Final project: Compressed sensing decoder

D. Ustinova, A. Smeshko, E. Avdotin

Skolkovo Institute of Science and Technology

NLA 2020 final project defence

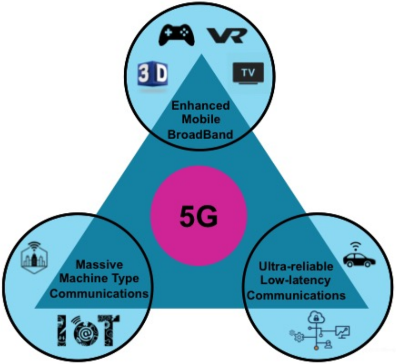
December 17, 2020

Moscow, Russia

Outline

Outline

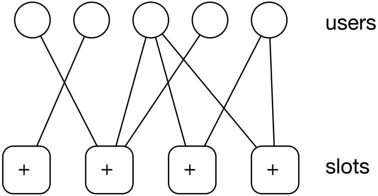
Problem Relevance



- Massive machine-type communication: 10^6devices/km^2
- $T-$fold ALOHA
- Minimize energy consumption
- Decoding of noisy sum in time slots is considered
- Compressed sensing decoder is used

Outline

System model



- K_a active users, V time slots
- B-bit messages corresponds to the set of active users as $\mathbf{W} = \{w_1, ..., w_{K_a}\}$
- Users encode the binary message w_i into a codeword of length n, and apply BPSK modulation ($x_{w_i} \in \{1, -1\}$)

System model

Denote the $\mathcal{N}_j$ set of users, who transmit to the $j-$th slot, $j = 1, \ldots, V$. Then in each slot we'll receive:

$$\mathbf{y}_j = \sum_{i \in \mathcal{N}_j} \mathbf{x}_{w_i} + \mathbf{z}_j,$$

where $\mathbf{x}_{w_j}$ is BPSK modulated codeword transmitted by the i-th user and $\mathbf{z}_j \sim \mathcal{N}(\mathbf{0}, \sigma^2)$ is an additive white Gaussian noise (AWGN). Thus, in slot we receive the sum of $|\mathcal{N}_j| = T$ messages and want to determine each x_{w_i}. We can represent this sum as a multiplication of matrix A and sparse vector b_j:

$$\mathbf{y}_j = A\mathbf{b}_j + \mathbf{z}_j,$$

where we consider matrix $A_{n \times m}$ - matrix with all possible modulated codewords as columns, $\mathbf{b}_j$ - activity vector with exactly T ones (on transmitted codewords places) and other zeros.

System model

Consequently, our decoding task can be reduced to searching the right activity vector (among all such vectors with T ones):

$$\widehat{\mathbf{b}} = \arg\min_{\mathbf{b}_j} ||A\mathbf{b}_j - \mathbf{y}_j||$$

- If we consider $\|\cdot\|_1$, this could be interpreted as compressed sensing problem
- If we consider $\|\cdot\|_2$, this is non-negative least squares problem (NNLS), or bounded-variable least squares (BVLS), because we have rectrictions for vector $\mathbf{b}_j$
- If we apply QR decomposition, we can find $\widehat{\mathbf{b}}$ via pseudoinverse matrix

Codebook generation

- Generate all possible binary vectors of length k (we used k = 10) - these are our information vectors
- Encode them with Low-density parity-check code generator matrix $\rightarrow$ achieve a codebook $C_{35\times1024}$ (each information word of length 10 was encoded to a codeword of length 35)
- Modulate it and achieve sensing matrix A

Outline

NNLS

- *scipy.optimize.nnls* Constrained least squares problem, where the coefficients aren't allowed to become negative:

$$\arg\min_{\mathbf{x}\geq 0} \|A\mathbf{x} - \mathbf{y}\|_2$$

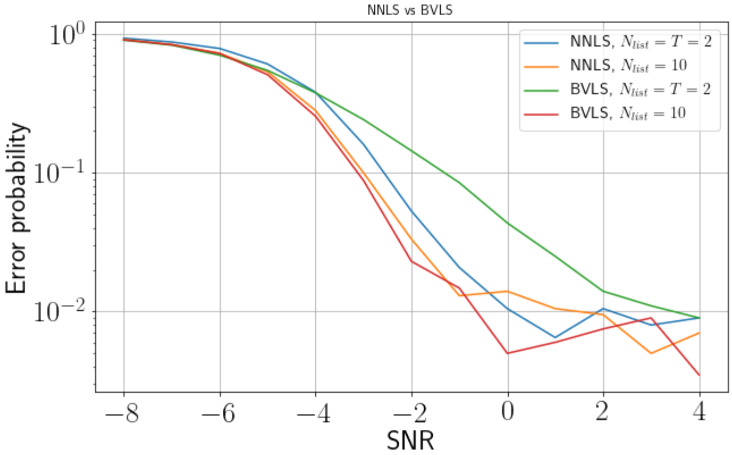
- The algorithm is an active set method. It solves the KKT (Karush-Kuhn-Tucker) conditions for the non-negative least squares problem
- Improvement: search not T maximal, but some number (e.g. $N_{list} = 10$). Then we consider all possible combinations $C_{N_{list}}^T$ and find the minimum distance in terms of $\|\cdot\|_2$

BVLS

We considered 2 approaches:

- $scipy.optimize.least\_squares$ solve a nonlinear least-squares problem with bounds on the variables. Given the residuals and the loss function finds a local minimum of the cost function
- $scipy.optimize.lsq\_linear$ solve a linear least-squares problem with bounds on the variables. Given a matrix A and a target vector $\mathbf{b}$ with m elements minimize $0.5 * ||Ax - b||^2$ subject to $lb <= x <= ub$

Comparison of BVLS and NNLS



QR and pseudoinverse

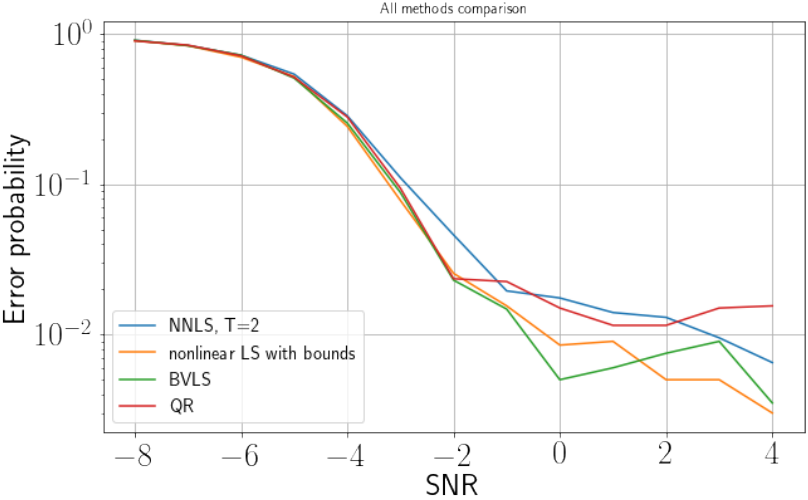
If we decompose the sensing matrix:

$$A = QR$$

Then our activity vector could be estimated as follows:

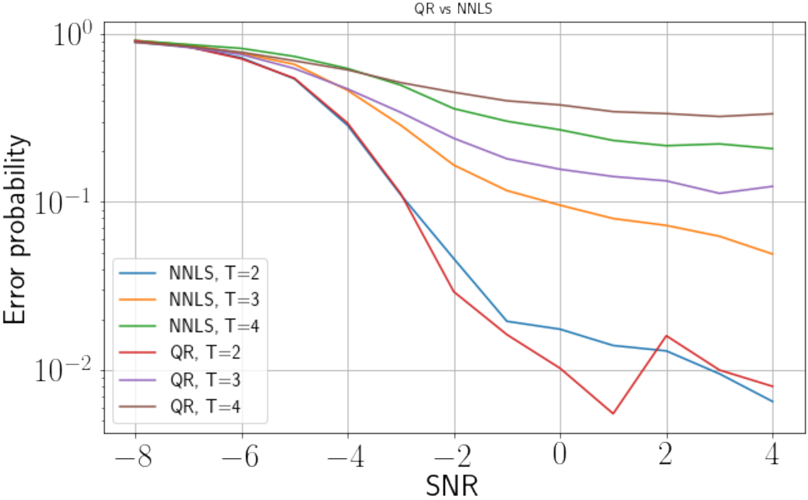
$$\widehat{\mathbf{b}} = A^{\dagger}\mathbf{y} = \left(A^{*}A\right)^{-1} A^{*}\mathbf{y} = \left(R^{*}Q^{*}QR\right)^{-1} R^{*}Q^{*}\mathbf{y} = R^{\dagger}Q^{*}\mathbf{y}$$

Comparison of all methods

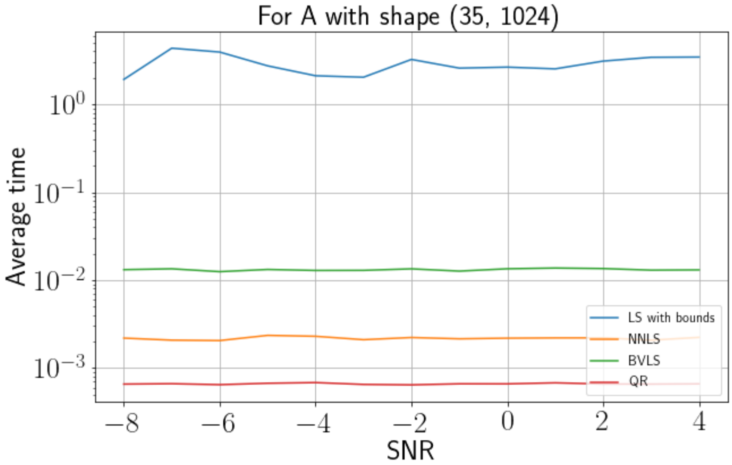


QR vs NNLS for different T

We can see, that for larger T NNLS outperforms QR



Execution time



Outline

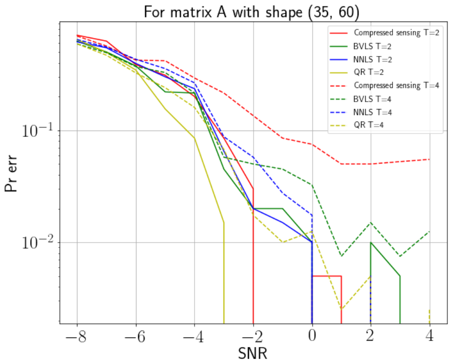
Another approach

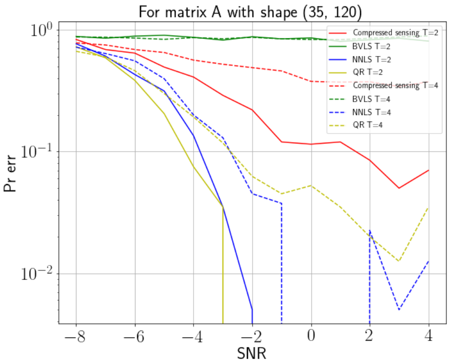
- Let us suppose we have a set of codewords to transmit less than 2^k
- Sensing matrix A has another properties, number of columns / length of codeword ≈ 2
- Compressed sensing is working solution

Compressed sensing and Fourier Transform

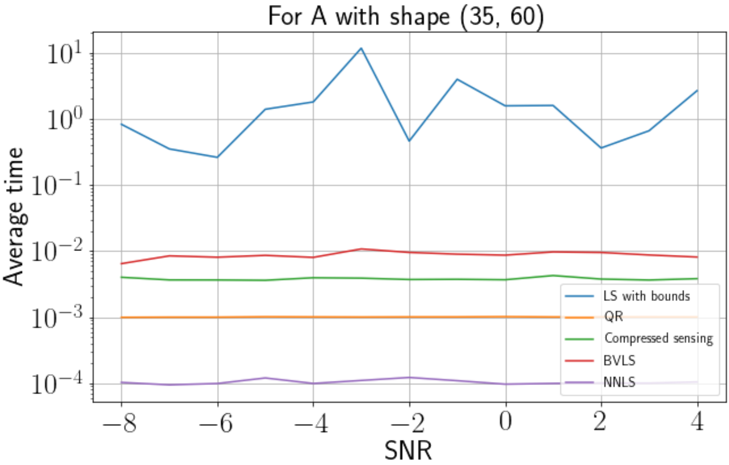
- Create the Fourier transform that will most minimize L1 norm
- Discrete Cosine Transform of sensing matrix dct(A)
- Use lasso to minimize the L1 norm
- Reconstruct coefficients
- Inverse Fourier transform to coefficients, take T with greatest absolute value

Methods comparison





Execution time



Outline

Summary

- In the original paper only NNLS was used
- We introduced additional optimization to the message set (N_{list})
- We have tried also CS and QR decomposition, different approaches of BVLS
- Comparison for different T was provided
- Execution time was compared
- BVLS has the best performance in terms of error probability

Thank you for your attention!