Compressed Sensing – Sheet 12

Ferdinand Vanmaele

November 8, 2022

Exercise 1 (Programming project part 1).

- The generation of sensors is done in `sheet12_ex1/sensors.py`. To make the matrices work with the assumptions of CoSaMP, OMP, SP and MP as defined in the lecture, the columns are normalized.

  - m rows are drawn uniformly at random (without repetition) by generating a random permutation $S(m) \to S(m)$.

- Random s-sparse vectors $x \in \mathbb{R}^n$ with $1 \leq s \leq m$ are constructed in `sheet12_ex1/main.py`. The `generate_problems()` function takes the problem dimension n and bounds for s, as well as the number repetitions for each sparsity value.

- Algorithms are run in `sheet12_ex1/main.py`. Every algorithm is implemented in its own function[1] and can be enabled or disabled selectively.

  Remark. The following modifications were made:

  - A tolerance of 10^{-8} was chosen te reduce oscillations for different sparsity levels.
  - Iterative hard thresholding was implemented with a variable step size [TBDR12],

  $$g := A^* r^{(k)}, \quad T := \begin{cases} \operatorname{supp}(\mathcal{T}(g, s)) & \text{if } k > 0 \\ \operatorname{supp}(\mathcal{T}(x^{(0)}, s)) & \text{if } k = 0 \end{cases}, \quad \mu := \frac{\|g_T\|_2^2}{\|A_T g_T\|_2^2}$$

  instead of $\mu = 1$. In the experiments performed, this was done to avoid divergence with $\mu = 1$.

- Let $\mathrm{RE}(\hat{x}) := \|x - \hat{x}\|_2 / \|x\|_2$. We consider successful recovery of a vector of sparsity $1 \leq s \leq m$ as the maximum s, such that

  $$s_{\max} := \max_s \left\{ \forall s' \leq s : \frac{1}{100} \sum_{i \in [100]} \mathrm{RE}(\hat{x}_{s',i}) < 10^{-6} \right\}$$

  holds. Summary of the trials, with $n = 128$, $m = 2^7$:

[1] This is a bit repetitive, especially for closely related algorithms such as CoSaMP and SP. However, every different algorithm to be tested is self-contained in this way. Outside of this exercise, I would likely summarize some functions and add additional parameters.

Matrix		Algorithm	$s_{\max}$	$\approx \sum$ RE
Random (A)	1	ℓ_1-min. (BP)	$s=17$	10^{-9}
	2	OMP	$s=13$	10^{-15}
	3	MP	$s=5$	10^{-9}
	4	IHT	$s=10$	10^{-9}
	5	CoSaMP	$s=14$	10^{-15}
	6	BT	$s=0$	–
	7	HTP	$s=10$	10^{-15}
	8	SP	$s=15$	10^{-15}
Fourier (F)	1	ℓ_1-min. (BP)	$s=40$	10^{-10}
	2	OMP	$s=34$	10^{-15}
	3	MP	$s=10$	10^{-7}
	4	IHT	$s=26$	10^{-9}
	5	CoSaMP	$s=25$	10^{-15}
	6	BT	$s=0$	–
	7	HTP	$s=26$	10^{-15}
	8	SP	$s=31$	10^{-15}

- Conclusions:
  - Methods using orthogonal projection recover the solution at a high accuracy (close to machine precision) for $1 \leq s$ sufficiently small. IHT is still competitive in terms of $s_{\max}$, with a higher recovery level.
  - The recovery error increased steeply from $s_{\max}$ to higher levels $s > s_{\max}$. Some trials also had oscillatory behavior.
  - The Fourier matrix was recovered for higher sparsity levels for all algorithms except Basic Thresholding. The coherence of this matrix is much lower than the Gaussian one (≈ 0.08 vs. ≈ 1.00) so this is unsurprising.
  - ℓ_1-minimization had constant performance for all trials. Methods using orthogonal projection had their CPU time increase exponentially. (See Figure 3 and 4)

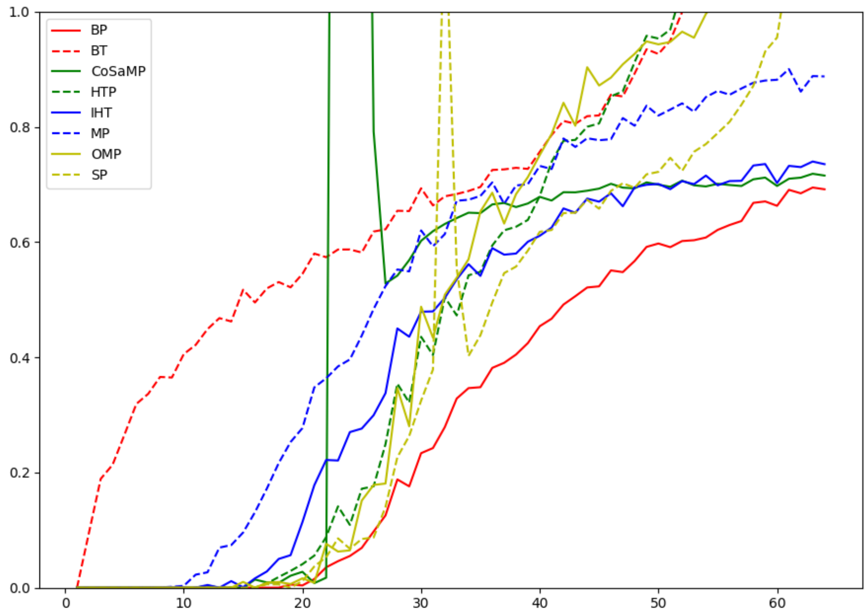


Figure 1: Relative recovery error (y-axis) over 100 trials per sparsity $1 \leq s \leq m$ (x-axis) for **Gaussian matrix** $A \in \mathbb{R}^{64 \times 128}$ with normalized columns.

Exercise 2.

Problem definition

The robust face recognition problem is given by

$$\min_{w \in \mathbb{R}^{m+n}} \frac{1}{2} \|Bw - b\|_2^2 + \lambda \|w\|_1$$

where

- $\lambda \in \mathbb{R}_{>0}$ is a regularization parameter,
  - $B = [A \quad I] \in \mathbb{R}^{m \times (m+n)}$ with $A \in \mathbb{R}^{m \times n}$, $I \in \mathbb{R}^{m \times m}$ a highly correlated dictionary of m face images $v \in \mathbb{R}^n$, stacked column-wise;
  - I the identity matrix $\in \mathbb{R}^{n \times n}$, $b = Ax + e$ a face image to be recognized not in the dictionary A, with $e \in \mathbb{R}^n$ an unknown error;
  - w the minimization variable.

The term $F(w) := \frac{1}{2}\|Bw - b\|_2^2$ is differentiable,

$$\nabla F(w) = B^T (Bw - b) \tag{1}$$

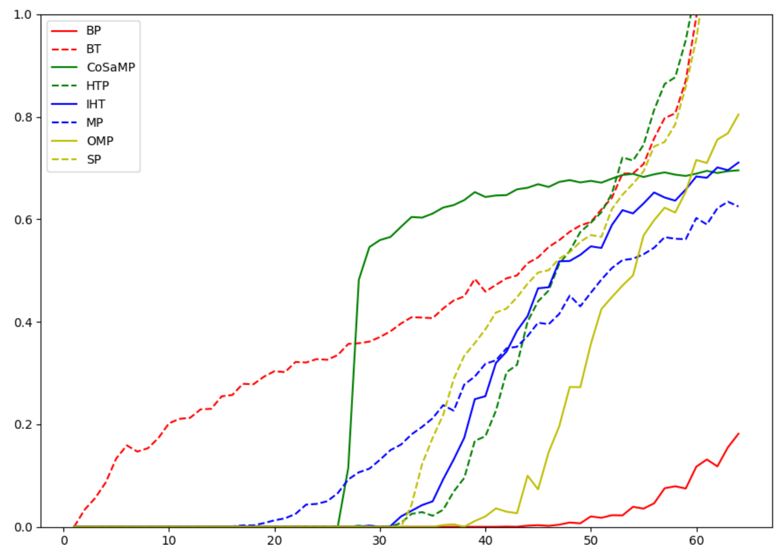


Figure 2: Relative recovery error (y-axis) averaged over 100 trials, by sparsity level $1 \leq s \leq m$ (x-axis) for **partial Fourier matrix** $F \in \mathbb{R}^{64 \times 128}$ with normalized columns.

with Lipschitz constant $L := \|B^T B\|_F$,

$$\begin{aligned} \|B^T(Bw - b) - B^T(Bz - b)\|_2 &= \|B^T Bw - B^T b - B^T Bz + B^T b\|_2 \\ &= \|B^T Bw - B^T Bz\|_2 \\ &\leq \|B^T B\|_F \|w - z\|_2. \end{aligned}$$

The norm $\|w\|_1$ is proper, convex and lsc and has as proximal operator **soft shrinkage**, given component-wise by

$$\left[\operatorname{Prox}_{\gamma\|\cdot\|}(w)\right]_i = \begin{cases} w_i + \gamma & \text{if } w_i < \gamma, \\ 0 & \text{if } -\gamma \leq w_i \leq \gamma, \\ w_i - \gamma & \text{if } w_i > \gamma. \end{cases}$$

Multiplying by $\lambda > 0$ we have:

$$\begin{aligned} \left[\operatorname{Prox}_{\gamma R}(w)\right]_i &= \lambda^{-1} \left[\operatorname{Prox}_{\lambda^2\gamma\|\cdot\|}(\lambda w)\right]_i \\ &= \begin{cases} w_i + \lambda\gamma & \text{if } w_i < \lambda\gamma, \\ 0 & \text{if } -\lambda\gamma \leq w_i \leq \lambda\gamma, \\ w_i - \lambda\gamma & \text{if } w_i > \lambda\gamma. \end{cases} \end{aligned}$$

The non-zero entries of x_* in the solution $w_* = [x_*^T, e_*^T]^T$ of problem (1) represent the person the image belongs to.

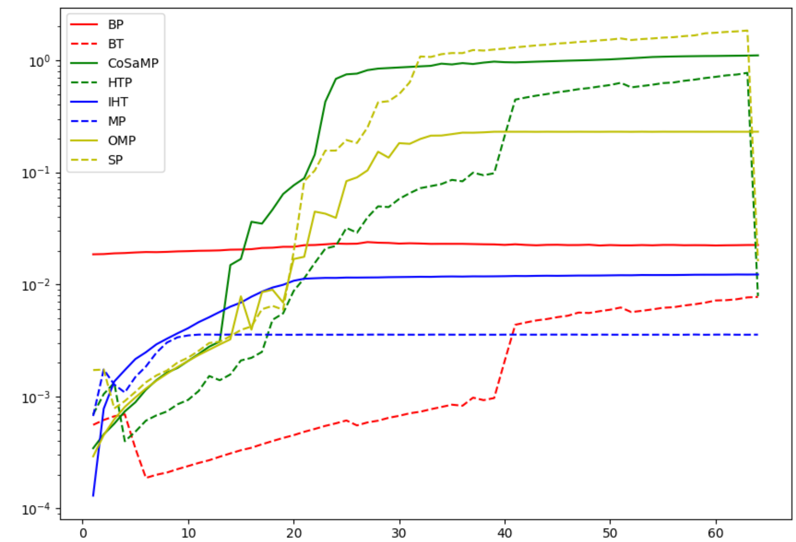


Figure 3: Combined CPU time in seconds per iteration (y-axis) averaged over 100 trials, by sparsity level $1 \leq s \leq m$ for **Gaussian matrix** $A \in \mathbb{R}^{64 \times 128}$ with normalized columns.

Remark. When we consider images without noise, we have the optimization problem

$$\min_{x \in \mathbb{R}^m} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \tag{2}$$

with gradient $\nabla F(w) = A^T(Ax - b)$ and proximal operator as above. The non-zero entries of the solution x_* represent the person the image belongs to. In pratice this can only hold approximately, and we look instead at the vector

$$|x|_{\downarrow} = (|x|_{[1]}, |x|_{[2]}, \dots), \quad |x|_{[1]} \geq |x|_{[2]} \geq \cdots$$

of sorted by absolute magnitude of its elements.

Face recognition data set

We assemble the dictionary A as follows. The baseline is the *Labeled Faces in the Wild*[2] (LFW) dataset, with each image of size $(250, 250)$. Images in this dataset are aligned, but not cropped so that background details (and potentially multiple faces) are visible. In a first attempt, A is constructed by randomly selecting 508 persons from this dataset which have at least 5 images available. Then, 4 images are selected randomly to construct A (training set), and 1 image is selected randomly for the complement for the reconstruction (verification set).

```python
for person in people:
```

[2]https://www.kaggle.com/datasets/jessicali9530/lfw-dataset

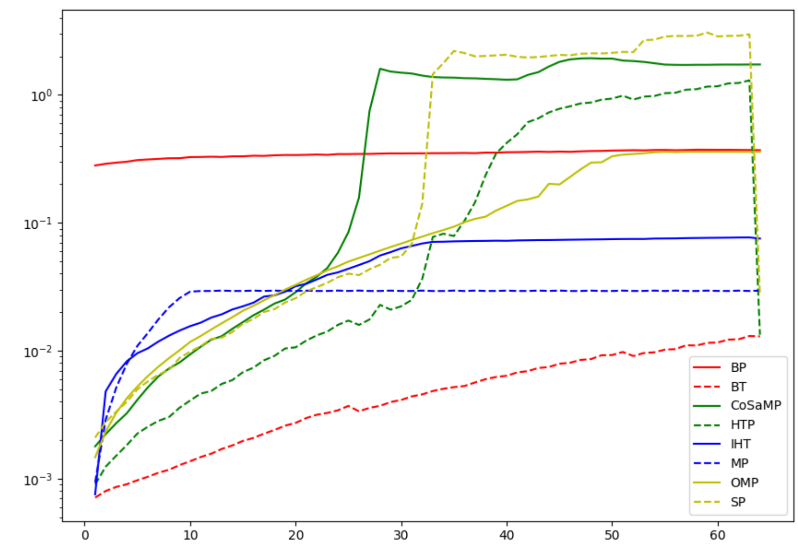


Figure 4: Combined CPU time in seconds per iteration (y-axis) averaged over 100 trials, by sparsity level $1 \leq s \leq m$ for **partial Fourier matrix** $F \in \mathbb{R}^{64\times 128}$ with normalized columns.

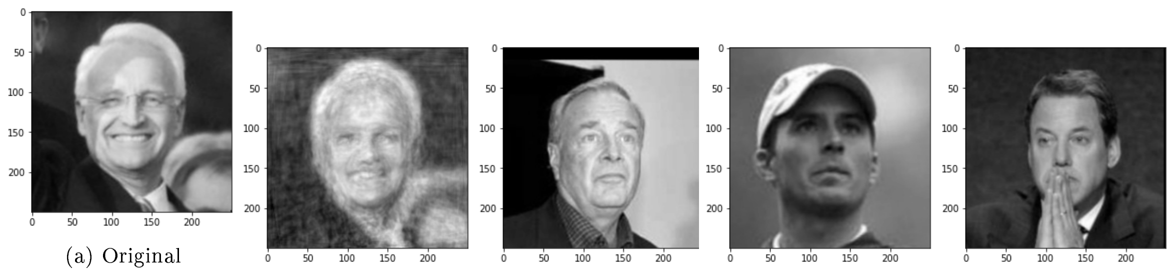
```
    # reduce number of people in data set for performance reasons
    dice_roll = np.random.choice([1,2,3,4,5,6])
    if dice_roll == 1 or dice_roll == 3:
        idx = people[person]

        if len(idx) >= min_samples:
            idx_train = np.random.choice(idx, num_train)
            idx_verif = np.random.choice(np.setdiff1d(idx, idx_train),
                num_verif)

            training_set[person] = idx_train
            verification_set[person] = idx_verif
```

The parameters for FISTA-Mod were chosen as follows:

- $\lambda = 10^{-3}$ and $x_0 = 0 \in \mathbb{R}^{n+m}$;
- $TOL = 10^{-7}$ where the algorithm terminates if $\|x^{(k-1)} - x^{(k)}\| < TOL$ or 5000 iterations were exceeded.
- $p = \frac{1}{20}$, $q = \frac{1}{2}$, $r = 4$.

For recognizing original images (that is, images with no noise added), FISTA-Mod applied to problem (2) could not uniquely recover a sample from the verification set, even though the sequence $\{x^{(k)}\}$

converged. See Figure 5. The cause was likely images having more than a single face available or other background artifacts. See Figure 5 and 6.



(a) Original

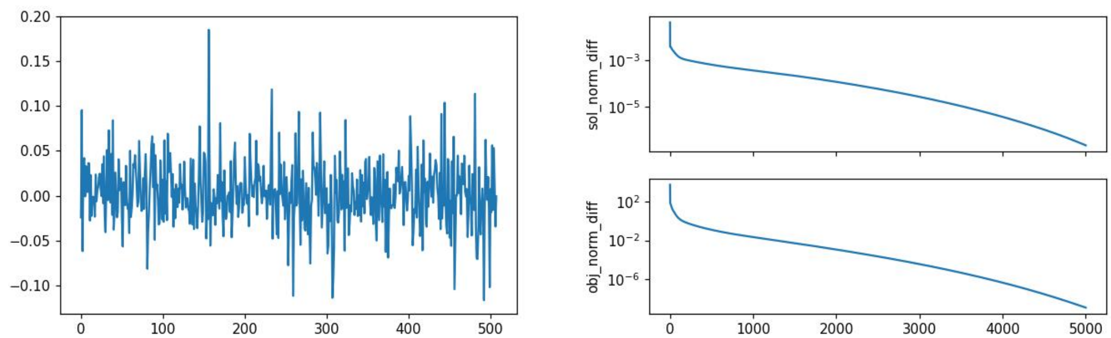
(b) Bx^*

(c) $B_{|x^*|_{[1]}}$

(d) $B_{|x^*|_{[2]}}$

(e) $B_{|x^*|_{[3]}}$

Figure 5: Recovered images for uncropped 250x250 sample



(a) Values of x^*

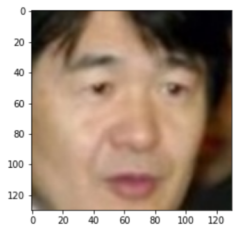
(b) Differences $J(x^{(k)}) - J(x^{(k-1)})$ and $\|x^{(k)} - x^{(k-1)}\|$ by number of iterations

Figure 6: Convergence and values of x^* for uncropped 250x250 sample and $\lambda = 10^{-3}$

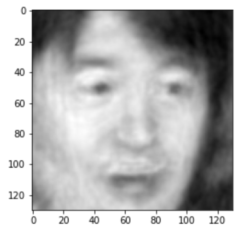
To address this, the images were cropped to regions containing a face using cascade classifiers.[3] Since the resulting images had different sizes, all samples below 100 pixels and above 130 pixels height and width were removed. The rest were then resized to 130 pixels height and width. The final training set contained 776 images, so that $A \in \mathbb{R}^{16900\times776}$, and an image from the verification set containing 245 items was selected. This time around, the algorithm managed to recognize the person belonging to given sample b. The parameter λ was increased to $\lambda = 1$, and the tolerance lowered to $TOL = 10^{-4}$ with a maximum of 5000 iterations. See Figure 7 and 8.

Conclusion. FISTA-Mod applied to the face recognition problem managed to recognize a sample, provided a clear image of the same person is available in the dictionary A is available. “Clear image” implies a fair amount of preprocessing required: alignment (“funneling”) as provided by the LFW dataset, and cropping with cascade classifiers so that background details are not considered.

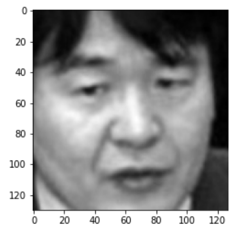
[3]https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html



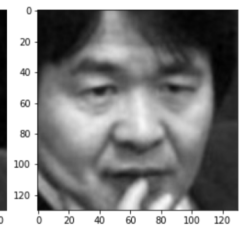
(a) Original



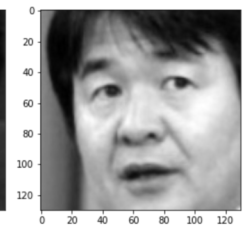
(b) Bx^*



(c) $B_{|x^*|_{[1]}}$

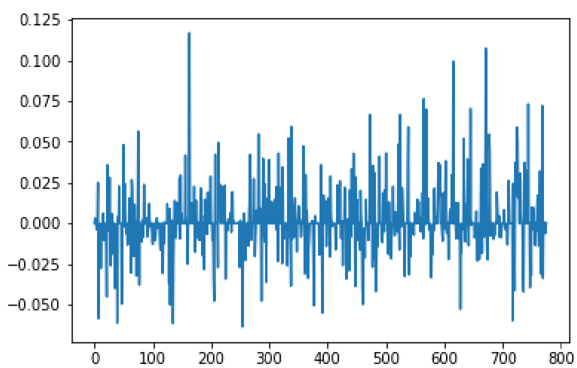


(d) $B_{|x^*|_{[2]}}$

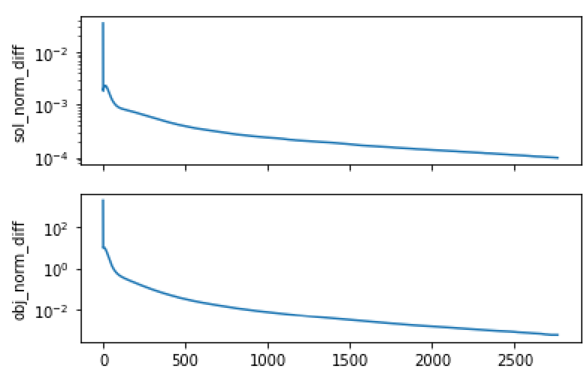


(e) $B_{|x^*|_{[3]}}$

Figure 7: Recovered images for cropped 130x130 sample and $\lambda = 1$



(a) Values of x^*



(b) Differences $J(x^{(k)}) - J(x^{(k-1)})$ and $\|x^{(k)} - x^{(k-1)}\|$ by number of iterations

Figure 8: Convergence and values of x^* for cropped 130x130 sample and $\lambda = 1$