Data Compression, compressed sensing and geometry of Banach spaces.

Digitalisation and compression

Let f be a function on the domain $(0,1)^n$ in $\mathbb{R}^n$.

Digitalisation and compression

Let f be a function on the domain $(0, 1)^n$ in $\mathbb{R}^n$.

If $n = 1$, f may represent a sound; $f(t) =$ air pressure at time t.

Digitalisation and compression

Let f be a function on the domain $(0,1)^n$ in $\mathbb{R}^n$.

If $n = 1$, f may represent a sound; $f(t) =$ air pressure at time t.

If $n = 2$, f may represent a picture, $f(t)$ being the *greyscale* of the picture at point t.

Digitalisation and compression

Let f be a function on the domain $(0,1)^n$ in $\mathbb{R}^n$.

If $n = 1$, f may represent a sound; $f(t) =$ air pressure at time t.

If $n = 2$, f may represent a picture, $f(t)$ being the *greyscale* of the picture at point t.

We want to associate to f a discrete, or better, finite, set of numbers, from which it is possible to reconstruct f up to a small error.

Digitalisation and compression

Let f be a function on the domain $(0,1)^n$ in $\mathbb{R}^n$.

If $n = 1$, f may represent a sound; $f(t) =$ air pressure at time t.

If $n = 2$, f may represent a picture, $f(t)$ being the *greyscale* of the picture at point t.

We want to associate to f a discrete, or better, finite, set of numbers, from which it is possible to reconstruct f up to a small error.

I will stick to the case $n = 1$ for simplicity, but the principle carries over to arbitrary dimension.

Digitalisation and compression

Let f be a function on the domain $(0,1)^n$ in $\mathbb{R}^n$.

If $n = 1$, f may represent a sound; $f(t) =$ air pressure at time t.

If $n = 2$, f may represent a picture, $f(t)$ being the *greyscale* of the picture at point t.

We want to associate to f a discrete, or better, finite, set of numbers, from which it is possible to reconstruct f up to a small error.

I will stick to the case $n = 1$ for simplicity, but the principle carries over to arbitrary dimension.

One may also consider a discrete version, replacing $(0,1)$ by $\{k/M; k = 0, 1, ...M - 1\}$ (think of pixels in a picture).

Digitalisation and compression

Let f be a function on the domain $(0,1)^n$ in $\mathbb{R}^n$.

If $n = 1$, f may represent a sound; $f(t) =$ air pressure at time t.

If $n = 2$, f may represent a picture, $f(t)$ being the *greyscale* of the picture at point t.

We want to associate to f a discrete, or better, finite, set of numbers, from which it is possible to reconstruct f up to a small error.

I will stick to the case $n = 1$ for simplicity, but the principle carries over to arbitrary dimension.

One may also consider a discrete version, replacing $(0,1)$ by $\{k/M; k = 0, 1, ...M - 1\}$ (think of pixels in a picture). Then the data is already finite, but we may want to replace it by a smaller set of data.

The Fourier method

Recall that we can write f as a Fourier series:

$$f(t) = \sum_{-\infty}^{\infty} c_k(f) e^{2\pi ikt},$$

where

$$c_k(f) = \int_0^1 f(t) e^{-2\pi ikt} dt$$

are the Fourier coefficients of f (cf slides 4Persp).

The Fourier method

Recall that we can write f as a Fourier series:

$$f(t) = \sum_{-\infty}^{\infty} c_k(f) e^{2\pi ikt},$$

where

$$c_k(f) = \int_0^1 f(t) e^{-2\pi ikt} dt$$

are the Fourier coefficients of f (cf slides 4Persp).
The discrete version of this is

$$f(t) = \sum_{0}^{M-1} c_k(f) e^{2\pi ikt/M}, \quad c_k = (1/M) \sum_{0}^{M-1} f(t) e^{2\pi ikt/M}.$$

The Fourier method

Recall that we can write f as a Fourier series:

$$f(t) = \sum_{-\infty}^{\infty} c_k(f) e^{2\pi ikt},$$

where

$$c_k(f) = \int_0^1 f(t) e^{-2\pi ikt} dt$$

are the Fourier coefficients of f (cf slides 4Persp).
The discrete version of this is

$$f(t) = \sum_0^{M-1} c_k(f) e^{2\pi ikt/M}, \quad c_k = (1/M) \sum_0^{M-1} f(t) e^{2\pi ikt/M}.$$

Exercise: Prove the discrete version!

Compression

This way we get the function represented by an infinite sequence of values $c_k(f)$. To get a finite set of values one may simply throw away the numbers that are small, or correspond to high frequences.

Compression

This way we get the function represented by an infinite sequence of values $c_k(f)$. To get a finite set of values one may simply throw away the numbers that are small, or correspond to high frequences.

But, there is not so much reason to believe that this will be the most efficient way of storing the signal, when efficiency is measured in terms of how many coefficients you need and how good your approximation

$$\sum_{k \in K} c_k(f) e^{2\pi i k t/M}$$

is.

Compression

This way we get the function represented by an infinite sequence of values $c_k(f)$. To get a finite set of values one may simply throw away the numbers that are small, or correspond to high frequences.

But, there is not so much reason to believe that this will be the most efficient way of storing the signal, when efficiency is measured in terms of how many coefficients you need and how good your approximation

$$\sum_{k \in K} c_k(f) e^{2\pi ikt/M}$$

is.

In many cases it turns out that one can take advantage of the fact that the functions one is interested in are rather special. . For instance, in the case of a picture, they tend to be almost constant over relatively large regions.

Compression

This way we get the function represented by an infinite sequence of values $c_k(f)$. To get a finite set of values one may simply throw away the numbers that are small, or correspond to high frequences.

But, there is not so much reason to believe that this will be the most efficient way of storing the signal, when efficiency is measured in terms of how many coefficients you need and how good your approximation

$$\sum_{k \in K} c_k(f) e^{2\pi ikt/M}$$

is.

In many cases it turns out that one can take advantage of the fact that the functions one is interested in are rather special. . For instance, in the case of a picture, they tend to be almost constant over relatively large regions.

In many cases it is better to use a method based on *wavelets*, i. e. to develop the function f in another basis than the Fourier basis $e_k = e^{2\pi ikt}$.

Informal motivation

Let us think of f as describing a picture, where $f(t) = 0$ means the picture is white at t, and $f(t) = 7$ means totally black. A uniformly grey picture would have, say, $f = 1$ everywhere.

Informal motivation

Let us think of f as describing a picture, where $f(t) = 0$ means the picture is white at t, and $f(t) = 7$ means totally black. A uniformly grey picture would have, say, $f = 1$ everywhere.

That is perhaps the simplest picture and the Fourier coefficients are also simple: All $c_k(f)$ are zero except c_0.

Informal motivation

Let us think of f as describing a picture, where $f(t) = 0$ means the picture is white at t, and $f(t) = 7$ means totally black. A uniformly grey picture would have, say, $f = 1$ everywhere.

That is perhaps the simplest picture and the Fourier coefficients are also simple: All $c_k(f)$ are zero except c_0.

Now say that the picture is grey on the left half, and white on the right half. That is also a very simple picture, but the Fourier coefficients are not simple at all; they tend to zero rather slowly when $k \to \infty$.

Informal motivation

Let us think of f as describing a picture, where $f(t) = 0$ means the picture is white at t, and $f(t) = 7$ means totally black. A uniformly grey picture would have, say, $f = 1$ everywhere.

That is perhaps the simplest picture and the Fourier coefficients are also simple: All $c_k(f)$ are zero except c_0.

Now say that the picture is grey on the left half, and white on the right half. That is also a very simple picture, but the Fourier coefficients are not simple at all; they tend to zero rather slowly when $k \to \infty$.

The reason is that the Fourier basis $e^{2\pi ikt}$ is not well localized in space (only in frequency), so it does not take advantage of the simple structure of the picture.

Wavelets

Wavelets form a different basis of functions, that are more localized in space.

Wavelets

Wavelets form a different basis of functions, that are more localized in space.

One starts from one function $\psi(t)$; the *mother wavelet*. Then one constructs the other basis elements as (something like)

$$\psi_{mn}(t) = 2^{m/2}\psi(2^m t - n), a > 1.$$

Wavelets

Wavelets form a different basis of functions, that are more localized in space.

One starts from one function $\psi(t)$; the *mother wavelet*. Then one constructs the other basis elements as (something like)

$$\psi_{mn}(t) = 2^{m/2}\psi(2^m t - n), a > 1.$$

If ψ is well localized around zero, i. e. (almost) vanishes when $|t| > \epsilon$, then

$$\psi(2^m t)$$

vanishes when $|t| > 2^{-m}\epsilon$, so it is even more localized around 0.

Wavelets

Wavelets form a different basis of functions, that are more localized in space.

One starts from one function $\psi(t)$; the *mother wavelet*. Then one constructs the other basis elements as (something like)

$$\psi_{mn}(t) = 2^{m/2}\psi(2^m t - n), a > 1.$$

If ψ is well localized around zero, i. e. (almost) vanishes when $|t| > \epsilon$, then

$$\psi(2^m t)$$

vanishes when $|t| > 2^{-m}\epsilon$, so it is even more localized around 0.

$$\psi(2^m t - n) = \psi(2^m(t - n2^{-m}))$$

is localised around $n2^{-m}$.

If ψ oscillates, ψ_{mn} will oscillate even more when m is large.

If ψ oscillates, ψ_{mn} will oscillate even more when m is large.

If we choose the mother wavelet carefully, we can make ψ_{mn} an orthonormal basis. We can then define

$$c_{mn}(f) = \int f(t)\psi_{mn}(t)dt,$$

and get

$$f(t) = \sum c_{mn}(f)\psi_{mn}(t).$$

If ψ oscillates, ψ_{mn} will oscillate even more when m is large.

If we choose the mother wavelet carefully, we can make ψ_{mn} an orthonormal basis. We can then define

$$c_{mn}(f) = \int f(t)\psi_{mn}(t)dt,$$

and get

$$f(t) = \sum c_{mn}(f)\psi_{mn}(t).$$

This uses much more the uniform character of f (almost constant on large regions for instance), and makes many coefficients very small. The reason for this is that most coefficients corresponding to wavelets supported in the region where f is nearly constant are very small.

So, discarding coefficients gives a better way of storing f.

JPEG

This method is used e. g. in the JPEG-method of storing pictures. Let us look at the discrete setting (there are discrete variants of the wavelet).

JPEG

This method is used e. g. in the JPEG-method of storing pictures. Let us look at the discrete setting (there are discrete variants of the wavelet).

A picture can be described by a function on

$$\{k/M; k = 0, 1, ...M - 1\}^2$$

which contains M^2 points. The function f at a point t is the greyscale at that point, taking values $0, 1, ..7$ (say). M is typically around one thousand, so the number of pixels is of the order of magnitude $N = 10^6$, a million. Thus the number of possible pictures is 2^{3N}, which means that the data in one picture represents 3 million bits, i. e. one megabyte of information.

JPEG

This method is used e. g. in the JPEG-method of storing pictures. Let us look at the discrete setting (there are discrete variants of the wavelet).
A picture can be described by a function on

$$\{k/M; k = 0, 1, ...M - 1\}^2$$

which contains M^2 points. The function f at a point t is the greyscale at that point, taking values $0, 1, ..7$ (say). M is typically around one thousand, so the number of pixels is of the order of magnitude $N = 10^6$, a million. Thus the number of possible pictures is 2^{3N}, which means that the data in one picture represents 3 million bits, i. e. one megabyte of information.

The vast majority of these 'pictures' are just meaningless, and do not appear in practice. (Compare the number of possible 'sentences' with one hundred letters, and the number of meaningful ones.)

JPEG

This method is used e. g. in the JPEG-method of storing pictures. Let us look at the discrete setting (there are discrete variants of the wavelet).

A picture can be described by a function on

$$\{k/M; k = 0, 1, ...M - 1\}^2$$

which contains M^2 points. The function f at a point t is the greyscale at that point, taking values $0, 1, ..7$ (say). M is typically around one thousand, so the number of pixels is of the order of magnitude $N = 10^6$, a million. Thus the number of possible pictures is 2^{3N}, which means that the data in one picture represents 3 million bits, i. e. one megabyte of information.

The vast majority of these 'pictures' are just meaningless, and do not appear in practice. (Compare the number of possible 'sentences' with one hundred letters, and the number of meaningful ones.) Applying the wavelet transform and keeping only around 10^5 coefficients one gets a good picture. So, we can compress the picture.

Compressed sensing

A natural question arises: If we are only gooing to keep 100kB, do we really need to measure 1MB, and throw away 90% of the info?

Compressed sensing

A natural question arises: If we are only gooing to keep 100kB, do we really need to measure 1MB, and throw away 90% of the info? Or, is it possible to measure (sense) just the 100 kB directly?

Compressed sensing

A natural question arises: If we are only gooing to keep 100kB, do we really need to measure 1MB, and throw away 90% of the info? Or, is it possible to measure (sense) just the 100 kB directly?

The answer to this is the method of compressed *sensing*. (Invented by Donoho, Candès, Tao, ...around 2000+5.)

Compressed sensing

A natural question arises: If we are only gooing to keep 100kB, do we really need to measure 1MB, and throw away 90% of the info? Or, is it possible to measure (sense) just the 100 kB directly?

The answer to this is the method of compressed *sensing*. (Invented by Donoho, Candès, Tao, ...around 2000+5.)
The problem is as follows. We have a vector x in $\mathbb{R}^N$ where N is around 1 million; the wavelet coefficients of a function describing a picture.

Compressed sensing

A natural question arises: If we are only gooing to keep 100kB, do we really need to measure 1MB, and throw away 90% of the info? Or, is it possible to measure (sense) just the 100 kB directly?

The answer to this is the method of compressed *sensing*. (Invented by Donoho, Candès, Tao, ...around 2000+5.)
The problem is as follows. We have a vector x in $\mathbb{R}^N$ where N is around 1 million; the wavelet coefficients of a function describing a picture. Experience tells us that only $s = 10^5$) of the coeffcients are significantly different from 0, but we don't know which ones if we have not taken the full picture and computed the wavelet coefficients. We want to find a good approximation of the vector from roughly m measurements. One expects to need $m > s$, and hopes to get away with $m << N$.

It is instructive to look first at an *extreme case*; we assume that most of the coefficients are not only small but *vanish completely*.

It is instructive to look first at an *extreme case*; we assume that most of the coefficients are not only small but *vanish completely*.

Let

$$x_i = (f, \psi_i),$$

the scalar product between f and the i:th wavelet. We assume that the vector x is s-sparse, i. e. only s of the x_i are nonzero.

It is instructive to look first at an *extreme case*; we assume that most of the coefficients are not only small but *vanish completely*.

Let

$$x_i = (f, \psi_i),$$

the scalar product between f and the i:th wavelet. We assume that the vector x is s-sparse, i. e. only s of the x_i are nonzero.

A measurement apparatus is a collection of m vectors ϕ_k, which applied to f gives m numbers (f, ϕ_k), or equaivalently a map

$$x \to \Phi x \in \mathbb{R}^m.$$

We want to design Φ so that m is not too big, but so that we can still reconstruct x, f from Φx, under the assumption that x is s-sparse.

Mathematical formulation

Let Φ be a linear map from $\mathbb{R}^N \to \mathbb{R}^m$. Let x be an s-sparse vector, meaning that only s of its entries x_i are different from 0.

Mathematical formulation

Let Φ be a linear map from $\mathbb{R}^N \to \mathbb{R}^m$. Let x be an s-sparse vector, meaning that only s of its entries x_i are different from 0.

We know

$$y = \Phi x$$

How do we find x?

Mathematical formulation

Let Φ be a linear map from $\mathbb{R}^N \to \mathbb{R}^m$. Let x be an s-sparse vector, meaning that only s of its entries x_i are different from 0.

We know

$$y = \Phi x$$

How do we find x? There are many solutions to the equation $\Phi x = y$. They form an *affine subspace* V of $\mathbb{R}^N$, an $(N - m)$-dimensional plane, in general not going through the origin.

Mathematical formulation

Let Φ be a linear map from $\mathbb{R}^N \to \mathbb{R}^m$. Let x be an s-sparse vector, meaning that only s of its entries x_i are different from 0.

We know

$$y = \Phi x$$

How do we find x? There are many solutions to the equation $\Phi x = y$. They form an *affine subspace* V of $\mathbb{R}^N$, an $(N - m)$-dimensional plane, in general not going through the origin.

The method of least squares gives the vector x that solves

$$\min_{x \in V} \sum x_i^2.$$

Mathematical formulation

Let Φ be a linear map from $\mathbb{R}^N \to \mathbb{R}^m$. Let x be an s-sparse vector, meaning that only s of its entries x_i are different from 0.

We know

$$y = \Phi x$$

How do we find x? There are many solutions to the equation $\Phi x = y$. They form an *affine subspace* V of $\mathbb{R}^N$, an $(N - m)$-dimensional plane, in general not going through the origin.

The method of least squares gives the vector x that solves

$$\min_{x \in V} \sum x_i^2.$$

Consider instead

$$\min_{x \in V} \sum |x_i|. \tag{1}$$

Theorem of Candès and Tao, 2006

Theorem

Let x be an s-sparse vector in $\mathbb{R}^N$ and let

$$m \geq Cs \log N.$$

Choose Φ more or less at random (to be explained). Then 'with overwhelming probability', x is the unique solution of

$$\min_{x \in V} \sum |x_i|.$$

Theorem of Candès and Tao, 2006

Theorem

Let x be an s-sparse vector in $\mathbb{R}^N$ and let

$$m \geq Cs \log N.$$

Choose Φ more or less at random (to be explained). Then 'with overwhelming probability', x is the unique solution of

$$\min_{x \in V} \sum |x_i|.$$

Similarly, if all entries x_i with i outside a set of cardinality s are *small*, then the solution to the 'least sum problem' will give a good approximation to x.

Theorem of Candès and Tao, 2006

Theorem

Let x be an s-sparse vector in $\mathbb{R}^N$ and let

$$m \geq Cs \log N.$$

Choose Φ more or less at random (to be explained). Then 'with overwhelming probability', x is the unique solution of

$$\min_{x \in V} \sum |x_i|.$$

Similarly, if all entries x_i with i outside a set of cardinality s are *small*, then the solution to the 'least sum problem' will give a good approximation to x. The point is that we don't need to construct a particular cleverly chosen Φ, almost any choice will do.

Why does it work?

Take $m > s$. Then $V = \{\Phi x = 0\}$ has dimension $N - m$.

Why does it work?

Take $m > s$. Then $V = \{\Phi x = 0\}$ has dimension $N - m$.

Then look at the s dimensional coordinate planes; the sets where only s coordinates are different from 0. They, naturally, have dimension s. If $m > s$, most choices of V of dimension $N - m$ will not intersect any such coordinate plane:

Why does it work?

Take $m > s$. Then $V = \{\Phi x = 0\}$ has dimension $N - m$.

Then look at the s dimensional coordinate planes; the sets where only s coordinates are different from 0. They, naturally, have dimension s. If $m > s$, most choices of V of dimension $N - m$ will not intersect any such coordinate plane:

If W is such a coordinate plane intersecting V we have a solution to $y = \Phi x$ in W, which is m equations with $s < m$ unknowns, which mostly cannot be solved.

Why does it work?

Take $m > s$. Then $V = \{\Phi x = 0\}$ has dimension $N - m$.

Then look at the s dimensional coordinate planes; the sets where only s coordinates are different from 0. They, naturally, have dimension s. If $m > s$, most choices of V of dimension $N - m$ will not intersect any such coordinate plane:

If W is such a coordinate plane intersecting V we have a solution to $y = \Phi x$ in W, which is m equations with $s < m$ unknowns, which mostly cannot be solved.

But in this case we have assumed that we have a solution, which means that it most likely is the only intersection of V with such a coordinate plane. This is the point we must find.

Look at first at the solution to the least square problem

$$\min_{x \in V} \sum x_i^2.$$

This we can obtain by looking at balls centered at the origin with radius r. We start with r small and let r grow until we hit V. The first point of contact is the solution to the least square problem. (Draw a figure!)

Look at first at the solution to the least square problem

$$\min_{x \in V} \sum x_i^2.$$

This we can obtain by looking at balls centered at the origin with radius r. We start with r small and let r grow until we hit V. The first point of contact is the solution to the least square problem. (Draw a figure!)

Now look at the least sum problem. We can obtain it the same way, but now the 'balls" are

$$B_r = \{x; \sum |x_i| < r\}.$$

Look at first at the solution to the least square problem

$$\min_{x \in V} \sum x_i^2.$$

This we can obtain by looking at balls centered at the origin with radius r. We start with r small and let r grow until we hit V. The first point of contact is the solution to the least square problem. (Draw a figure!)

Now look at the least sum problem. We can obtain it the same way, but now the 'balls" are

$$B_r = \{x; \sum |x_i| < r\}.$$

From a figure one is easily convinced that the first point of contact as r grows will always be on a coordinate plane. (Draw a figure!)

Look at first at the solution to the least square problem

$$\min_{x \in V} \sum x_i^2.$$

This we can obtain by looking at balls centered at the origin with radius r. We start with r small and let r grow until we hit V. The first point of contact is the solution to the least square problem. (Draw a figure!)

Now look at the least sum problem. We can obtain it the same way, but now the 'balls" are

$$B_r = \{x; \sum |x_i| < r\}.$$

From a figure one is easily convinced that the first point of contact as r grows will always be on a coordinate plane. (Draw a figure!) And, we have assumed that there is only one such point. So we have reconstructed x, as the unique solution to the 'least sum problem'!