

# „Magische“ Rekonstruktion: Compressed Sensing

Von Cleve Moler

Als ich zum ersten Mal von Compressed Sensing hörte, war ich skeptisch. Es gab Behauptungen, dass es die für die Darstellung von Signalen und Bildern erforderliche Datenmenge um enorme Größenordnungen verringert und anschließend die Originale exakt wiederherstellt. Aus dem Nyquist-Shannon-Abtasttheorem wusste ich, dass das unmöglich ist. Nachdem ich aber mehr über Compressed Sensing erfahren hatte, erkannte ich, dass sowohl die Behauptungen als auch das Theorem völlig korrekt sind, sofern bestimmte Bedingungen gelten.

**D**as Abtasttheorem nach Nyquist und Shannon besagt, dass ein Signal zu seiner exakten und eindeutigen Wiederherstellung mit mindestens der doppelten Frequenz abgetastet werden muss. Natürlich gilt dieses Theorem auch weiterhin; wenn Sie ein Byte in einem aus weißem Rauschen bestehenden Signal oder Bild auslassen, können Sie das Original nicht wiederherstellen. Die meisten interessanten Signale und Bilder bestehen aber nicht aus weißem Rauschen. Stellt man sie durch geeignete Basisfunktionen wie etwa trigonometrische Funktionen oder Wavelets dar, dann haben viele Signale nur relativ wenige von Null verschiedene Koeffizienten. In der Terminologie des Compressed (oder Compressive) Sensing sind sie *dünn besetzt*.

## Das zugrunde liegende Matrizen-Problem

Natürlich war der Aspekt des Compressed Sensing, der zuerst meine Aufmerksamkeit erregt hat, die zugrunde liegende Matrizen-Mathematik. Ein Rohsignal oder -bild kann man als einen Vektor  $f$  mit Millionen von Komponenten auffassen. Wir nehmen an, dass  $f$  als Linearkombination bestimmter Basisfunktionen aufgefasst werden kann:

$$f = \Psi c$$

Diese Basisfunktionen müssen für eine konkrete Anwendung geeignet sein. In unserem Beispiel ist  $\Psi$  die diskrete Cosinus-Transformation. Wir nehmen außerdem

an, dass die meisten Koeffizienten  $c$  praktisch gleich Null sind und  $c$  damit dünn besetzt ist. Zur Abtastung des Signals benötigen wir noch einen anderen linearen Operator,

$$b = \Phi f$$

In unserem Beispiel besteht  $b$  aus einigen zufälligen Abtastwerten von  $f$ , wodurch  $\Phi$  eine Teilmenge der Zeilen des Identitäts-Operators darstellt. Es sind aber auch kompliziertere Abtastoperatoren möglich.

Zur Rekonstruktion des Signals müssen wir versuchen, die Koeffizienten wiederherzustellen, indem wir die Gleichung lösen:

$$Ax = b, \text{ mit } A = \Phi\Psi$$

Haben wir erst die Koeffizienten, dann können wir das Signal selbst rekonstruieren durch Berechnung von:

$$f \approx \Psi x$$

Da dies eine Kompression ist, ist  $A$  rechteckig mit viel mehr Spalten als Zeilen. Zur Berechnung der Koeffizienten  $x$  muss ein unterbestimmtes lineares Gleichungssystem,  $Ax = b$ , gelöst werden. In dieser Situation gibt es viel mehr Unbekannte als Gleichungen. Der Schlüssel des fast schon magischen Rekonstruktionsprozesses ist die nichtlineare Regularisierung mit der  $l_1$ -Norm.

Eine sehr kleine Version dieser Situation habe ich 1990 in einem Cleve's Corner unter dem Titel "The World's Simplest Impossible Problem" beschrieben. Ich denke mir zwei Zahlen, deren Mittelwert gleich 3 ist. Wie lauten die Zahlen? Nachdem Sie sich beschwert haben, dass ich Ihnen nicht genug Informationen gegeben habe, antworten Sie vielleicht 2 und 4. Falls Sie das tun, dann wenden Sie unbewusst eine Art der Regularisierung an, die voraussetzt, dass das Ergebnis aus zwei konkreten Ganzzahlen besteht.

MATLAB® kann zwei verschiedene Antworten geben. In algebraischer Hinsicht ist das Problem ein 1-mal-2 lineares Gleichungssystem mit der Matrix

$$A = [1/2 \ 1/2]$$

und der rechten Seite

$$b = 3$$

Wir wollen einen aus zwei Elementen bestehenden Vektor  $y$  finden, der die Gleichung  $Ay = b$  löst. Die Kleinste-Quadrate-Lösung wird durch die Pseudoinverse berechnet:

$$\begin{aligned} y &= \text{pinv}(A) * b \\ y &= \\ & 3 \\ & 3 \end{aligned}$$

Aber der Backslash-Operator erzeugt eine andere Lösung:

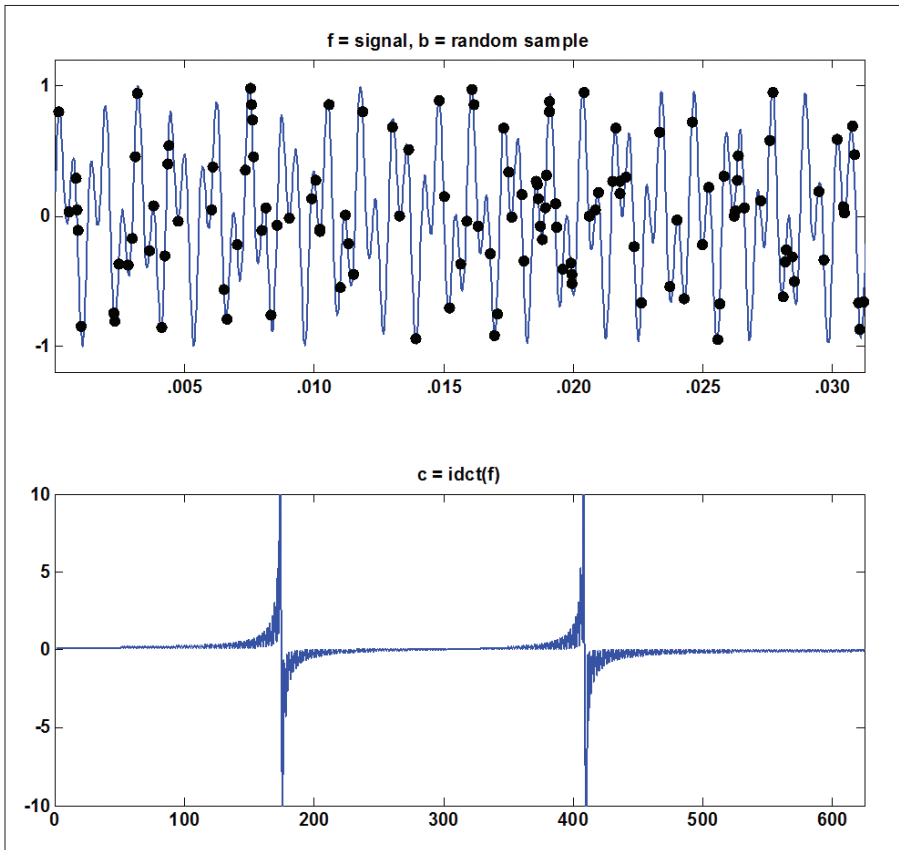


ABB. 1. Oben: Zufällige Stichproben des Ursprungssignals, das die „A“-Taste auf einem Tonwahltelefon erzeugt. Unten: Die inverse Diskrete Cosinus-Transformation des Signals.

```
x = A\b
x =
     6
     0
```

Beide Lösungen sind gültig, werden aber trotzdem von menschlichen Rätsellösern nur selten angeführt. Beachten Sie, dass die vom Backslash berechnete Lösung dünn besetzt ist; eine ihrer Komponenten ist gleich Null.

### Eine größere Variante derselben Aufgabe

Die Wiederherstellung von Signalen oder Bildern ist eine umfangreichere Anwendung derselben Aufgabe. Vorgegeben sind tausende gewichtete Mittelwerte von mehreren Millionen Signal- oder Bildwerten. Unsere

Aufgabe ist, das ursprüngliche Signal oder Bild zu rekonstruieren. Wir haben die komprimierte Probe  $b$  und müssen  $Ax = b$  lösen. Die Anzahl möglicher Lösungen ist gewaltig. Um die richtige herauspicken zu können, benötigt man Vektornormen. Die bekannte euklidische Abstandsnorm,  $l_2$ , lautet

$$l_2: \text{norm}(x,2) = \text{sqrt}(\text{sum}(x.^2))$$

Die „Manhattan“-Norm,  $l_1$ , ist nach der Reisezeit entlang eines von Straßen gebildeten quadratischen Gitters benannt.

$$l_1: \text{norm}(x,1) = \text{sum}(\text{abs}(x))$$

Die Notation  $l_0$  wird für eine Funktion verwendet, die nur die Zahl der von Null

verschiedenen Komponenten zählt. Sie ist eigentlich keine Norm.

$$l_0: \text{norm}_0(x) = \text{sum}(x \neq 0)$$

Diese Normen erlegen unserem unterbestimmten linearen Gleichungssystem drei verschiedene nichtlineare Regularisierungen oder Beschränkungen auf.

Finde unter allen  $x$ , die  $Ax = b$  erfüllen, ein  $x$ , das  $l_p(x)$  minimiert.

Für unser 20 Jahre altes Beispiel ist  $l_2(y)$  kleiner als  $l_2(x)$ , und  $l_1(y)$  und  $l_1(x)$  sind zufällig gleich.  $l_0(y)$  ist aber größer als  $l_0(x)$ .

### Dünne Besetzung und die $l_1$ -Norm

Die Schlüssel zum Compressed Sensing sind dünne Besetzung und die  $l_1$ -Norm. Wenn die Darstellung des Ursprungssignals oder -bildes als Linearkombination der gewählten Basisfunktionen viele Koeffizienten enthält, die gleich Null sind, dann kann man das Signal oft exakt rekonstruieren. Im Prinzip würde man zur Berechnung dieser Rekonstruktion die von Null verschiedenen Koeffizienten mit  $l_0$  zählen. Die rechnerische Komplexität dieses kombinatorischen Problems lässt das aber in der Praxis gar nicht zu (es ist NP-schwer). Glücklicherweise kann, wie David Donoho, Emmanuel Candès und Terence Tao gezeigt haben,  $l_0$  durch  $l_1$  ersetzt werden. Sie erklären, dass die beiden Probleme „mit überwältigender Wahrscheinlichkeit“ dieselbe Lösung haben. Die Berechnung von  $l_1$  ist durchführbar, weil sie als lineares Programmierungsproblem formuliert und mit dem gebräuchlichen Simplex-Algorithmus oder modernen Innere-Punkt-Methoden gelöst werden kann.

### Ein Beispiel

Die Basisfunktion für unser Beispiel ist die Diskrete Cosinus-Transformation (DCT). Das von der Taste „A“ auf einem Tonwahl-Telefon

erzeugte Signal ist die Summe zweier sinusförmiger Töne mit unvereinbaren Frequenzen,

$$f(t) = \sin(1394\pi t) + \sin(3266\pi t)$$

Tasten wir diesen Ton für eine Achtelsekunde mit 40000 Hz ab, resultiert ein Vektor der Länge  $n = 5000$ . Das obere Diagramm in Abbildung 1 zeigt einen Ausschnitt dieses Signals sowie einige der  $m = 500$  zufällig herausgegriffenen Abtastwerte. Das untere Diagramm zeigt die Koeffizienten  $c$ , die durch Berechnung der inversen Diskreten Cosinus-Transformation von  $f$  erhalten wurden, mit zwei Spikes bei den korrekten Frequenzen. Weil die beiden Frequenzen unvereinbar sind, fällt dieses Signal nicht genau in den von den DCT-Basisfunktionen abgedeckten Bereich, weshalb es einige Dutzend signifikante von Null verschiedene Koeffizienten gibt.

Für unser Beispiel ist das komprimierte Signal ein Vektor  $b$  aus  $m$  zufälligen Stichproben des Ursprungssignals. Wir konstruieren eine Matrix  $A$ , indem wir  $m$  Zeilen aus der  $n$ -mal- $n$ -DCT-Matrix extrahieren

$$D = \text{dct}(\text{eye}(n, n));$$

$$A = D(k, :)$$

wobei  $k$  der Vektor der für die Stichprobe  $b$  verwendeten Indizes ist. Das resultierende lineare System,  $Ax = b$ , hat die Dimension  $m$ -mal- $n$ , also 500-mal-5000. Es gibt 10 Mal so viele Unbekannte wie Gleichungen.

Zur Rekonstruktion des Signals müssen wir diejenige Lösung für  $Ax = b$  finden, die die  $l_1$ -Norm von  $x$  minimiert. Dies ist ein nichtlineares Optimierungsproblem und es gibt eine ganze Reihe von MATLAB-basierten Programmen zu seiner Lösung. Ich habe mich für  $l_1$ -magic entschieden, das von Justin Romberg und Emmanuel Candès während ihrer Zeit am Caltech geschrieben wurde. Das obere Diagramm in Abbildung 2 zeigt die resultierende Lösung,  $x$ . Wir sehen,

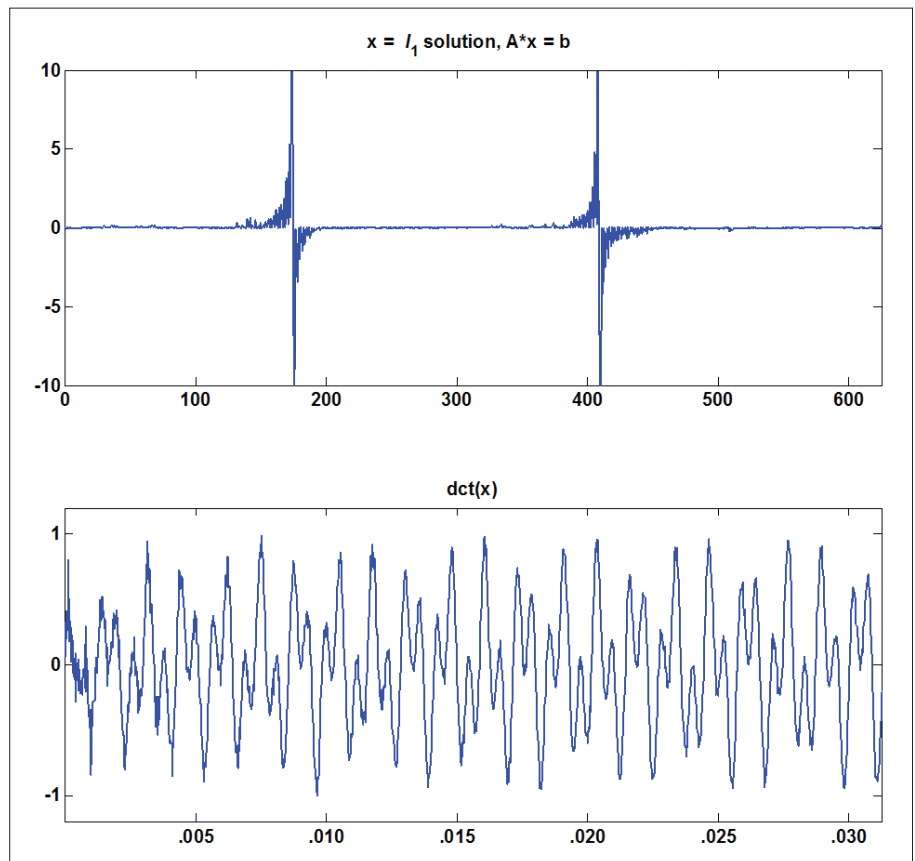


ABB. 2. Die  $l_1$ -Lösung für  $Ax = b$  erzeugt  $\text{dct}(x)$ , ein Signal, das fast identisch mit dem Original ist.

dass sie relativ wenige große Komponenten besitzt und dass sie der DCT des Ursprungssignals sehr nahe kommt. Darüber hinaus sieht die im unteren Diagramm dargestellte Diskrete Cosinus-Transformation von  $x$  dem Ursprungssignal ebenfalls sehr ähnlich. Könnten wir hier Töne wiedergeben, dann würden Sie hören, dass die Audioausgaben der beiden Befehle `sound(f)` und `sound(dct(x))` fast gleich klingen.

Zum Vergleich zeigt Abbildung 3 die übliche  $l_2$ - oder Kleinste-Quadrate-Lösung für  $Ay = b$ , die berechnet wird durch

$$y = \text{pinv}(A) * b$$

In den Diagrammen erkennt man einen schwachen Anflug des Ursprungssignals,

aber `sound(dct(y))` ist lediglich ein verrauschtes Brummen.

Es ist noch zu früh um vorauszusagen, wann – oder ob – wir Compressed Sensing in unseren Mobiltelefonen, Digitalkameras und Magnetresonanz-Scannern antreffen werden, aber ich finde die zugrunde liegende Mathematik und Software faszinierend. ■

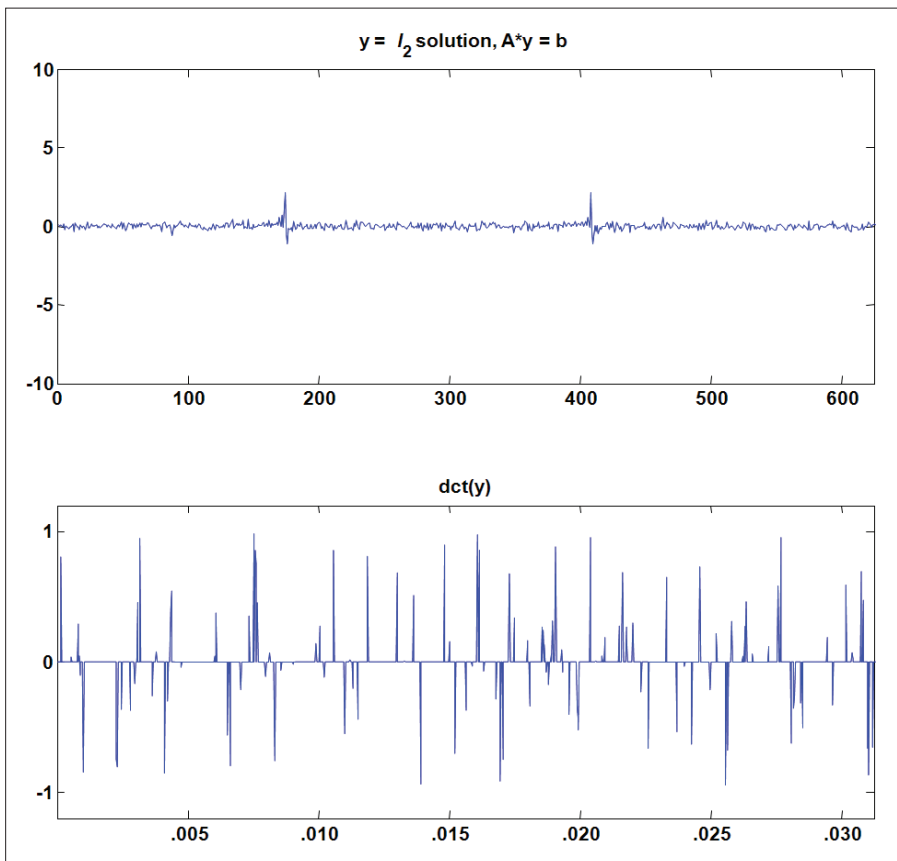


ABB. 3. Die  $l_2$ -Lösung für  $Ay = b$  erzeugt  $dct(y)$ , ein Signal, das mit dem Original nur wenig Ähnlichkeit hat.

## Literatur

### Ein 20 Jahre altes Cleve's Corner:

“The World’s Simplest Impossible Problem,” *News & Notes* (1990)

[www.mathworks.de/clevescorner/dec1990](http://www.mathworks.de/clevescorner/dec1990)

### Eine Quelle mit über 600 Links, von denen sich 36 auf MATLAB beziehen:

“Compressive Sensing Resources,”

Rice University

<http://dsp.rice.edu/cs>

### Zwei Übersichtsartikel für ein allgemeines technisches Publikum:

Dana Mackenzie, “Compressed Sensing Makes Every Pixel Count,” *What’s Happening in the Mathematical Sciences* 7 (2009),

American Math Society

[www.ams.org/happening-series/hap7-pixel.pdf](http://www.ams.org/happening-series/hap7-pixel.pdf)

Bryan Hayes, “The Best Bits,” *American Scientist* 97, no.4 (2009)

[www.americanscientist.org/issues/pub/the-best-bits/7](http://www.americanscientist.org/issues/pub/the-best-bits/7)

### Eines von vielen Tutorien:

Richard Baraniuk, Justin Romberg, and Michael Wakin, “Tutorial on Compressive Sensing,” *Information Theory and Applications Workshop* (2008)

[www.ece.rice.edu/~richb/talks/cs-tutorial-ITA-feb08-complete.pdf](http://www.ece.rice.edu/~richb/talks/cs-tutorial-ITA-feb08-complete.pdf)

### MATLAB-basierte Software:

Justin Romberg,  $l_1$ -magic

[www.acm.caltech.edu/l1magic](http://www.acm.caltech.edu/l1magic)

### Die zwei Original-Artikel:

Emmanuel Candès, Justin Romberg, and Terence Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, 52(2): 489 - 509, February 2006

[www.acm.caltech.edu/~jrom/publications/CandesRombergTao\\_revisedNov2005.pdf](http://www.acm.caltech.edu/~jrom/publications/CandesRombergTao_revisedNov2005.pdf)

David Donoho. “Compressed Sensing,”

*IEEE Transactions on Information Theory*, 52(4): 1289 - 1306, April 2006

[www-stat.stanford.edu/~donoho/Reports/2004/CompressedSensing091604.pdf](http://www-stat.stanford.edu/~donoho/Reports/2004/CompressedSensing091604.pdf)

### Eine Audio-Demonstration:

Laura Balzano, Robert Nowak, and Jordan Ellenberg, “Compressed Sensing Audio Demonstration”

<http://sunbeam.ece.wisc.edu/csaudio/>

## Weitere Informationen online

### MATLAB-CODE ZUM DOWNLOAD

[www.mathworks.de/mlc28250](http://www.mathworks.de/mlc28250)

### CLEVE'S CORNER COLLECTION

[www.mathworks.de/clevescorner](http://www.mathworks.de/clevescorner)