_k - \mathbf{b}\|^2 &= (x_k \mathbf{a}_k - \mathbf{b}) \cdot (x_k \mathbf{a}_k - \mathbf{b}) \\ &= \|\mathbf{a}_k\|^2 x_k^2 - 2(\mathbf{a}_k \cdot \mathbf{b}) x_k + \|\mathbf{b}\|^2.\end{aligned}$$

## Example: Finding a 1-Sparse Solution with Imperfect Data

To minimize $\|x_k \mathbf{a}_k - \mathbf{b}\|^2$ with respect to $x_k$ write

$$\begin{aligned}\|x_k \mathbf{a}_k - \mathbf{b}\|^2 &= (x_k \mathbf{a}_k - \mathbf{b}) \cdot (x_k \mathbf{a}_k - \mathbf{b}) \\ &= \|\mathbf{a}_k\|^2 x_k^2 - 2(\mathbf{a}_k \cdot \mathbf{b}) x_k + \|\mathbf{b}\|^2.\end{aligned}$$

Minimizing with respect to $x_k$ is pure Calc 1: Differentiate with respect to $x_k$, set the derivative to 0, then solve to find minimizer

$$x_k = \frac{\mathbf{a}_k \cdot \mathbf{b}}{\|\mathbf{a}_k\|^2}.$$

# Example: Finding a 1-Sparse Solution with Imperfect Data

To minimize $\|x_k\mathbf{a}_k - \mathbf{b}\|^2$ with respect to $x_k$ write

$$\begin{aligned}\|x_k\mathbf{a}_k - \mathbf{b}\|^2 &= (x_k\mathbf{a}_k - \mathbf{b}) \cdot (x_k\mathbf{a}_k - \mathbf{b}) \\ &= \|\mathbf{a}_k\|^2 x_k^2 - 2(\mathbf{a}_k \cdot \mathbf{b})x_k + \|\mathbf{b}\|^2.\end{aligned}$$

Minimizing with respect to $x_k$ is pure Calc 1: Differentiate with respect to $x_k$, set the derivative to 0, then solve to find minimizer

$$x_k = \frac{\mathbf{a}_k \cdot \mathbf{b}}{\|\mathbf{a}_k\|^2}.$$

If you substitute this back into $\|x_k\mathbf{a}_k - \mathbf{b}\|^2$ and simplify you find the minimized squared residual can be written as

$$\text{residual} = \|\mathbf{b}\|^2 \left(1 - \left(\frac{\mathbf{a}_k \cdot \mathbf{b}}{\|\mathbf{a}_k\|\|\mathbf{b}\|}\right)^2\right)$$

# Example: Finding a 1-Sparse Solution with Imperfect Data

To minimize $\|x_k \mathbf{a}_k - \mathbf{b}\|^2$ with respect to $x_k$ write

$$\begin{aligned}\|x_k \mathbf{a}_k - \mathbf{b}\|^2 &= (x_k \mathbf{a}_k - \mathbf{b}) \cdot (x_k \mathbf{a}_k - \mathbf{b}) \\ &= \|\mathbf{a}_k\|^2 x_k^2 - 2(\mathbf{a}_k \cdot \mathbf{b}) x_k + \|\mathbf{b}\|^2.\end{aligned}$$

Minimizing with respect to $x_k$ is pure Calc 1: Differentiate with respect to $x_k$, set the derivative to 0, then solve to find minimizer

$$x_k = \frac{\mathbf{a}_k \cdot \mathbf{b}}{\|\mathbf{a}_k\|^2}.$$

If you substitute this back into $\|x_k \mathbf{a}_k - \mathbf{b}\|^2$ and simplify you find the minimized squared residual can be written as

$$\text{residual} = \|\mathbf{b}\|^2 \left(1 - \left(\frac{\mathbf{a}_k \cdot \mathbf{b}}{\|\mathbf{a}_k\| \|\mathbf{b}\|}\right)^2\right) = \|\mathbf{b}\|^2 (1 - \mu^2(\mathbf{a}_k, \mathbf{b})).$$

# Example: Finding a 1-Sparse Solution with Imperfect Data

In summary, if we construct a 1-sparse solution $\mathbf{x}$ to $x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$ with support index $k$, the best choice for $x_k$ is $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$.

# Example: Finding a 1-Sparse Solution with Imperfect Data

In summary, if we construct a 1-sparse solution $\mathbf{x}$ to $x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$ with support index $k$, the best choice for $x_k$ is $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$. The resulting squared residual is

$$\|\mathbf{b}\|^2(1 - \mu^2(\mathbf{a}_k, \mathbf{b})).$$

## Example: Finding a 1-Sparse Solution with Imperfect Data

In summary, if we construct a 1-sparse solution $\mathbf{x}$ to $x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$ with support index $k$, the best choice for $x_k$ is $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$. The resulting squared residual is

$$\|\mathbf{b}\|^2(1 - \mu^2(\mathbf{a}_k, \mathbf{b})).$$

Which column $1 \leq k \leq n$ gives the best 1-sparse solution (smallest residual)? The column that maximizes $\mu(\mathbf{a}_k, \mathbf{b})$.

# Example: Finding a 1-Sparse Solution with Imperfect Data

In summary, if we construct a 1-sparse solution $\mathbf{x}$ to $x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \mathbf{b}$ with support index $k$, the best choice for $x_k$ is $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$. The resulting squared residual is

$$\|\mathbf{b}\|^2(1 - \mu^2(\mathbf{a}_k, \mathbf{b})).$$

Which column $1 \leq k \leq n$ gives the best 1-sparse solution (smallest residual)? The column that maximizes $\mu(\mathbf{a}_k, \mathbf{b})$.

Note that if $\mu(\mathbf{a}_k, \mathbf{b}) = 1$ then the residual is 0 and we obtain a perfect solution, since $\mathbf{b}$ is parallel to $\mathbf{a}_k$.

# Example: Finding a 1-Sparse Solution with Imperfect Data

Consider the system

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3.10 \\ 3.00 \end{bmatrix}.$$

# Example: Finding a 1-Sparse Solution with Imperfect Data

Consider the system

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3.10 \\ 3.00 \end{bmatrix}.$$

First, compute each coherence $\mu(\mathbf{a}_j, \mathbf{b})$ to find the column "most parallel" to $\mathbf{b}$. We find coherences $0.719, 0.695, 0.999, 0.943$ for $1 \leq j \leq 4$.

# Example: Finding a 1-Sparse Solution with Imperfect Data

Consider the system

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3.10 \\ 3.00 \end{bmatrix}.$$

First, compute each coherence $\mu(\mathbf{a}_j, \mathbf{b})$ to find the column "most parallel" to $\mathbf{b}$. We find coherences $0.719, 0.695, 0.999, 0.943$ for $1 \leq j \leq 4$.

The best 1-sparse solution will come from support index 3. Also, from $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$ we find optimal choice $x_3 = 3.05$.

# Summary: Computing the Best 1-sparse solution

To find the best 1-sparse solution $\mathbf{x} = \mathbf{x}^1$ to $\mathbf{A}\mathbf{x} = \mathbf{b}$ we

# Summary: Computing the Best 1-sparse solution

To find the best 1-sparse solution $\mathbf{x} = \mathbf{x}^1$ to $\mathbf{A}\mathbf{x} = \mathbf{b}$ we

1. Find the column index $k$ that maximizes $\mu(\mathbf{a}_k, \mathbf{b})$.

# Summary: Computing the Best 1-sparse solution

To find the best 1-sparse solution $\mathbf{x} = \mathbf{x}^1$ to $\mathbf{A}\mathbf{x} = \mathbf{b}$ we

1. Find the column index $k$ that maximizes $\mu(\mathbf{a}_k, \mathbf{b})$.
2. The solution will have support index $k$, and so $\mathbf{x}^1 = x_k \mathbf{e}_k$ where $\mathbf{e}_k$ is the $k$th standard basis vector.

# Summary: Computing the Best 1-sparse solution

To find the best 1-sparse solution $\mathbf{x} = \mathbf{x}^1$ to $\mathbf{Ax} = \mathbf{b}$ we

1. Find the column index $k$ that maximizes $\mu(\mathbf{a}_k, \mathbf{b})$.
2. The solution will have support index $k$, and so $\mathbf{x}^1 = x_k \mathbf{e}_k$ where $\mathbf{e}_k$ is the $k$th standard basis vector.
3. Choose $x_k$ to minimize $\|x_k \mathbf{a}_k - \mathbf{b}\|^2$; this value is $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$.

# Summary: Computing the Best 1-sparse solution

To find the best 1-sparse solution $\mathbf{x} = \mathbf{x}^1$ to $\mathbf{Ax} = \mathbf{b}$ we

1. Find the column index $k$ that maximizes $\mu(\mathbf{a}_k, \mathbf{b})$.
2. The solution will have support index $k$, and so $\mathbf{x}^1 = x_k \mathbf{e}_k$ where $\mathbf{e}_k$ is the $k$th standard basis vector.
3. Choose $x_k$ to minimize $\|x_k \mathbf{a}_k - \mathbf{b}\|^2$; this value is $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$.

This is a "greedy" algorithm: We choose the index support $k$ to make $\|x_k \mathbf{a}_k - \mathbf{b}\|$ as small as possible.

# Summary: Computing the Best 1-sparse solution

To find the best 1-sparse solution $\mathbf{x} = \mathbf{x}^1$ to $\mathbf{Ax} = \mathbf{b}$ we

1. Find the column index $k$ that maximizes $\mu(\mathbf{a}_k, \mathbf{b})$.
2. The solution will have support index $k$, and so $\mathbf{x}^1 = x_k \mathbf{e}_k$ where $\mathbf{e}_k$ is the $k$th standard basis vector.
3. Choose $x_k$ to minimize $\|x_k \mathbf{a}_k - \mathbf{b}\|^2$; this value is $x_k = (\mathbf{a}_k \cdot \mathbf{b})/\|\mathbf{a}_k\|^2$.

This is a "greedy" algorithm: We choose the index support $k$ to make $\|x_k \mathbf{a}_k - \mathbf{b}\|$ as small as possible.

We can extend this greedy approach to iteratively construct 2-, 3-, 4-, or higher-sparsity solutions.

# Example: Finding a 2-Sparse Solution

Consider finding a 2-sparse solution to $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 & -1 & 3 & 0 \\ 3 & 1 & -1 & 1 & 1 & 3 \\ 0 & 1 & -1 & 2 & 4 & -2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -3 \\ 5 \\ 2 \end{bmatrix}.$$

# Example: Finding a 2-Sparse Solution

Consider finding a 2-sparse solution to $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 & -1 & 3 & 0 \\ 3 & 1 & -1 & 1 & 1 & 3 \\ 0 & 1 & -1 & 2 & 4 & -2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -3 \\ 5 \\ 2 \end{bmatrix}.$$

The sparsest solution is $\mathbf{x} = \langle 1, 0, -2, 0, 0, 0 \rangle$, support set $S = \{1, 3\}$ (but we don't know this).

# Example: Finding a 2-Sparse Solution

Consider finding a 2-sparse solution to $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 & -1 & 3 & 0 \\ 3 & 1 & -1 & 1 & 1 & 3 \\ 0 & 1 & -1 & 2 & 4 & -2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -3 \\ 5 \\ 2 \end{bmatrix}.$$

The sparsest solution is $\mathbf{x} = \langle 1, 0, -2, 0, 0, 0 \rangle$, support set $S = \{1, 3\}$ (but we don't know this).

If we knew even the support of the 2-sparse solution things would be easy—but we don't.

# Example: Finding a 2-Sparse Solution

Start by finding the best 1-sparse solution. The coherence of each column of $\mathbf{A}$ with $\mathbf{b}$ is

$$0.6156, 0.3746, 0.8609, 0.7947, 0.1273, 0.4949.$$

# Example: Finding a 2-Sparse Solution

Start by finding the best 1-sparse solution. The coherence of each column of **A** with **b** is

$$0.6156, 0.3746, 0.8609, 0.7947, 0.1273, 0.4949.$$

Since column 3 is most highly correlated with **b**, we will take a 1-sparse solution of the form $\mathbf{x}^1 = x_3\mathbf{e}_3$.

# Example: Finding a 2-Sparse Solution

Start by finding the best 1-sparse solution. The coherence of each column of $\mathbf{A}$ with $\mathbf{b}$ is

$$0.6156, 0.3746, 0.8609, 0.7947, 0.1273, 0.4949.$$

Since column 3 is most highly correlated with $\mathbf{b}$, we will take a 1-sparse solution of the form $\mathbf{x}^1 = x_3\mathbf{e}_3$.

Choosing $x_3$ to minimize $\|x_3\mathbf{a}_3 - \mathbf{b}\|^2$ yields $x_3 = -13/6$. The best 1-sparse solution is $\mathbf{x}^1 = \langle 0, 0, -13/6, 0, 0, 0\rangle$ with support set $S = \{3\}$.

# Example: Finding a 2-Sparse Solution

Start by finding the best 1-sparse solution. The coherence of each column of $\mathbf{A}$ with $\mathbf{b}$ is

$$0.6156, 0.3746, 0.8609, 0.7947, 0.1273, 0.4949.$$

Since column 3 is most highly correlated with $\mathbf{b}$, we will take a 1-sparse solution of the form $\mathbf{x}^1 = x_3\mathbf{e}_3$.

Choosing $x_3$ to minimize $\|x_3\mathbf{a}_3 - \mathbf{b}\|^2$ yields $x_3 = -13/6$. The best 1-sparse solution is $\mathbf{x}^1 = \langle 0, 0, -13/6, 0, 0, 0\rangle$ with support set $S = \{3\}$.

We'll now boost $\mathbf{x}^1$ to a 2-sparse solution $\mathbf{x}^2$.

# Example: Finding a 2-Sparse Solution

Try $\mathbf{x}^2 = \mathbf{x}^1 + x_j \mathbf{e}_j$ ($j$ and $x_j$ to be determined).

# Example: Finding a 2-Sparse Solution

Try $\mathbf{x}^2 = \mathbf{x}^1 + x_j \mathbf{e}_j$ ($j$ and $x_j$ to be determined). Note that

$$\mathbf{A}\mathbf{x}^2 = \mathbf{A}\mathbf{x}^1 + x_j \mathbf{A}\mathbf{e}_j = \mathbf{A}\mathbf{x}^1 + x_j \mathbf{a}_j.$$

# Example: Finding a 2-Sparse Solution

Try $\mathbf{x}^2 = \mathbf{x}^1 + x_j \mathbf{e}_j$ ($j$ and $x_j$ to be determined). Note that

$$\mathbf{A}\mathbf{x}^2 = \mathbf{A}\mathbf{x}^1 + x_j \mathbf{A}\mathbf{e}_j = \mathbf{A}\mathbf{x}^1 + x_j \mathbf{a}_j.$$

If we want $\mathbf{A}\mathbf{x}^2 = \mathbf{b}$ then we need $\mathbf{A}\mathbf{x}^1 + x_j \mathbf{a}_j = \mathbf{b}$ or

$$x_j \mathbf{a}_j = \mathbf{b} - \mathbf{A}\mathbf{x}^1.$$

# Example: Finding a 2-Sparse Solution

Try $\mathbf{x}^2 = \mathbf{x}^1 + x_j \mathbf{e}_j$ ($j$ and $x_j$ to be determined). Note that

$$\mathbf{A}\mathbf{x}^2 = \mathbf{A}\mathbf{x}^1 + x_j \mathbf{A}\mathbf{e}_j = \mathbf{A}\mathbf{x}^1 + x_j \mathbf{a}_j.$$

If we want $\mathbf{A}\mathbf{x}^2 = \mathbf{b}$ then we need $\mathbf{A}\mathbf{x}^1 + x_j \mathbf{a}_j = \mathbf{b}$ or

$$x_j \mathbf{a}_j = \mathbf{b} - \mathbf{A}\mathbf{x}^1.$$

This is just the problem of finding the best 1-sparse solution to $\mathbf{A}\mathbf{x} = \mathbf{b} - \mathbf{A}\mathbf{x}^1$ (instead of $\mathbf{A}\mathbf{x} = \mathbf{b}$), which we know how to do!

# Example: Finding a 2-Sparse Solution

So we seek a 1-sparse solution to $\mathbf{Ax} = \mathbf{b} - \mathbf{Ax}^1$. Note that

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 & -1 & 3 & 0 \\ 3 & 1 & -1 & 1 & 1 & 3 \\ 0 & 1 & -1 & 2 & 4 & -2 \end{bmatrix}, \quad \mathbf{b} - \mathbf{Ax}^1 = \begin{bmatrix} 4/3 \\ 17/6 \\ -1/6 \end{bmatrix}.$$

# Example: Finding a 2-Sparse Solution

So we seek a 1-sparse solution to $\mathbf{Ax} = \mathbf{b} - \mathbf{Ax}^1$. Note that

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 2 & -1 & 3 & 0 \\ 3 & 1 & -1 & 1 & 1 & 3 \\ 0 & 1 & -1 & 2 & 4 & -2 \end{bmatrix}, \quad \mathbf{b} - \mathbf{Ax}^1 = \begin{bmatrix} 4/3 \\ 17/6 \\ -1/6 \end{bmatrix}.$$

These coherence of $\mathbf{b} - \mathbf{Ax}^1$ with each column of $\mathbf{A}$ is

$$0.9916, 0.7365, 0.0000, 0.1519, 0.3857, 0.7813$$

so we'll take $k = 1$ and take $\mathbf{x}^2 = x_1\mathbf{e}_1 + x_3\mathbf{e}_3$ where we already have $x_3 = -13/6$.

# Example: Finding a 2-Sparse Solution

Choose $x_1$ to minimize $\|\mathbf{Ax}^2 - \mathbf{b}\|^2$ or equivalently, $\|x_1\mathbf{a}_1 - (\mathbf{b} - x_3\mathbf{a}_3)\|^2$, which leads to $x_1 = 59/60$.

# Example: Finding a 2-Sparse Solution

Choose $x_1$ to minimize $\|\mathbf{A}\mathbf{x}^2 - \mathbf{b}\|^2$ or equivalently, $\|x_1\mathbf{a}_1 - (\mathbf{b} - x_3\mathbf{a}_3)\|^2$, which leads to $x_1 = 59/60$.

The best 2-sparse solution (by this algorithm) is estimated to be

$$\mathbf{x}^2 = \langle 59/60, 0, -13/6, 0, 0, 0 \rangle.$$

# Example: Finding a 2-Sparse Solution

Choose $x_1$ to minimize $\|\mathbf{A}\mathbf{x}^2 - \mathbf{b}\|^2$ or equivalently, $\|x_1\mathbf{a}_1 - (\mathbf{b} - x_3\mathbf{a}_3)\|^2$, which leads to $x_1 = 59/60$.

The best 2-sparse solution (by this algorithm) is estimated to be

$$\mathbf{x}^2 = \langle 59/60, 0, -13/6, 0, 0, 0\rangle.$$

Compare to the correct sparsest solution

$$\mathbf{x}^* = \langle 1, 0, -2, 0, 0, 0\rangle.$$

# Example: Finding a $p$-Sparse Solution

This process can be continued and yields an algorithm called *matching pursuit*.

# Example: Finding a $p$-Sparse Solution

This process can be continued and yields an algorithm called *matching pursuit*.

1. Construct a 1-sparse solution $\mathbf{x}^1$ using the above procedure.

# Example: Finding a $p$-Sparse Solution

This process can be continued and yields an algorithm called *matching pursuit*.

1. Construct a 1-sparse solution $\mathbf{x}^1$ using the above procedure.
2. Let $\mathbf{x}^2 = \mathbf{x}^1 + x_j\mathbf{e}_j$; we'd like $A\mathbf{x}^2 = \mathbf{b}$, or

$$x_j\mathbf{a}_j = \mathbf{b} - \mathbf{A}\mathbf{x}^1.$$

This is just the problem of finding a 1-sparse solution to $\mathbf{A}\mathbf{x} = \mathbf{b} - \mathbf{A}\mathbf{x}^1$, so proceed as before.

# Example: Finding a $p$-Sparse Solution

This process can be continued and yields an algorithm called *matching pursuit*.

1. Construct a 1-sparse solution $\mathbf{x}^1$ using the above procedure.
2. Let $\mathbf{x}^2 = \mathbf{x}^1 + x_j \mathbf{e}_j$; we'd like $A\mathbf{x}^2 = \mathbf{b}$, or

$$x_j \mathbf{a}_j = \mathbf{b} - \mathbf{A}\mathbf{x}^1.$$

   This is just the problem of finding a 1-sparse solution to $\mathbf{A}\mathbf{x} = \mathbf{b} - \mathbf{A}\mathbf{x}^1$, so proceed as before.
3. Let $\mathbf{x}^3 = \mathbf{x}^2 + x_j \mathbf{e}_j$; we'd like $A\mathbf{x}^2 = \mathbf{b}$, or

$$x_j \mathbf{a}_j = \mathbf{b} - \mathbf{A}\mathbf{x}^2.$$

   This is just the problem of finding a 1-sparse solution to $\mathbf{A}\mathbf{x} = \mathbf{b} - \mathbf{A}\mathbf{x}^2$, so proceed as before.

# Observation

In the 2-sparse example the true solution was

$$\mathbf{x}^* = \langle 1, 0, -2, 0, 0, 0 \rangle$$

with support $S = \{1, 3\}$.

# Observation

In the 2-sparse example the true solution was

$$\mathbf{x}^* = \langle 1, 0, -2, 0, 0, 0 \rangle$$

with support $S = \{1, 3\}$.

The recovered solution was

$$\mathbf{x}^2 = \langle 59/60, 0, -13/6, 0, 0, 0 \rangle,$$

also with support $S = \{1, 3\}$.

# Observation

In the 2-sparse example the true solution was

$$\mathbf{x}^* = \langle 1, 0, -2, 0, 0, 0 \rangle$$

with support $S = \{1, 3\}$.

The recovered solution was

$$\mathbf{x}^2 = \langle 59/60, 0, -13/6, 0, 0, 0 \rangle,$$

also with support $S = \{1, 3\}$.

Note that $\mathbf{x}^2$ has the correct support, but the component entries are off.

# Improving Matching Pursuit

An improvement is this: we've identified a 2-sparse solution with support $S = \{1, 3\}$ so we think the best 2-sparse solution is of the form

$$\mathbf{x}^2 = x_1\mathbf{e}_1 + x_3\mathbf{e}_3.$$

# Improving Matching Pursuit

An improvement is this: we've identified a 2-sparse solution with support $S = \{1, 3\}$ so we think the best 2-sparse solution is of the form

$$\mathbf{x}^2 = x_1\mathbf{e}_1 + x_3\mathbf{e}_3.$$

We don't settle for the $x_1$ and $x_3$ values obtained from matching pursuit. Instead, we go back and "re-choose" $x_1$ and $x_3$ simultaneously to minimize the residual quantity $\|\mathbf{A}\mathbf{x}^2 - \mathbf{b}\|^2$.

# Improving Matching Pursuit

An improvement is this: we've identified a 2-sparse solution with support $S = \{1, 3\}$ so we think the best 2-sparse solution is of the form

$$\mathbf{x}^2 = x_1\mathbf{e}_1 + x_3\mathbf{e}_3.$$

We don't settle for the $x_1$ and $x_3$ values obtained from matching pursuit. Instead, we go back and "re-choose" $x_1$ and $x_3$ simultaneously to minimize the residual quantity $\|\mathbf{A}\mathbf{x}^2 - \mathbf{b}\|^2$.

This leads to minimizing $\|x_1\mathbf{a}_1 + x_3\mathbf{a}_3 - \mathbf{b}\|^2$, an easy quadratic minimization in variables $x_1$ and $x_3$ (it also yields the correct 2-sparse solution here).

# Improving Matching Pursuit

An improvement is this: we've identified a 2-sparse solution with support $S = \{1, 3\}$ so we think the best 2-sparse solution is of the form

$$\mathbf{x}^2 = x_1\mathbf{e}_1 + x_3\mathbf{e}_3.$$

We don't settle for the $x_1$ and $x_3$ values obtained from matching pursuit. Instead, we go back and "re-choose" $x_1$ and $x_3$ simultaneously to minimize the residual quantity $\|\mathbf{A}\mathbf{x}^2 - \mathbf{b}\|^2$.

This leads to minimizing $\|x_1\mathbf{a}_1 + x_3\mathbf{a}_3 - \mathbf{b}\|^2$, an easy quadratic minimization in variables $x_1$ and $x_3$ (it also yields the correct 2-sparse solution here).

This modification yields *Orthogonal Matching Pursuit* (OMP), a standard computational algorithm in CS.

# Convergence of OMP

**Theorem**

*Suppose* $\mu(\mathbf{A}) < 1/(2k-1)$ *and let* $\mathbf{x} = \mathbf{x}^*$ *be the (unique)* $k$*-sparse solution to* $\mathbf{A}\mathbf{x} = \mathbf{b}$. *Then OMP will recover* $\mathbf{x}^*$ *exactly, and will recover the components of* $\mathbf{x}^*$ *in descending order of magnitude.*

# Basis Pursuit

In the first lecture we saw some motivation that made it plausible that finding a sparse solution to an underdetermined linear system $\mathbf{Ax} = \mathbf{b}$ (which probably has many solutions) might be accomplished by solving the minimization problem

$$\min_{\mathbf{x}} \ \|\mathbf{x}\|_1$$

where $\|\mathbf{x}\|_1 = \sum_{j=1}^{n} |x_j|$, subject to the constraint $\mathbf{Ax} = \mathbf{b}$.

# Basis Pursuit

In the first lecture we saw some motivation that made it plausible that finding a sparse solution to an underdetermined linear system $\mathbf{Ax} = \mathbf{b}$ (which probably has many solutions) might be accomplished by solving the minimization problem

$$\min_{\mathbf{x}} \ \|\mathbf{x}\|_1$$

where $\|\mathbf{x}\|_1 = \sum_{j=1}^{n} |x_j|$, subject to the constraint $\mathbf{Ax} = \mathbf{b}$.

This would be a Calculus 3 problem if not for the fact that the objective function is not differentiable.

# Linear Programming

One of the great accomplishments of 20th century mathematics was the development of techniques for solving optimization problems of the form

$$\min_{\mathbf{y}} \sum_{j=1}^{n} c_j y_j$$

(this function is linear) subject to equality constraints $\mathbf{A}\mathbf{y} = \mathbf{b}$ and inequality constraints $\mathbf{B}\mathbf{y} \leq \mathbf{d}$ (inequality taken component-by-component).

# Linear Programming

One of the great accomplishments of 20th century mathematics was the development of techniques for solving optimization problems of the form

$$\min_{\mathbf{y}} \sum_{j=1}^{n} c_j y_j$$

(this function is linear) subject to equality constraints $\mathbf{A}\mathbf{y} = \mathbf{b}$ and inequality constraints $\mathbf{B}\mathbf{y} \leq \mathbf{d}$ (inequality taken component-by-component).

Two main classes of methods are the *simplex method* and *interior point methods*.

# Conversion to Linear Programming

The problem of minimizing $\|\mathbf{x}\|_1$ subject to constraint $\mathbf{Ax} = \mathbf{b}$ can be cast as a linear programming problems as follows:

# Conversion to Linear Programming

The problem of minimizing $\|\mathbf{x}\|_1$ subject to constraint $\mathbf{Ax} = \mathbf{b}$ can be cast as a linear programming problems as follows:

1. Introduce new variables $x_{n+1}, \ldots, x_{2n}$ (doubling the size of $\mathbf{x}$).

# Conversion to Linear Programming

The problem of minimizing $\|\mathbf{x}\|_1$ subject to constraint $\mathbf{Ax} = \mathbf{b}$ can be cast as a linear programming problems as follows:

1. Introduce new variables $x_{n+1}, \ldots, x_{2n}$ (doubling the size of $\mathbf{x}$).
2. Add inequality constraints $x_j \leq x_{j+n}$ and $-x_j \leq x_{j+n}$ for $1 \leq j \leq n$, which are equivalent to $-x_{j+n} \leq x_j \leq x_{j+n}$ or $|x_j| \leq x_{j+n}$.

# Conversion to Linear Programming

The problem of minimizing $\|\mathbf{x}\|_1$ subject to constraint $\mathbf{Ax} = \mathbf{b}$ can be cast as a linear programming problems as follows:

1. Introduce new variables $x_{n+1}, \ldots, x_{2n}$ (doubling the size of $\mathbf{x}$).
2. Add inequality constraints $x_j \leq x_{j+n}$ and $-x_j \leq x_{j+n}$ for $1 \leq j \leq n$, which are equivalent to $-x_{j+n} \leq x_j \leq x_{j+n}$ or $|x_j| \leq x_{j+n}$.
3. We replace minimizing $\|\mathbf{x}\|_1$ with the problem of minimizing $x_{n+1} + \cdots + x_{2n}$ subject to $\mathbf{Ax} = \mathbf{b}$ and the new inequality constraints in step 2 above.

# Conversion to Linear Programming

The problem of minimizing $\|\mathbf{x}\|_1$ subject to constraint $\mathbf{Ax} = \mathbf{b}$ can be cast as a linear programming problems as follows:

1. Introduce new variables $x_{n+1}, \ldots, x_{2n}$ (doubling the size of $\mathbf{x}$).
2. Add inequality constraints $x_j \leq x_{j+n}$ and $-x_j \leq x_{j+n}$ for $1 \leq j \leq n$, which are equivalent to $-x_{j+n} \leq x_j \leq x_{j+n}$ or $|x_j| \leq x_{j+n}$.
3. We replace minimizing $\|\mathbf{x}\|_1$ with the problem of minimizing $x_{n+1} + \cdots + x_{2n}$ subject to $\mathbf{Ax} = \mathbf{b}$ and the new inequality constraints in step 2 above.
4. Then $x_1, \ldots, x_n$ provide the solution to the original problem.

# Small Example

Consider minimizing $|x_1| + |x_2|$ subject to $x_1 + 2x_2 = 6$. We

## Small Example

Consider minimizing $|x_1| + |x_2|$ subject to $x_1 + 2x_2 = 6$. We

1. Introduce new variables $x_3, x_4$.

# Small Example

Consider minimizing $|x_1| + |x_2|$ subject to $x_1 + 2x_2 = 6$. We

1. Introduce new variables $x_3, x_4$.
2. Add inequality constraints $x_1 \leq x_3$ and $-x_1 \leq x_3$ (equivalent to $|x_1| \leq x_3$ , and $x_2 \leq x_4$ and $-x_2 \leq x_4$ (equivalent to $|x_2| \leq x_4$).

## Small Example

Consider minimizing $|x_1| + |x_2|$ subject to $x_1 + 2x_2 = 6$. We

1. Introduce new variables $x_3, x_4$.
2. Add inequality constraints $x_1 \leq x_3$ and $-x_1 \leq x_3$ (equivalent to $|x_1| \leq x_3$ , and $x_2 \leq x_4$ and $-x_2 \leq x_4$ (equivalent to $|x_2| \leq x_4$).
3. The new problem is to minimize $x_3 + x_4$ subject to the inequalities $x_1 \leq x_3$, $-x_1 \leq x_3$, $x_2 \leq x_4$, and $-x_2 \leq x_4$, and the equality constraint $x_1 + 2x_2 = 6$.

# Convergence of BP

Seeking a sparse solution to $\mathbf{Ax} = \mathbf{b}$ by minimizing $\|\mathbf{x}\|_1$ is called *basis pursuit* (BP).

**Theorem**

*Suppose* $\mu(\mathbf{A}) < 1/(2k-1)$ *and let* $\mathbf{x}^* = \mathbf{x}^*$ *be the (unique) k-sparse solution to* $\mathbf{Ax} = \mathbf{b}$. *Then BP will recover* $\mathbf{x}^*$ *exactly.*

# Other Algorithms and Issues

Another class of algorithms for finding sparse solutions has also been developed, *iterative hard thresholding*.

# Other Algorithms and Issues

Another class of algorithms for finding sparse solutions has also been developed, *iterative hard thresholding*.

If the vector $\mathbf{b}$ is noisy then enforcing $\mathbf{Ax} = \mathbf{b}$ may be too restrictive (you're forcing a fit to noise). In this case we might minimize $\|\mathbf{x}\|_1$ subject to the constraint $\|\mathbf{Ax} - \mathbf{b}\|_2 \leq \epsilon$ for some $\epsilon$. This is called *basis pursuit denoising*.

# Other Algorithms and Issues

Another class of algorithms for finding sparse solutions has also been developed, *iterative hard thresholding*.

If the vector $\mathbf{b}$ is noisy then enforcing $\mathbf{Ax} = \mathbf{b}$ may be too restrictive (you're forcing a fit to noise). In this case we might minimize $\|\mathbf{x}\|_1$ subject to the constraint $\|\mathbf{Ax} - \mathbf{b}\|_2 \leq \epsilon$ for some $\epsilon$. This is called *basis pursuit denoising*.

In OMP we would iterate only until the residual $\|\mathbf{Ax}^p - \mathbf{b}\|$ is comparable to the noise level of $\mathbf{b}$.