INTRODUCTION TO COMPRESSIVE SENSING

This script is an introduction to sparse signal representation and compressive sensing.

Contents


```
close all
clearvars
clc
```

Basis Representation Fundamentals

Every signal $x \in \mathbf{R}^N$ is representable in terms of N coefficients $\{s_i\}_{i=1}^N$ in a given basis $\{\psi_i\}_{i=1}^N$ for $\mathbf{R}^N$ as

$$x = \sum_{i=1}^{N} \psi_i s_i$$

Arranging the ψ_i as columns into the $N \times N$ matrix Ψ and the coefficients s_i into the $N \times 1$ coefficient vector $\mathbf{s}$, we can write that $x = \Psi s$, with $s \in \mathbf{R}$.

We say that signal x is K-sparse in the basis Ψ if there exists a vector $s \in \mathbf{R}^N$ with only $K \ll N$ nonzero entries such that $x = \Psi s$.

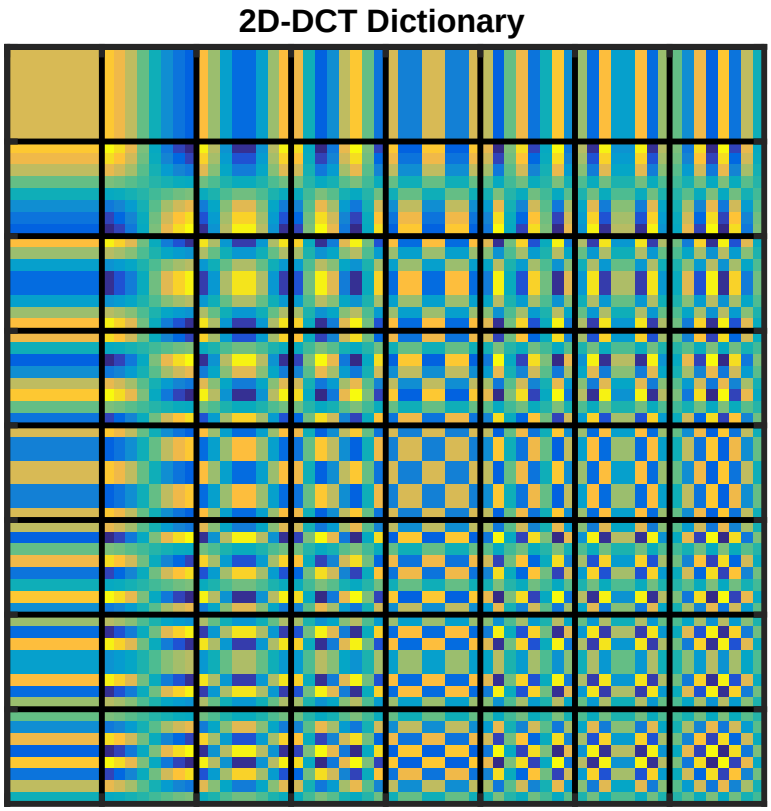
If we use basis matrix containing N unit-norm column vectors of length L where $L < N$ (i.e. $\Psi \in \mathbf{R}^{L \times N}$), then for any vector $x \in \mathbf{R}^L$ there exist infinitely many decompositions $s \in \mathbf{R}^N$ such that $x = \Psi s$. We refer to Ψ as the overcomplete sparsifying dictionary.

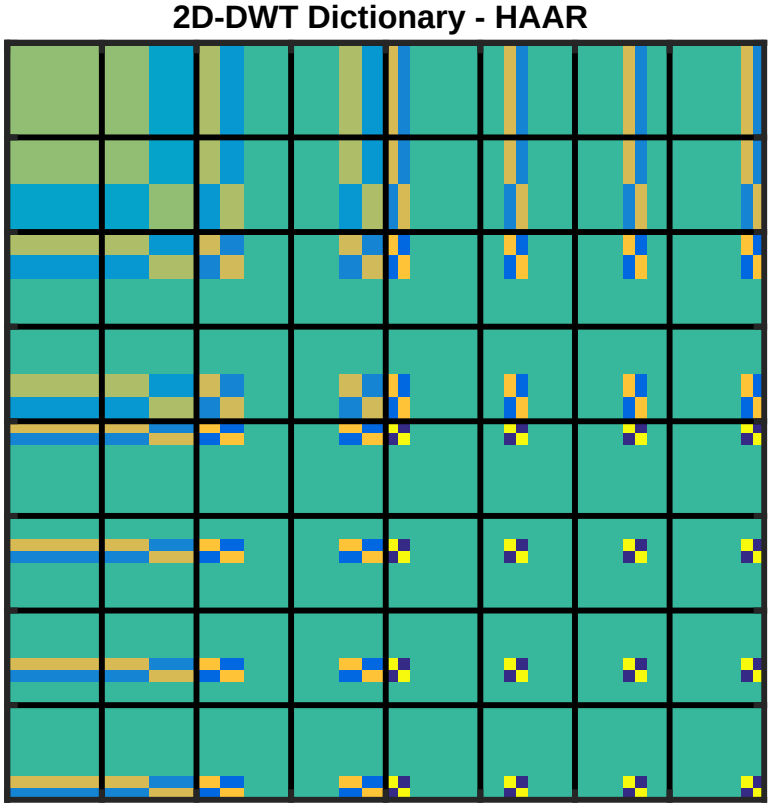
Sparsifying dictionary can be:

1. chosen from a known set of transforms like DCT, wavelet...
2. learnt from data

Example dictionary for 2D-DCT transformation

```
figure
visualizeDictionary(kron(dctmtx(8),dctmtx(8))')
title('2D-DCT Dictionary')
figure
haar1D = full(wmpdictionary(8, 'lstcpt', {{'haar',3}}));
visualizeDictionary(kron(haar1D, haar1D))
title('2D-DWT Dictionary - HAAR')
```





Uncertainty Principle

As an illustrative example, we will consider the case where our dictionary is the union of two particular orthobases: the identity (spike) basis and the Fourier (sine) basis $\Psi = [I \quad F]$.

Spike and Fourier basis are mutually fully incoherent in the sense that it takes n spikes to build up a single sinusoid and also it takes n sinusoids to build up a single spike.

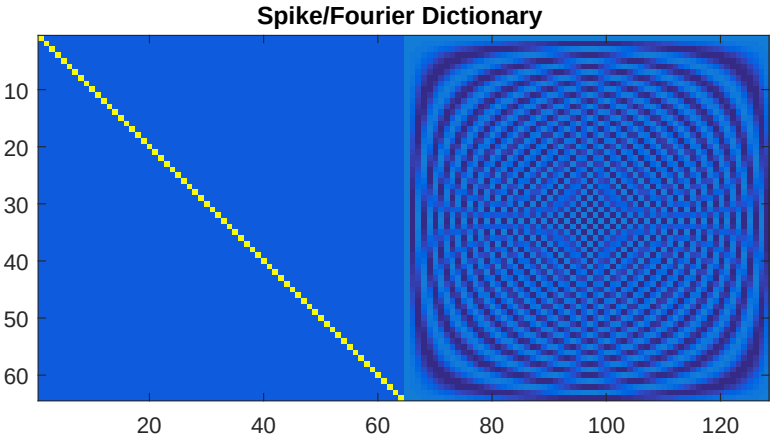
```
n = 64;

fourier = (1/sqrt(n))*exp((1j*2*pi*[0:n-1]'*[0:n-1])/n);
spike = eye(n);

psi = [spike, fourier];

% figure
imagesc(real(psi))
```

```
axis image
title('Spike/Fourier Dictionary')
```



Sines and spikes example

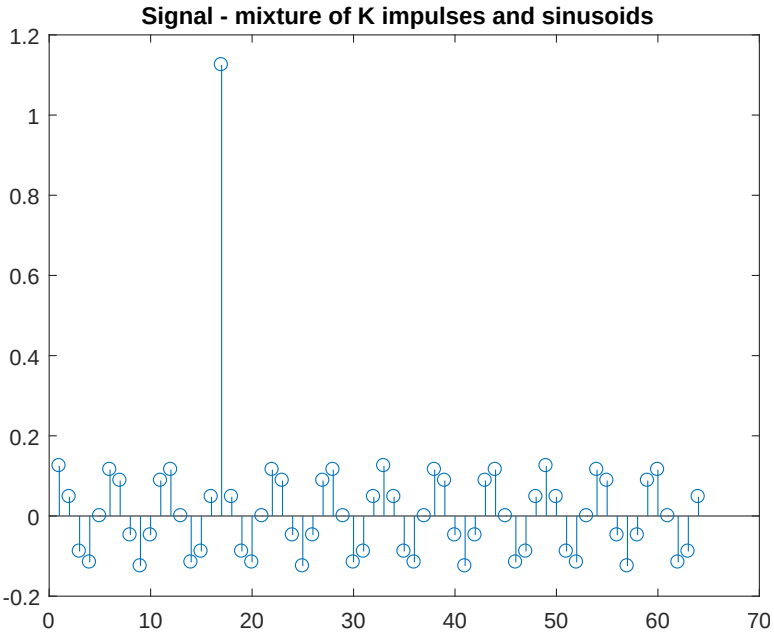
Now we will create a signal that is a mixture of spikes and sines. As we know that the first half of our matrix Ψ contains spike functions and the second half corresponds to sine functions, we can construct random sparsity pattern with sparsity K and compute $x = \Psi * s$ to obtain a signal which is a mixture of impulses and sinusoids.

```
%desired sparsity K
K = 2;
%randomly selected basis coefficients
idx = randi([1,2*n], 1, K);

%random sparsity vector
s = zeros(2*n, 1);
s(idx) = 1;

%obtain the signal which is a mixture of impulses and sinusoids
x = psi*s;

%visualize signal
% figure
stem(real(x))
title('Signal - mixture of K impulses and sinusoids')
```



Minimum Energy Decomposition - l_2 norm

There are infinite number of ways we can decompose signal x using atoms from our dictionary. Most natural way would be to use certain basis functions which correspond to previously selected basis function indices. This way we get the sparsest possible representation.

Another way we can get representation for x is by applying Ψ^* and by dividing the result by 2. Since $\Psi\Psi^* = 2I$, we get next reproducing formula for x.

$$x = \frac{1}{2}\Psi(\Psi^* x)$$

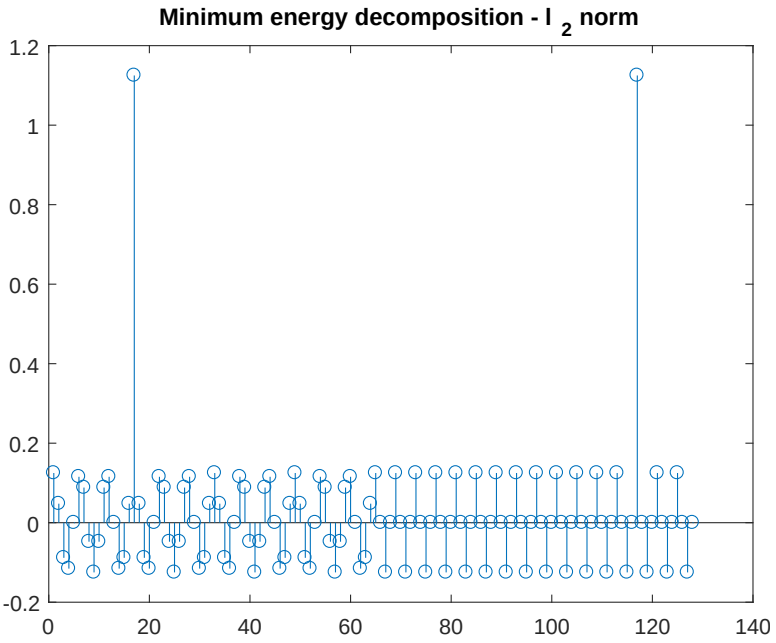
When we apply $s = \frac{1}{2}\Psi^* x$ we get result that corresponds to the minimum energy decomposition of our signal into a coefficient vector that represents x. Minimum energy solution corresponds to l_2 norm minimizaton. Unfortunately, minimum energy decomposition almost never yields the sparsest possible soultion. The reason for this is that the vector that a vector has minimum nenergy when its total energy is distribured over all the coefficients of the vector. l_2 gives us a solution that is dense, but has small values for the coefficients.

Our ability to separate sines part from the spikes part of our signal of interest is what will determine whether or nor we can find a unique sparse decomposition. Being able to tell these two components apart comes from a new kind of

uncertainty principle which states that a signal can't be too sparse in time and frequency domain simoultaneously.

```
%minimum energy decomposition
s = psi'*x;

% figure
stem(real(s))
title('Minimum energy decomposition - l_2 norm')
```



Sparse decomposition

Since our goal of finding sparsest possible representation of our signal x over some basis Ψ is equivalent to finding the solution with the smallest number of nonzero elements in basis coefficient vector s we will use l_0 pseudo-norm to find our solution. Sparse signal recovery can be formulated as finding minimum-cardinality solution to a constrained optimization problem. In the noiseless case, our constraint is simply $x = \Psi s$, while in the noisy case(assuming Gaussian noise), the solution must satisfy $\|x - x^*\|_2 \leq \epsilon$ where $x^* = \Psi s$ is the hypothetical noiseless representation and the actual representation is ϵ-close to it in l_2 norm. The objective function is the cardinality of s(number of nonzeros) which is often denoted $\|x\|_0$ and called l_0 norm of s.

Optimization problems corresponding to noiseless and noisy sparse signal recovery can be written as:

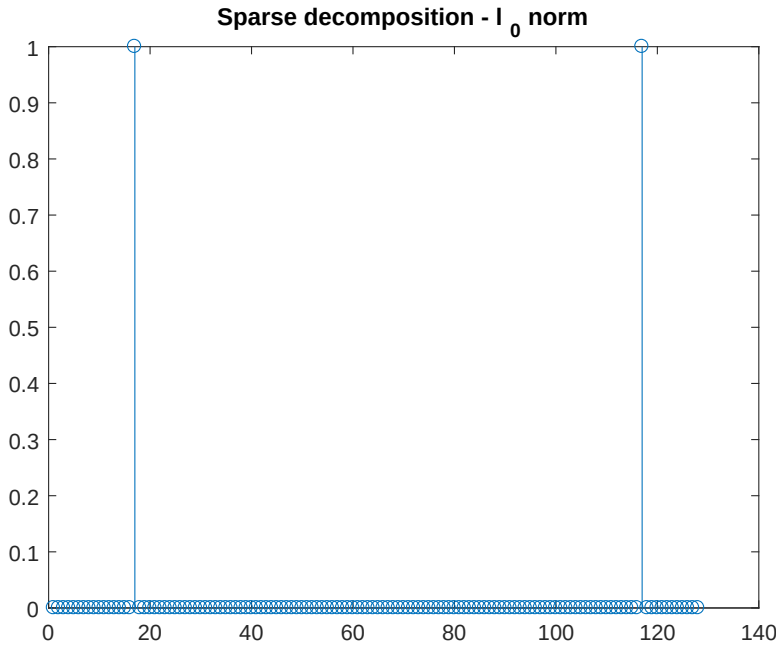
- $\min_x \|x\|_0 \quad s.t. \quad x = \Psi s$
- $\min_x \|x\|_0 \quad s.t. \quad \|x - \Psi s\|_2 \leq \epsilon$

In general, finding a minimum-cardinality solution satisfying linear constraints is an NP-combinatorial problem and an approximation is necessary to achieve computational efficiency. Two main approximation approaches are typically used in sparse recovery: the first one is to address the original NP-combinatorial problem via approximate methods, such as greedy search, while the second one is to replace the intractable problem with its convex relaxation that is easy to solve. In other words, one can either solve the exact problem approximately, or solve an approximate problem exactly.

In figure below, we can see sparse decomposition of our signal of interest. Notice that there are only K coefficients active, while others are equal to zero and that is exactly what we wanted to achieve.

```
%sparse decomposition
nIter = 10;
s = sparseCode(x, psi, K, nIter);

% figure
stem(real(s))
title('Sparse decomposition - l_0 norm')
```



Compressed Sensing

The key idea behind compressive sensing is that the majority of real-life signals(images, audio...) can be well approximated by sparse representation vectors, given some appropriate basis Ψ, and that exploiting the sparse signal structure can dramatically reduce the signal acquisition cost. Traditional approach to signal acquisition is based on the classical Shannon- Nyquist result stating that in order to preserve information about a signal, one must sample the signal at a rate which is at least twice the signal's bandwidth, defined as the highest frequency in the signal's spectrum. Note, however, that such classical scenario gives a worst-case bound, since it does not take advantage of any specific structure that the signal may possess. In practice, sampling at the Nyquist rate usually produces a tremendous number of samples, e.g., in digital and video cameras, and must be followed by a compression step in order to store or transmit this information efficiently.

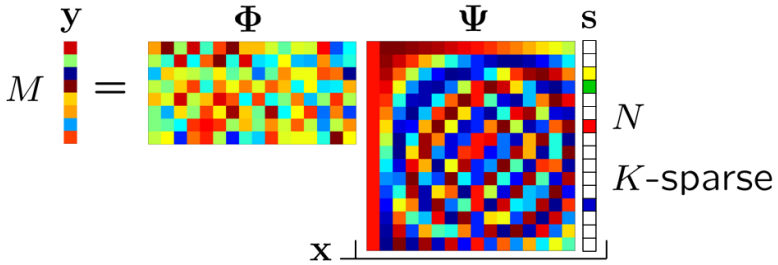
The compression step uses some basis to represent a signal (e.g., Fourier, wavelets, etc.) and essentially throws away a large fraction of coefficients, leaving a relatively few important ones. Thus, a natural question is whether the compression step can be combined with the acquisition step, in order to avoid the collection of an unnecessarily large number of samples.

Compressive sensing offers positive answer to the above question. Let $x \in \mathbf{R}^N$ be a signal that can be represented sparsely in some basis Ψ i.e. $x = \Psi s$ where Ψ is an $N \times N$ matrix of basis vectors(columns), and where $s \in \mathbf{R}^N$ is a sparse vector of the signal's coordinates with only $K \ll N$ nonzeros. Though the signal is not observed directly, we can obtain a set of linear measurements:

$$y = \Phi x = \Phi \Psi s = As$$

where Ψ is an $N \times M$ measurement matrix and $y \in \mathbf{R}^M$ is a set of M measurements or samples where M can be much smaller than the original dimensionality of the signal, hence the name compressive sensing(CS).

The central problem of compressed sensing is reconstruction of a high-dimensional sparse signal representation x from a low-dimensional linear observation y.



In general, l_q norms for particular values of q, denoted $\|x\|_q$ or more precisely their q-th power $\|x\|_q^q$ are frequently used as regularizers $R(x)$ in constrained optimization problems:

$$\min_{x \in \mathbf{R}^N} R(x) \quad s.t. \quad y = Ax$$

For a $q \geq 1$, the l_q norm, also called just q-norm of a vector $x \in \mathbf{R}^N$ is defined as:

$$\|x\|_q = (\sum_{i=1}^{N} |x_i|^q)^{\frac{1}{q}}$$

We can now observe relation between cardinality and $\|l_q\|$-norms. The function $\|x\|_0$ referred to as l_0-norm of x is defined as a limit of $\|x\|_q^q$ when $q \to 0$:

$$\|x\|_0 = \lim_{q \to 0} \|x\|_q^q = \lim_{q \to 0} \sum_{i=1}^{p} |x_i|^q = \sum_{i=1}^{p} \lim_{q \to 0} |x_i|^q$$

For each x_i, when $q \to 0$, $|x_i|^q \to I(x_i)$, the indicator function, which is 0 at $x = 0$ and 1 otherwise. Thus, $\|x\|_0 = \sum i = 1^p I(x_i)$, which gives exactly the number of nonzero elements of vector x called cardinality. Using the cardinality function, we can now write the problem of sparse signal recovery from noiseless linear measurements as:

$$\min_x \|x\|_0 \quad s.t. \quad y = Ax$$

As already mentioned before, the above optimization problem is NP-hard and no known algorithm can solve it efficiently in polynomial time. Therefore, approximations are necessary and were already presented before. Under appropriate

conditions the optimal solution can be recovered efficiently by certain appproximate techniques.

First approach to approximation is a heuristic-based search such as gready search. In gready search method, one can start with a zero vector and keep adding nonzero coefficients one by one, selecting at each step the coefficient that leads to the best improvement in the objective function(gready coordinate descent). In general, such heuristic search methods are not guaranteed to find the global optimum. However, in practive, they are simple to implement, very computationally efficient and under certain conditions they are even guaranteed to recovel the optimal solution.

An alternative approximation technique is the relaxation approach based on replacing an intractable objective function or constraint by a tractable one. For example, convex relaxations approximates a non-convex optimization problem by a convex one, i.e. by a problem with convex objective and convex constraints. Such problems are known to be "easy", i.e. there exists a variety of efficient optimization methods for solving convex problems. Clearly, besides being easy to solve, e.g., convex, the relaxed version of the optimization problem must also enforce solution sparsity. In the following sections, we discuss lq-norm based relaxations, and show that the Zi-norm occupies a unique position among them, combining convexity with sparsity. A convex optimization problem is minimization of a convex function over a convex set of feasible solutions defined by the constraints. Convex problems are easier to solve than general optimization problems because of the important property that any local minima of a convex function is also a global one. Optimization problem using l_1 norm writes:

$$\min_x \|x\|_1 \quad s.t. \quad y = Ax$$

Algorithms for Sparse Recovery

We will focus on the noisy sparse recovery problems:

- l_0-norm minimization: $\min_x \|x\|_0 \quad s.t. \quad \|x - \Psi s\|_2 \leq \epsilon$
- l_1-norm relaxation: $\min_x \|x\|_1 \quad s.t. \quad \|x - \Psi s\|_2 \leq \epsilon$
- Lagrangian form l_1 minimization (LASSO): $\min_x \frac{1}{2}\|y - Ax\|_2^2 + \lambda\|x\|_1$

Recall that x is an N-dimensional unknown sparse signal, which in a statistical setting corresponds to a vector of coefficients of a linear regression model, where each coefficient x_i signifies the amount of influence the i-th input, or predictor variable A_i, has on the output y, an M-dimensional vector of observations of a target variable Y. A is an $M \times N$ design matrix, where the i-th column is

an M-dimensional sample of a random variable A_i, i.e. a set of M independent and identically distributed, or i.i.d., observations.

We would like to focus on the specific case of orthogonal design matrices. It turns out that in such case both l_0- and l_1-norm optimization problems decompose into independent univariate problems, and their optimal solutions can be easily found by very simple univariate thresholding procedures.

Univariate Thresholding

An orthonormal(orthogonal) matrix A is an $N \times N$ square matrix satisfying $A^T A = AA^T = I$ where I denotes the identity matrix. A linear transformation defined by an orthogonal matrix A has a nice property, it preserves the l_2-norm of a vector.

$$\|Ax\|_2^2 = (Ax)^T(Ax) = x^T(A^T A)x = x^T x = \|x\|_2^2$$

The same is also true for A^T and we get:

$$\|y - Ax\|_2^2 = \|A^T(y - Ax)\|_2^2 = \|\hat{x} - x\|_2^2 = \sum_{i=1}^{N}(\hat{x}_i - x_i)^2$$

where $\hat{x} = A^T y$ corresponds to the ordinary least squares(OLS) solution when A is orthogonal, i.e. $\hat{x} = \min_x \|y - Ax\|^2$ This transformation of the sum-squared loss will greatly simplify our optimization problems.

l_0-norm Minimization

The problem of l_0-norm minimization can now be rewritten as:

$$\min_x \|x\|_0 \quad s.t. \quad \sum_{i=1}^{N}(\hat{x}_i - x_i)^2 \leq \epsilon^2$$

In other words, we are looking for the sparsest (i.e., smallest l_0-norm) solution x^* that is ϵ-close in l_2-sense to the OLS solution $x = A^T y$.

It is easy to construct such solution by choosing k largest (in the absolute value) coordinates of x and by setting the rest of the coordinates to zero, where k is

the smallest number of such coordinates needed to get ϵ-close to $\hat{x}$, i.e. to make the solution feasible.

This can also be viewed as an univariate hard thresholding of the OLS solution $\hat{x}$, namely:

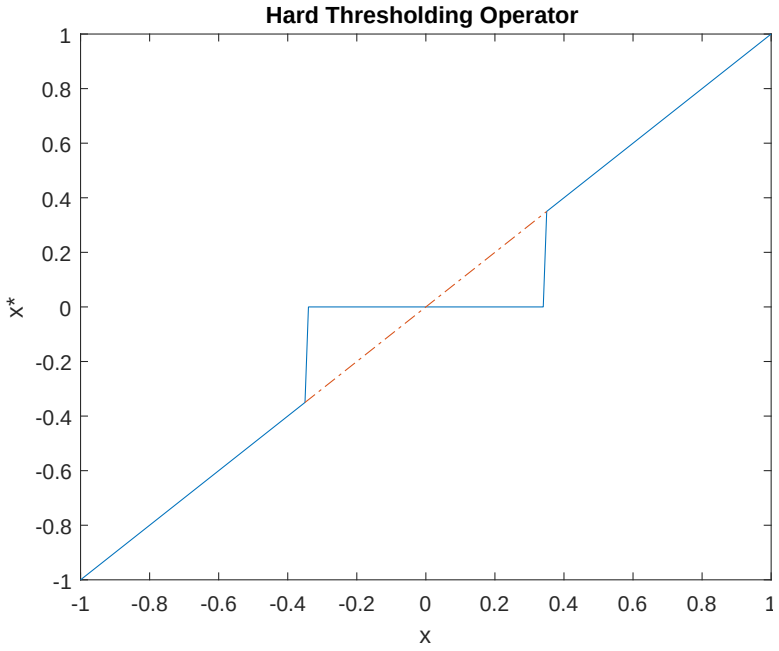
$$x_i^* = H(\hat{x}_i, \epsilon) = \begin{cases} \hat{x}_i & \text{if} |\hat{x}_i| \geq t(\epsilon) \\ 0 & \text{if} |\hat{x}_i| < t(\epsilon) \end{cases}$$

where $t(\epsilon)$ is a threshold value below the k-th largest, but above the $(k+1)$-th largest value among $\{\hat{x}_i\}$.

```
%hard thresholding
hardThresholding = @(x, th) x.*(abs(x)>=th);

x = -1:0.01:1;
threshold = 0.35;
x_ht = hardThresholding(x, threshold);

figure, axis tight
plot(x, x_ht)
hold on
plot([-threshold:0.01:threshold], [-threshold:0.01:threshold], '-.')
title('Hard Thresholding Operator')
xlabel('x')
ylabel('x*')
```



l_1-norm Minimization

For an orthogonal A, the LASSO problem becomes:

$$\min_x \frac{1}{2}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2 + \lambda \sum_{i=1}^{N}|x_i|$$

which trivially decomposes into N independent, univariate optimization problems, one per each x_i variable, $i = 1, ...N$:

$$\min_{x_i} \frac{1}{2}(\hat{x}_i - x_i)^2 + \lambda|x_i|$$

Global minimum solution for l_1-norm minimization can be obtained using soft thresholding operator:

$$x_i^* = S(\hat{x}, \lambda) = sign(\hat{x}(|\hat{x}|) - \lambda)_+$$

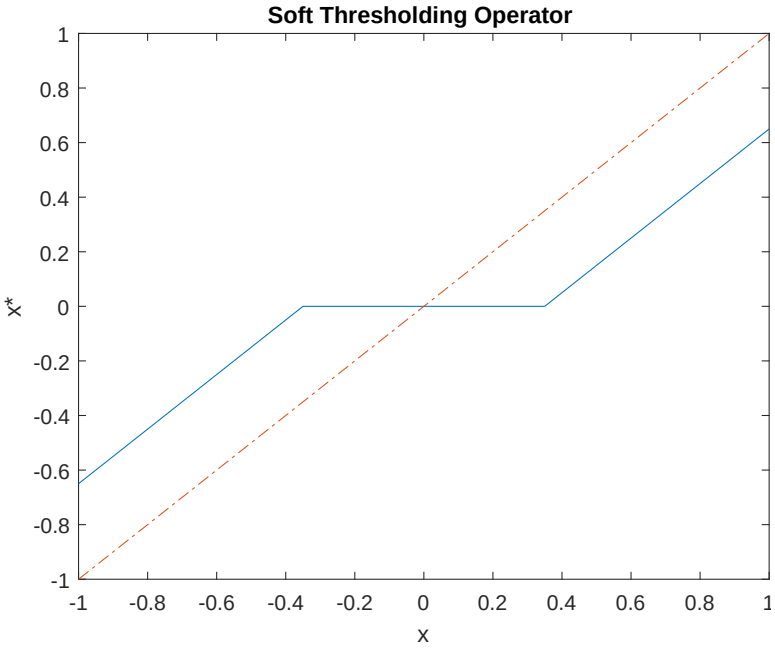
When the design matrix A is orthogonal, both l_0- and l_1-norm minimization problems decompose into a set of independent univariate problems which can

be easily solved by first computing the OLS solution $\hat{x} = A^T y$ and then applying thresholding operators to each coefficient.

```
%soft thresholding
softThresholding = @(x, th) sign(x).*max(abs(x)-th,0);

x = -1:0.01:1;
threshold = 0.35;
x_st = softThresholding(x, threshold);

figure, axis tight
plot(x, x_st)
hold on
plot(x, x, '-.')
title('Soft Thresholding Operator')
xlabel('x')
ylabel('x*')
```



ALGORITHMS FOR l_0-norm minimization