Analog-to-digital conversion in compressive sampling

Özgür Yılmaz

Department of Mathematics
University of British Columbia

December 8, 2014

Collaborators

Joint work with:

- Rayan Saab (UCSD)
- Rongrong Wang (UBC)

Will also mention work with:

- Sinan Güntürk (Courant)
- Felix Krahmer (Göttingen)
- Mark Lammers (UNCW)
- Alex Powell (Vanderbilt)

Main problem

Goal: Acquire an analog signal and *store/process/transmit it digitally*. Main focus in this meeting.

Signal Model: x is a high dimensional signal (say in $\mathbb{R}^N$) that is sparse w.r.t. some basis/frame.

Sampling technique: Compressive: Non-adaptive, linear "generalized measurements" $\langle \phi_i, x \rangle$.

$$x \in \mathbb{R}^N \xrightarrow{\Phi} y \in \mathbb{R}^m$$

Here: Φ is an $m \times N$ compressive sampling matrix ($m \ll N$), e.g., random subsampling.

Problem: The compressive samples are analog quantities. Accordingly:

- Compressive sampling is an efficient dimension reduction method.
- Dimension reduction is not compression: we need to have error analysis in terms of the bit budget.
- We will focus on this (and putting "compressive" back to "compressive sampling").

Main problem

Goal: Acquire an analog signal and *store/process/transmit it digitally*. Main focus in this meeting.

Signal Model: x is a high dimensional signal (say in $\mathbb{R}^N$) that is sparse w.r.t. some basis/frame.

Sampling technique: Compressive: Non-adaptive, linear "generalized measurements" $\langle \phi_i, x \rangle$.

$$x \in \mathbb{R}^N \xrightarrow{\Phi} y \in \mathbb{R}^m \xrightarrow{Q} q \in \mathcal{A}^m \xrightarrow{\mathcal{E}} \tilde{q} \xrightarrow{\Delta_{Q,\mathcal{E}}} x^\# \in \mathbb{R}^N$$

Color code: Acquisition, A/D conversion, Compression, Decoding

Main problem

Goal: Acquire an analog signal and *store/process/transmit it digitally*. Main focus in this meeting.

Signal Model: x is a high dimensional signal (say in $\mathbb{R}^N$) that is sparse w.r.t. some basis/frame.

Sampling technique: Compressive: Non-adaptive, linear "generalized measurements" $\langle \phi_i, x \rangle$.

$$x \in \mathbb{R}^N \xrightarrow{\Phi} y \in \mathbb{R}^m \xrightarrow{Q} q \in \mathcal{A}^m \xrightarrow{\mathcal{E}} \tilde{q} \xrightarrow{\Delta_{Q,\mathcal{E}}} x^\# \in \mathbb{R}^N$$

Color code: Acquisition, A/D conversion, Compression, Decoding

Here: Φ is an $m \times N$ compressive sampling matrix ($m \ll N$).

Want: Design Q, $\mathcal{E}$, and $\Delta_{Q,\mathcal{E}}$ such that

- $\sup_{x \in K} \|x^\# - x\|$ is (nearly) optimally small for a given bit budget R.
- Q is robustly implementable on analog hardware.
- $\Delta_{Q,\mathcal{E}}$ is tractable.

Why? Any applications in seismic?

Signal acquisition devices, e.g., geophones, have several design bottlenecks:

- battery life (wireless)
- hardware complexity
- storage

Given the humongous size of the data we collect, we want to explore if we can incorporate compressive sensing into the analog-to-digital conversion stage where

- we can compress without requiring to perform transform coding (thus save on storage and battery life required for transmission)
- we collect less samples (thus save on storage and battery life).

The schemes we will propose are simple to implement, thus realistic. In addition, they achieve exponential accuracy.

Notations: compressed sensing

- $x \in \mathbb{R}^N$ is k-sparse if x has at most k non-zero entries.
- $\Sigma_k^N := \{x \in \mathbb{R}^N : \ x \text{ is } k\text{-sparse}\}$
- Measurement matrix: Φ, an $m \times N$ real matrix.
- Measurements: $y = \Phi x + e$ (e denotes additive noise)
- Dimensional setting: $k < m < N$.

Main conclusion of CS. Suppose $x \in \Sigma_k$ or can be well approximated from Σ_k. Given the (noisy) measurements $y = \Phi x + e$, one can recover x exactly (approximately), in a computationally efficient manner. The reconstruction is robust to noise and stable with respect to model mismatch.

Notations: quantization

Quantizer: Any map $Q: \mathbb{R}^m \mapsto \mathcal{A}^m$ where the alphabet $\mathcal{A}$ is a finite or discrete set.

Scalar quantizer with alphabet $\mathcal{A}$ is the map

$$Q_{\mathcal{A}}: \ x \in \mathbb{R} \mapsto \arg\min_{v \in \mathcal{A}} |x - v| \in \mathcal{A}.$$

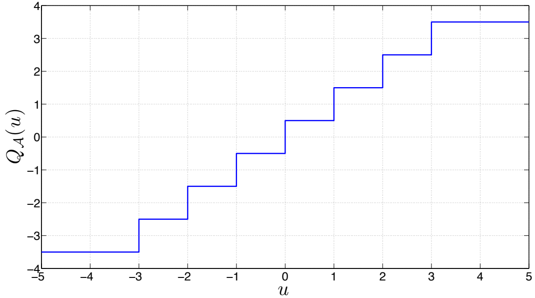
Bit depth of the scalar quantizer $Q_{\mathcal{A}}$ is $b = \log_2 |\mathcal{A}|$.

Midrise uniform quantizer with step size δ: $Q_{\mathcal{A}_L^\delta}$ with

$$\mathcal{A}_L^\delta = \{\pm(2j+1)\delta/2 : \ j \in [L]\}.$$

Corresponding bit-depth: $b = \log_2(2L)$ bits.

A midrise quantizer

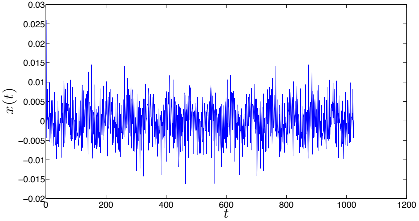


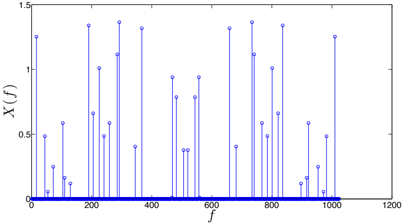
Above:

- $\mathcal{A} = \{-3.5, -2.5, -1.5, -0.5, 0.5, 1.5, 2.5, 3.5\} =: \mathcal{A}_3^1$,
- $b = \log_2 8 = 3$
- $|x - Q_{\mathcal{A}}(u)| \leq 1/2$ provided $|u| \leq 4$.
- The midrise scalar quantizer saturates if $|u| > 4$

An illustrative example

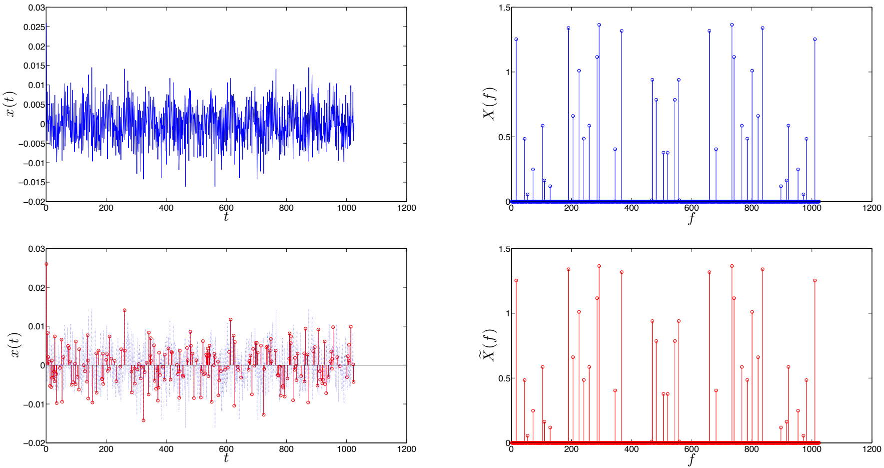
Below, $x \in \mathbb{R}^{1024}$, 40-sparse w.r.t. Fourier basis. We collect 200 uniformly random samples in the time domain and reconstruct using SPGL1. Note: $\|x - \tilde{x}\|_\infty \leq 3 \times 10^{-2}$





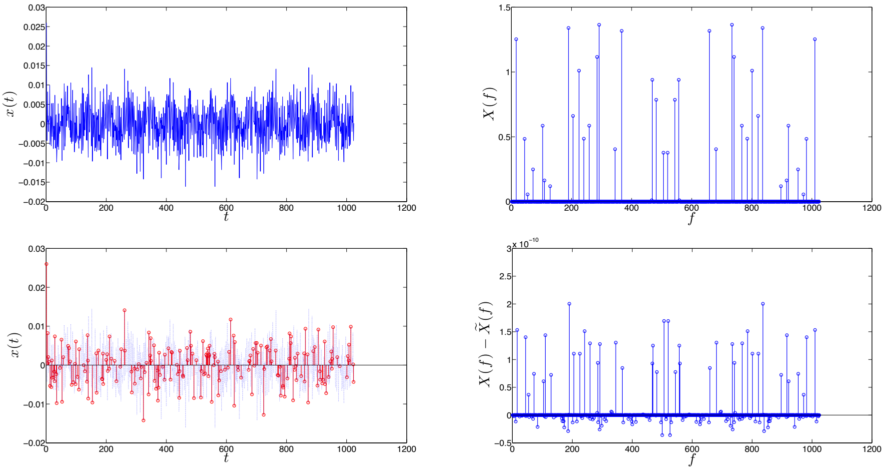
An illustrative example

Below, $x \in \mathbb{R}^{1024}$, 40-sparse w.r.t. Fourier basis. We collect 200 uniformly random samples in the time domain and reconstruct using SPGL1. Note: $\|x - \tilde{x}\|_\infty \leq 3 \times 10^{-2}$



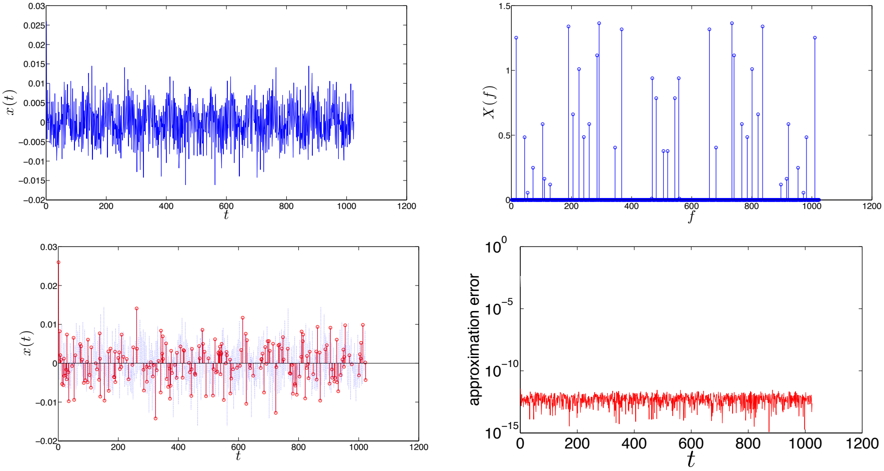
An illustrative example

Below, $x \in \mathbb{R}^{1024}$, 40-sparse w.r.t. Fourier basis. We collect 200 uniformly random samples in the time domain and reconstruct using SPGL1. Note: $\|x - \tilde{x}\|_\infty \leq 3 \times 10^{-2}$



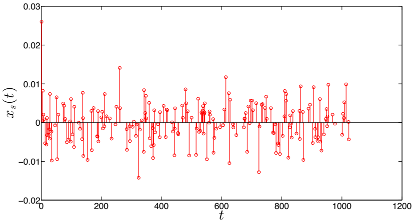
An illustrative example

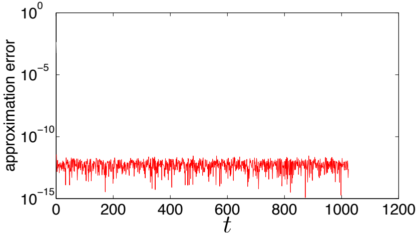
Below, $x \in \mathbb{R}^{1024}$, 40-sparse w.r.t. Fourier basis. We collect 200 uniformly random samples in the time domain and reconstruct using SPGL1. Note: $\|x - \tilde{x}\|_\infty \leq 3 \times 10^{-2}$



An illustrative example

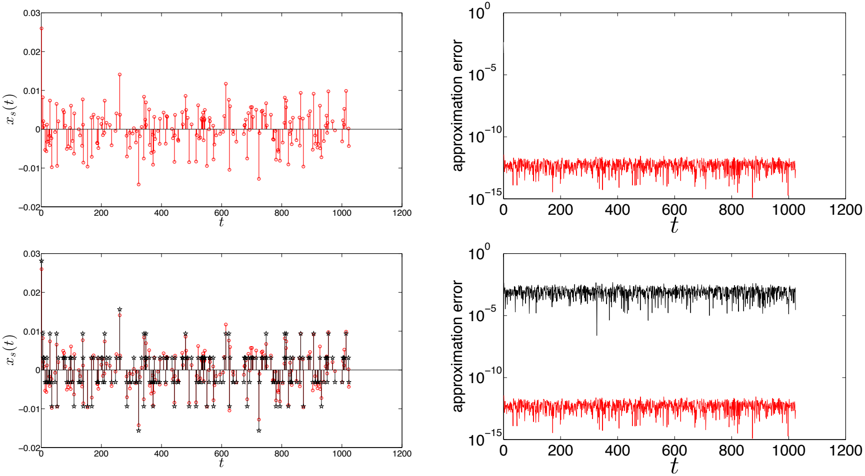
Next, quantize the compressive samples:





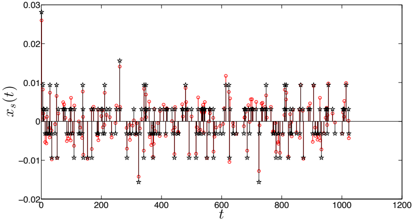
An illustrative example

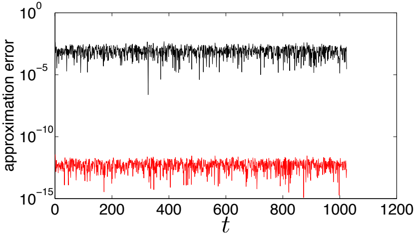
Next, quantize the compressive samples:



An illustrative example

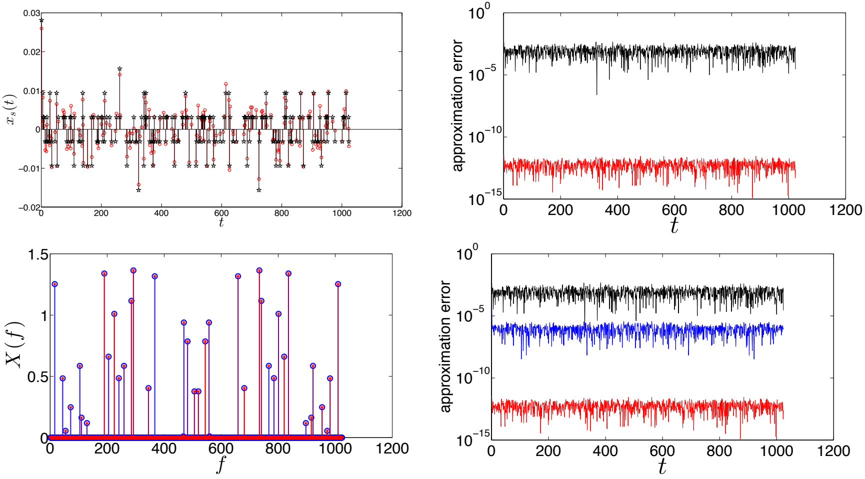
Compare with direct quantization using the same bit-budget.





An illustrative example

Compare with direct quantization using the same bit-budget.



An illustrative example

In the above example:

- The *signal* $x \in \mathbb{R}^N$ with $N = 1024$.
- The *sparsity basis*: discrete Fourier basis. Specifically, $X = \text{DFT}(x)$ is $k = 40$ sparse.
- *Sampling scheme*: uniformly random subsampling: collect $m = 200$ samples.
- With no quantization, we can recover x perfectly: error $\sim 10^{-13}$.
- With a total bit budget $R = 600$
    - optimal quantization: max error $\sim 1.6 \times 10^{-4}$
    - quantizing compressive samples: max error $\sim 0.46 \times 10^{-2}$
- "Loss in bit depth"$= \log_2(\text{max err CS}/\text{max err opt}) \approx 4.7$ bits

Perspective

The above discussion highlights a number of important issues: Let's rephrase what we have observed:

- In the optimal case, we used $b_1 = (R - \log_2 \binom{N}{k})/k$ bits to round-off each non-zero entry (and use the symmetry of the Fourier coefficients). The resulting ℓ_∞ approximation error was 2^{-b_1}. When we plug in numbers, $b_1 \approx 12.5$.
- Using the same bit budget R, in the CS case, we get an accuracy of 2^{-b_2} with $b_2 \approx 7.8$.
- In the CS case, to get an approximation with "12.5-bit accuracy", we would need to use 5 additional bits per measurement.

Perspective

The above discussion highlights a number of important issues: Let's rephrase what we have observed:

- In the optimal case, we used $b_1 = (R - \log_2 \binom{N}{k})/k$ bits to round-off each non-zero entry (and use the symmetry of the Fourier coefficients). The resulting ℓ_∞ approximation error was 2^{-b_1}. When we plug in numbers, $b_1 \approx 12.5$.
- Using the same bit budget R, in the CS case, we get an accuracy of 2^{-b_2} with $b_2 \approx 7.8$.
- In the CS case, to get an approximation with "12.5-bit accuracy", we would need to use 5 additional bits per measurement.
- In the optimal case, the total bit budget is 600 whereas to get the same accuracy *in the CS case, we need a bit budget of 1600* – not compressed.

Perspective

The above discussion highlights a number of important issues: Let's rephrase what we have observed:

- In the optimal case, we used $b_1 = (R - \log_2 \binom{N}{k})/k$ bits to round-off each non-zero entry (and use the symmetry of the Fourier coefficients). The resulting ℓ_∞ approximation error was 2^{-b_1}. When we plug in numbers, $b_1 \approx 12.5$.
- Using the same bit budget R, in the CS case, we get an accuracy of 2^{-b_2} with $b_2 \approx 7.8$.
- In the CS case, to get an approximation with "12.5-bit accuracy", we would need to use 5 additional bits per measurement.
- In the optimal case, the total bit budget is 600 whereas to get the same accuracy *in the CS case, we need a bit budget of 1600* – not compressed.
- More importantly, if we want to have a high-accuracy approximation using CS, say 24 bits, in the above example we need to quantize each CS measurement using a uniform scalar quantizer with a depth of 29 bits! This bit depth overhead is given by

$$b_{\text{MSQ}} = b_{\text{opt}} + \left\lceil \log_2 C\sqrt{k} \right\rceil$$

Observations and issues

Rounding off CS measurements gives exponential accuracy in terms of the bit budget:

$$\|x - \tilde{x}_{\mathrm{MSQ}}\|_2 \leq Ck2^{-R/(C_1 k \log(N/k))}.$$

However, this is not very useful:

- Overhead in terms of the bit budget: roughly, $R_{\mathrm{CS}} \approx \frac{m}{k} R_{\mathrm{opt}}$.
- **Problem:** Source coding is combined with the A/D conversion:
  - For a b-bit quantization of each measurement, a quantizer with bit depth of b-bits must be implemented on hardware.

Observations and issues

Rounding off CS measurements gives exponential accuracy in terms of the bit budget:

$$\|x - \tilde{x}_{\mathrm{MSQ}}\|_2 \leq Ck2^{-R/(C_1 k \log(N/k))}.$$

However, this is not very useful:

- Overhead in terms of the bit budget: roughly, $R_{\mathrm{CS}} \approx \frac{m}{k} R_{\mathrm{opt}}$.
- **Problem:** Source coding is combined with the A/D conversion:
  - For a b-bit quantization of each measurement, a quantizer with bit depth of b-bits must be implemented on hardware.
  - There is a physical limit to how small this step size can be (approximately b is 20-21 bits). This limits the best accuracy one can obtain in the CS setting.

Observations and issues

Rounding off CS measurements gives exponential accuracy in terms of the bit budget:

$$\|x - \tilde{x}_{\mathrm{MSQ}}\|_2 \leq Ck2^{-R/(C_1 k \log(N/k))}.$$

However, this is not very useful:

- Overhead in terms of the bit budget: roughly, $R_{\mathrm{CS}} \approx \frac{m}{k} R_{\mathrm{opt}}$.
- **Problem:** Source coding is combined with the A/D conversion:
  - For a b-bit quantization of each measurement, a quantizer with bit depth of b-bits must be implemented on hardware.
  - There is a physical limit to how small this step size can be (approximately b is 20-21 bits). This limits the best accuracy one can obtain in the CS setting.
  - Including the overhead $\log_2 C\sqrt{k} \approx 8$ for a 1000-sparse signal, the limit on accuracy is 12-13 bits.

Coarse quantization

Main Problem: Increasing the bit depth of a quantizer indefinitely is not possible: there are physical constraints that make this expensive and after a point impossible. Need to devise quantization schemes such that

- they can be implemented using low bit depth scalar quantizers, i.e., scalar quantizers with a relatively large step size δ. Such quantizers are cheap and require low power.
- they yield approximations to the original signal with error much smaller than $O(\delta)$ (for example, by increasing the number of measurements),
- they yield compressed representations after very light computation (again low battery and low storage)

Such quantization schemes are called coarse quantization schemes.

$\Sigma\Delta$ quantization

Our coarse quantization method will be rth-order $\Sigma\Delta$ quantization: Let $y = \Phi x$ where $\Phi \in \mathbb{R}^{m \times N}$ is a frame or a compressive sampling matrix.

$$(\Delta^r u)_j = y_j - q_j.$$

Here $q_j \in \mathcal{A}_L^\delta$ are chosen such that $\|u\|_\infty \lesssim d$ – **stable** r**th-order** $\Sigma\Delta$ **scheme** In this case:

$$y - q_{\Sigma\Delta} = D^r u, \quad \text{with } \|u\|_\infty \lesssim d$$

where $$D = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{m \times m}.$$

Coarse quantization – classical setup

Coarse quantization in the classical setup. Given a fixed δ, as large as $\delta = 2$ in the 1-bit case, i.e., when $\mathcal{A} = \{\pm 1\}$, increase the number of samples and exploit redundancy.
Here "samples" are typically basis or frame coefficients, and $\lambda > 1$ is the *oversampling factor*.

Coarse quantization – classical setup

Coarse quantization in the classical setup. Given a fixed δ, as large as $\delta = 2$ in the 1-bit case, i.e., when $\mathcal{A} = \{\pm 1\}$, increase the number of samples and exploit redundancy.
Here "samples" are typically basis or frame coefficients, and $\lambda > 1$ is the *oversampling factor*.

- MSQ yields approximation error $\sim \lambda^{-1/2}$ (under white noise assumption).

Coarse quantization – classical setup

Coarse quantization in the classical setup. Given a fixed δ, as large as $\delta = 2$ in the 1-bit case, i.e., when $\mathcal{A} = \{\pm 1\}$, increase the number of samples and exploit redundancy.
Here "samples" are typically basis or frame coefficients, and $\lambda > 1$ is the *oversampling factor*.

- MSQ yields approximation error $\sim \lambda^{-1/2}$ (under white noise assumption).
- $\Sigma\Delta$ schemes exploit redundancy more efficiently: an rth order $\Sigma\Delta$ quantizer yields approximations with error:
    - $O(\lambda^{-r})$ in the bandlimited setting (Daubechies-DeVore, Güntürk)
    - $O(\lambda^{-r})$ in the finite frame setting: Blum-Lammers-Powell-Y ("smooth" frames), Güntürk-Lammers-Powell-Saab-Y (Gaussian random frames), Krahmer-Saab-Y (sub-Gaussian random frames).

$\Sigma\Delta$ quantization – error vs. bits used

In the formulas above, $\lambda \sim R$ where R is total number of bits used (per Nyquist interval in the bandlimited setting). That is, using an rth order $\Sigma\Delta$ quantizer, we get an approximation with error

$$\|x - \tilde{x}_r\| \leq C_r R^{-r}.$$

This is significantly inferior compared to the optimal error: $O(2^{-cR})$. However, when we optimize the order r of the $\Sigma\Delta$ scheme:

- approx. error $\sim 2^{-0.1R}$ in the bandlimited setting (Deift- Krahmer-Güntürk)
- approx. error $\sim 2^{-C\sqrt{R}}$ in the finite frame setting (Krahmer, Saab, Ward)

A recent result by Iwen and Saab: Exponential accuracy *without optimizing the order*, but instead *compressing the resulting quantized values further*.

Coarse quantization – compressed sensing

What do we know? Set $R \sim m/k$.

- MSQ: best one can hope for is $O((R)^{-1})$ (GLPSY – follows from a theorem of Goyal-Vetterli-Thao on frame quantization).

Coarse quantization – compressed sensing

What do we know? Set $R \sim m/k$.

- MSQ: best one can hope for is $O((R)^{-1})$ (GLPSY – follows from a theorem of Goyal-Vetterli-Thao on frame quantization).
- $\Sigma\Delta$: Using an rth-order $\Sigma\Delta$ scheme to quantize compressive samples of x
  - Φ is Gaussian or sub-Gaussian: distortion $\sim R^{-\alpha(r-1/2)}$ via a two-stage scheme: (i) recover the support, (ii) refine using Sobolev duals. (GLPSY, 2013), (Krahmer-Saab-Y, 2014)

Coarse quantization – compressed sensing

What do we know? Set $R \sim m/k$.

- MSQ: best one can hope for is $O((R)^{-1})$ (GLPSY – follows from a theorem of Goyal-Vetterli-Thao on frame quantization).
- $\Sigma\Delta$: Using an rth-order $\Sigma\Delta$ scheme to quantize compressive samples of x
  - Φ is Gaussian or sub-Gaussian: distortion $\sim R^{-\alpha(r-1/2)}$ via a two-stage scheme: (i) recover the support, (ii) refine using Sobolev duals. (GLPSY, 2013), (Krahmer-Saab-Y, 2014)
- Two main disadvantages of $\Sigma\Delta$ with the two-stage scheme
  - Smallest non-zero entry of x must be $\geq C_r\delta$. This essentially rules out low-bit schemes, e.g., 1 or 2 bits per sample. Also, the set of allowed signals depends on the order r.
  - Not robust to noise and not stable with compressible signals.

$\Sigma\Delta$ for CS: A one-stage reconstruction method

Original signal: $x \in \mathbb{R}^N$ sparse or compressible.

Compressive samples: $y = \Phi x + w$, $\|w\|_\infty \leq \epsilon$

After A/D conversion: $q := Q^r_{\Sigma\Delta}(y)$ where $Q^r_{\Sigma\Delta}$ is a stable rth-order $\Sigma\Delta$ quantizer with step size δ. Then

$$\Phi x - q = D^r u, \text{ with } \|u\|_\infty \leq C(r)\delta$$

where D is bidiagonal with $D(i,i) = 1$ and $D(i+1,i) = -1$, $i = 1, \ldots, m$.

Proposed one-stage reconstruction algorithm (R. Saab, R. Wang,Y)

$$(\hat{x}, \hat{v}) := \arg\min_{(z,v)} \|z\|_1 \quad \textit{subject to} \ \|D^{-r}(\Phi z + v - q)\|_2 \leq c(r)\sqrt{m}$$

$$\textit{and} \quad \|v\|_2 \leq \epsilon\sqrt{m},$$

$\Sigma\Delta$ for CS: A one-stage reconstruction method

In the setting described above:

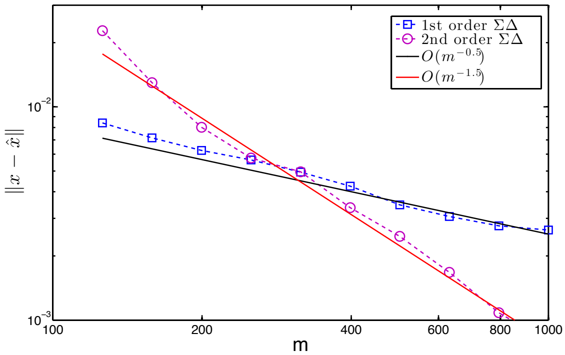
- $x \in \mathbb{R}^N$, $y = \Phi x + w$ with $\|w\|_\infty \leq \epsilon$
- $q = Q^r_{\Sigma\Delta}(y)$
- $\hat{x}$ is obtained using the one-stage algorithm.

Theorem (Saab-Wang-Y, 2014)

If Φ belongs to a wide class of matrices (that include sub-Gaussian random matrices whp) with $m \geq m_{\min} = C_0 k \log(N/k)$, we have

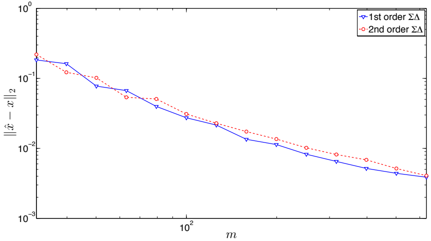
$$\|x - \hat{x}\|_2 \leq C_1(r)\left(\frac{m}{k}\right)^{-a(r-1/2)} + C_2\frac{\sigma_k(x)}{\sqrt{k}} + C_3\sqrt{\frac{m}{m_{\min}}}\epsilon.$$

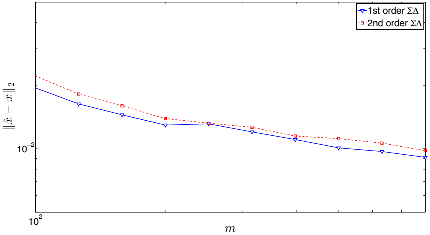
One stage reconstruction – numerical experiments



$N = 1000$, $k = 10$, $\delta = 0.01$, m varies between 100 and 100, $\Phi_{ij} \sim \mathcal{N}(0, 1)$, worst case error among 80 independent trials.

One stage reconstruction – numerical experiments





Left: Compressible x with $x[j] \sim j^{-2}$.
Right: Noisy measurements: $y = \Phi x + e$ with $\|e\|_\infty = 0.001$, x is 10-sparse with standard Gaussian non-zero entries.

In both cases: $N = 1000$, m varies between 100 and 1000, $\Phi_{ij} \sim \mathcal{N}(0,1)$, $\delta = 0.01$, and we show the worst case error among 10 independent trials.

One stage reconstruction: is this good enough?

ΣΔ quantization and one stage reconstruction algorithm:

- Utilizes "redundancy": more measurements gives better reconstruction with the same scalar quantizer.
- When we optimize the order r, we get root exponential accuracy with respect to the number of measurements, equivalently, bit budget.
- Stable and robust!

One stage reconstruction: is this good enough?

$\Sigma\Delta$ quantization and one stage reconstruction algorithm:

- Utilizes "redundancy": more measurements gives better reconstruction with the same scalar quantizer.
- When we optimize the order r, we get root exponential accuracy with respect to the number of measurements, equivalently, bit budget.
- Stable and robust!

However: In the above examples:

- We use a quantizer with $\delta = 0.01$ to quantize measurements in $[-10, 10]$.
- This gives a bit depth of 11 bits per measurement.
- Approximation error approximately 2^{-9} with a total bit budget of 7000 bits.
- Compare this with less than 200 bits in the optimal case, and about 800 bits if we can quantize as finely as we want.

Compressing the $\Sigma\Delta$ bit stream

Question: Can we compress the resulting $\Sigma\Delta$ quantized measurements even further?

Answer: Yes! With a scalar quantizer of bit-depth 5 (5 bits per measurement), we can get the same accuracy as above with a bit budget of less than 4500 bits.

Compressing the $\Sigma\Delta$ bit stream

Question: Can we compress the resulting $\Sigma\Delta$ quantized measurements even further?

Answer: Yes! With a scalar quantizer of bit-depth 5 (5 bits per measurement), we can get the same accuracy as above with a bit budget of less than 4500 bits.

$$x \in \mathbb{R}^N \xrightarrow{\Phi} y \in \mathbb{R}^m \xrightarrow{Q} q_{\Sigma\Delta} \in \mathcal{A}^m \xrightarrow{\mathcal{E}} \tilde{q} \xrightarrow{\Delta_{Q,\mathcal{E}}} x^{\#} \in \mathbb{R}^N$$

Color code: Acquisition (CS), A/D conversion ($\Sigma\Delta$), Compression, Decoding

- Acquisition: CS with Bernoulli Φ
- A/D conversion: $\Sigma\Delta$ quantization of order r.
- **Focus** on designing $\mathcal{E}$ (compression)and $\Delta_{Q,\mathcal{E}}$ (decoding).

New scheme – compression

Let $x \in \mathbb{R}^N$, $\Phi \in \mathbb{R}^{m \times N}$, $y = \Phi x$. Fix an alphabet $\mathcal{A}$ and let $q \in \mathcal{A}^m$ be the rth-order $\Sigma\Delta$ quantization y.

Compression: Based on a recent construction by Iwen and Saab in the setting of frames. Encode $\Sigma\Delta$ quantized mesurements $q \in \mathcal{A}^m$ using the map

$$\mathcal{E} : q \mapsto BD^{-r}q =: \tilde{q}$$

where B is an $L \times m$ Bernoulli matrix with $L = m_{\min} \sim k \log(N/k)$.

Note that:

- Assign binary labels to the entries of $\tilde{q}$: will need $m^{r+1}|\mathcal{A}|$ such labels.
- Accordingly: need $R = L\big((r+1)\log_2(m)| + \log|\mathcal{A}|\big)$ bits to represent $\tilde{q}$.
- If we keep L fixed as m increases—and distortion decreases as $O(m^{-r})$—this will give us exponential accuracy if we can decode.

New scheme – decoding

Finally, we need a decoder $\Delta_{Q,\mathcal{E}} : \tilde{q} \mapsto x^{\#}$ such that $\|x^{\#} - x\|$ is small.

Decoder: Based on a modification of the one-stage recovery algorithm. Specifically:

- Recover $\tilde{q} = BD^{-r} q_{\Sigma\Delta}$ from the binary labels,
- Obtain $x^{\#}$ by solving

$$x^{\#} := \arg\min \|z\|_1 \text{ subject to } \|BD^{-r}\Phi z - \tilde{q}\|_2 \leq C(r)\delta\sqrt{mL}.$$

Note that the size of this optimization problem is significantly smaller than the one without compression. For example, $m = 1000, L = 100$.

New scheme – exponential accuracy

Theorem. (Wang, Saab, Y, 2014)

Let $\Phi \in \mathbb{R}^{m\times N}$ and $B \in \mathbb{R}^{L\times m}$ be Bernoulli matrices where $L = c_3 k \log(N/k)$ and $m \geq L$ for some $k < m$. Then with high probability the following holds for all k-sparse $x \in \mathbb{R}^N$ with $\|x\|_2 \leq \frac{1}{3\sqrt{k}}$.

Denote by $q = Q^r_{\Sigma\Delta}(\Phi x)$ with alphabet $\mathcal{A}$ and $r \geq 2$. Let $\mathcal{E}(q) = BD^{-r}q$ be the encoding of q. Then
(i) $\mathcal{E}(q)$ can be represented by $R = L(2\log m + \log|\mathcal{A}| + 1)$ bits,
(ii) Approximating x by the decoder above yields $x^{\#}$ which satisfies

$$\|\hat{x} - x\|_2 \leq c 2^{-\frac{R}{L}\frac{r-3/2}{2(r+1)}} =: \mathcal{D}$$

where c is a constant that depends on L and the $\Sigma\Delta$ scheme.

Remarks

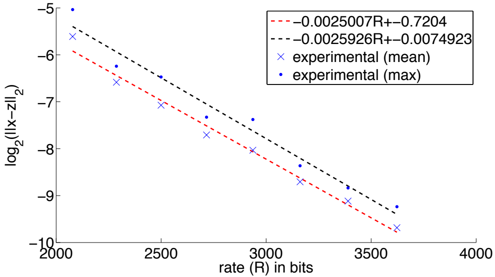
Note that:

- Above, we keep L and the quantization alphabet $\mathcal{A}$ for the $\Sigma\Delta$ quantizer fixed, i.e., we have a coarse quantization scheme.
- We increase the bit-budget, i.e., R, by increasing the number of measurements m.
- Surprisingly, we can utilize these additional bits (almost) as efficiently as if we are increasing the bit depth, i.e., refining $\mathcal{A}$.
- This scheme gives us exponential accuracy with $\Sigma\Delta$ schemes of any fixed order r, i.e., no need for optimizing the order for the given bit budget R.
- Decoding from the quantized and compressed measurements is done using a convex optimization algorithm.
- The results are valid for 1-bit $\Sigma\Delta$ quantized compressed sensing as well.

Numerical experiments

Let: $N = 1024$, $k = 10$, $\Phi \in \mathbb{R}^{m \times N}$ Bernoulli, $m \in \{100 \cdot 2^p : \ p = 0, \ldots, 7\}$

First: distortion vs. bit budget when we use the new scheme with $\Sigma\Delta$ schemes of order $r = 1$ with $\delta = 0.01$ (what we had before):



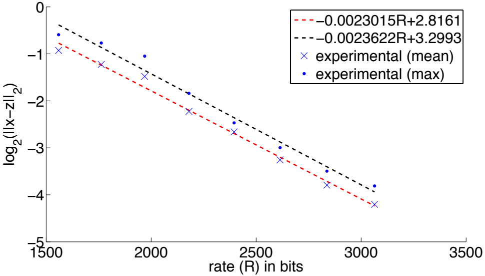
Note: We get the same accuracy level ($\sim 2^{-9}$) with less than 3500 bits instead of 7000 bits of "pre-compression". How about larger δ, i.e., coarser quantization?

Numerical experiments

Again: $N = 1024$, $k = 10$, $\Phi \in \mathbb{R}^{m \times N}$ Bernoulli, $m \in \{100 \cdot 2^p : \ p = 0, \ldots, 7\}$.

Next: distortion vs. bit budget when we use the new scheme with $\Sigma\Delta$ schemes with $\delta = 0.5$!

$\Sigma\Delta$ quantizer of order $r = 1$

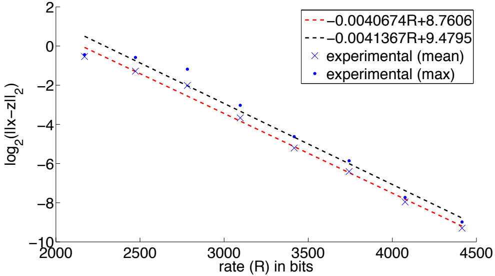


Numerical experiments

Again: $N = 1024$, $k = 10$, $\Phi \in \mathbb{R}^{m\times N}$ Bernoulli, $m \in \{100 \cdot 2^p : \ p = 0, \dots, 7\}$.

Next: distortion vs. bit budget when we use the new scheme with $\Sigma\Delta$ schemes with $\delta = 0.5$!

$\Sigma\Delta$ quantizer of order $r = 2$

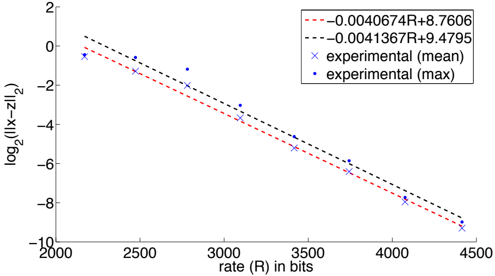


Numerical experiments

Again: $N = 1024$, $k = 10$, $\Phi \in \mathbb{R}^{m \times N}$ Bernoulli, $m \in \{100 \cdot 2^p : \ p = 0, \ldots, 7\}$.

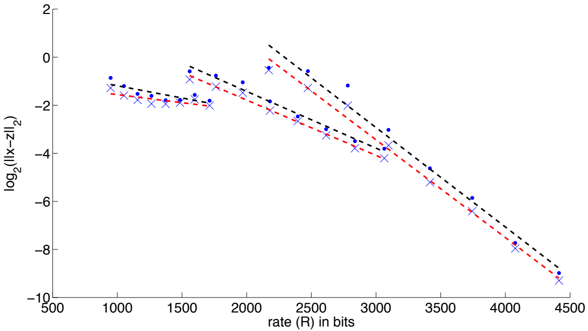
Next: distortion vs. bit budget when we use the new scheme with $\Sigma\Delta$ schemes with $\delta = 0.5$!

$\Sigma\Delta$ quantizer of order $r = 2$



Note: We still get the same accuracy level ($\sim 2^{-9}$) with approximately 4500 bits instead of 7000 bits of "pre-compression", this time with very coarse quantization (scalar quantizer with depth of 3.5 bits).

Numerical experiments – big picture



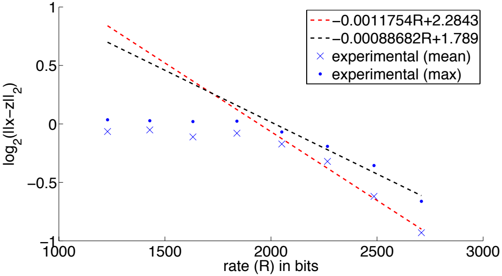
Left: MSQ
Middle: $\Sigma\Delta$ with $r = 1$
Right: $\Sigma\Delta$ with $r = 2$

One-bit compressed sensing

Here: $N = 4000$, $k = 10$, $\Phi \in \mathbb{R}^{m\times N}$, $m \in \{100 \cdot 2^p : \ p = 0, \ldots, 7\}$.

Distortion vs. bit budget when we use the new scheme with one-bit $\Sigma\Delta$ schemes with $\delta = 6$!

$\Sigma\Delta$ quantizer of order $r = 1$

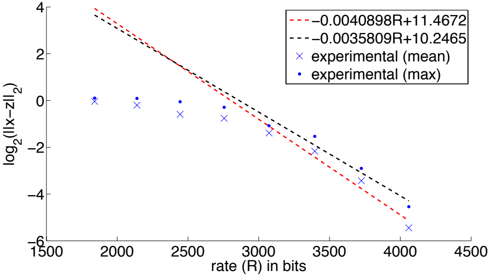


One-bit compressed sensing

Here: $N = 4000$, $k = 10$, $\Phi \in \mathbb{R}^{m \times N}$, $m \in \{100 \cdot 2^p : \; p = 0, \dots, 7\}$.

Distortion vs. bit budget when we use the new scheme with one-bit $\Sigma\Delta$ schemes with $\delta = 6$!

$\Sigma\Delta$ quantizer of order $r = 2$

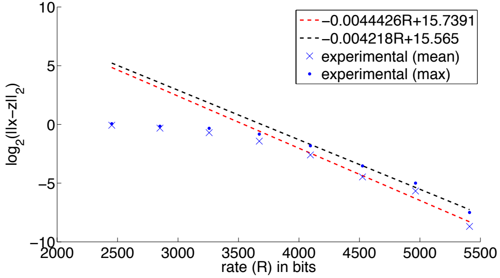


One-bit compressed sensing

Here: $N = 4000$, $k = 10$, $\Phi \in \mathbb{R}^{m \times N}$, $m \in \{100 \cdot 2^p : \; p = 0, \ldots, 7\}$.

Distortion vs. bit budget when we use the new scheme with one-bit $\Sigma\Delta$ schemes with $\delta = 6$!

$\Sigma\Delta$ quantizer of order $r = 3$



Concluding remarks

- Efficient quantization for compressed sensing is crucial.
- Fine quantization schemes have physical limitations which in turn limit the best accuracy one can obtain.
- Coarse quantization schemes, such as $\Sigma\Delta$ quantization, provide a remedy.
- We introduce a novel CS quantization/compression/recovery scheme that achieves exponential accuracy with respect to bit budget while using a fixed, coarse quantization alphabet.
- Done? Our "bit counting" method is rudimentary: it does not incorporate the structure of the quantized values that are generated by $\Sigma\Delta$ schemes. It might be possible to compress even further.