Mathematical Programming: Modelling and Applications

Leo Liberti

LIX, École Polytechnique

liberti@lix.polytechnique.fr

January 2018

Outline

Compressed sensing

Compressed sensing is an approach to signal processing that allows for signals and images to be reconstructed with lower sampling rates than with Shannon-Nyquist's Law.

This makes signal processing and reconstruction much simpler and has a wide variety of applications in the real world, including photography, holography and facial recognition.

Compressed sensing is also known as compressive sensing, compressive sampling and sparse sampling.

Compressed sensing: exe. 1

We want to send a message **w** (“Hello world!", “Carpe diem", etc.) and we want to make sure that all components of **w** can be recovered correctly even if some fraction of the transmitted numbers are corrupted due to random errors.

The message have to be encrypted using a “key".

We use AMPL to simulate a message **w** sending and receiving session.

Compressed sensing: exe. 1

The message is sent as a sequence of Boolean values (i.e. ASCII code of each character of the message in base-2): download the script makedat.run and use it to convert your message and to create a suitable .dat file (usage: ampl < makedat.run > file.dat) or use the sample files hello.dat, ciao bella.dat.

Create a AMPL script that:

- includes the .dat file
- codes the message using a “key"

Remark: both sender and receiver know the key.

- decodes the message (using just the received message and the key)
- checks if original and received message are different and count the errors
- displays original and received message

Hint: to encrypt the message, choose a **square** matrix Q and “send" a vector z = Qw. In other words, the matrix Q is your “key”.

Compressed sensing: exe. 1

Reminder: in an ampl .run script you can use the command problem in the following way:

var x;
minimize *your objective*: .....;
subject to *your constraint*....;
problem *your problem*: *x, your objective,your constraint*;
option solver cplex;
solve *your problem*;

Compressed sensing: exe. 2

Do the same, as in exercise 1, but now code the message using a matrix Q, $n * k$ with $n < k$, being k the length of the message w (i.e. the message is compressed).

Hint: seek for the sparsest vector.
Hint: relax the problem of minimizing norm 0 to the problem of minimizing norm 1.

Compressed sensing: exe. 3

Now code a message using a matrix Q, $n * k$ with $n > k$ (in particular $n = 2 * k$).
In other words, the message is stretched.
Moreover, the message is garbled. Thus, before sending it, add a casual error (i.e. add a sparse vector, namely with less than 10% of non-zero values)

Hint: attention, Q is not square.

Compressed sensing: exe. 4

Use random projections to reduce the dimension of the matrix.
Build your projector matrix R using Achlioptas' method or similar.

Basically, for your first trial, build your projector matrix R as follows:

$R_{i,j} = 1/\sqrt{k}$

OR

$R_{i,j} = -1/\sqrt{k}$

OR

zero

in a random way, so that zero values are more frequent and the other two values have the same probability.

Compressed sensing: exe. 5

Build your projector using other methods.