TD #7: Compressed sensing

Large-scale Mathematical Programming

Leo Liberti, CNRS LIX Ecole Polytechnique
liberti@lix.polytechnique.fr

INF580

Summary

- **Costly channels**: construct sparse signals, encoding matrices, encode into sample, decode, verify result precision
- **Noisy channels**: construct words, encode them into longer signals, send them on a noisy channel, receive signal + error, reconstruct the error by solving a basis pursuit LP, retrieve the corresponding word, verify result precision
- **Robustness of compressed sensing**: try above with almost sparse signals

Costly channels

Use AMPL or Python

- **Input:** m, n, signal density $\beta \in (0, 1)$
- **Encoding matrix:** $A \sim \mathsf{N}(0, 1)^{mn}$
- **Original signal:** $\hat{x} \sim \mathsf{U}(-1, 1)$ with density β
- **Encode signal into sample:** $b = A\hat{x}$
- **Decode sample to signal:** $x^* = \arg\min\{\|x\|_1 \mid Ax = b\}$
- **Evaluate result:** compare $|\mathrm{supp}(x^*)|$, $|\mathrm{supp}(\hat{x})|$; compute $\|x^* - \hat{x}\|_p$ for $p \in \{1, 2\}$
- **Task:** Is it better to sample A from normal or uniform?

Noisy channels

Use Python

- **Input:** sentence, noise $\Delta \in (0, 1)$, redundancy R
- **Sentence to vector:** turn the sentence to a vector in $\{0, 1\}^n$
- **Encoding matrices:** find A, Q s.t. $\dim \text{Im}(A^\top | Q) = n$ and $A\,Q = 0$
- **Sample noise vector:** $\hat{x} \in \mathbb{R}^n$ with density Δ
- **Compute noisy message**
- **Retrieve noise vector:** sove a basis pursuit LP
- **Retrieve sentence:** vector back to string
- **Compare** sent and retrieved string
- **Tasks**: normal or uniform encoding? do random projections help?