Project: Reconstruction from Non-Uniform Samples Using a DCT-ℓ_p Prior

Majorization–Minimization with Conjugate Gradients (MM–CG)

1. Overview

This project introduces a modern optimization problem arising in *image reconstruction from incomplete data*. You will reconstruct a 2D image from only a subset of its pixels, assuming that the image is approximately sparse in the DCT (Discrete Cosine Transform) domain. The optimization problem uses a smooth, nonconvex ℓ_p-type penalty ($0.2 < p < 0.5$) to encourage sparsity.

Students will:

- Derive and implement a Majorization–Minimization (MM) algorithm with Conjugate Gradient (CG) inner solves.
- Explore the effect of sampling rate, noise, and regularization parameter λ.
- Evaluate results quantitatively (PSNR) and visually.
- Produce plots summarizing convergence and reconstruction quality.

This problem connects key optimization ideas (nonconvex objectives, quadratic majorization, iterative reweighting, matrix-free CG) to an applied inverse problem.

2. Mathematical formulation

Let $x^\star \in \mathbb{R}^N$ denote the true image (e.g., a 256×256 grayscale photo flattened into a vector). You observe only a subset of its pixels through

$$m \;=\; Wx^\star + \eta,$$

where $W : \mathbb{R}^N \to \mathbb{R}^M$ selects the M observed pixels, and η is additive Gaussian noise. Let M/N be the *sampling ratio* (e.g., 0.2 means 20% of pixels are known).

The reconstruction is obtained by solving

$$J(x) \;=\; \|Wx - m\|_2^2 + \lambda \sum_{i=1}^{N} (\varepsilon + (\mathrm{DCT}\, x)_i^2)^p, \qquad 0.2 < p < 0.5,\ \varepsilon > 0. \tag{1}$$

The first term enforces data consistency on measured pixels, while the second promotes sparsity of $\mathrm{DCT}\, x$. For $p < 0.5$, this prior is nonconvex but differentiable, producing stronger sparsity.

3. MM–CG algorithm (image domain)

At iteration k, define weights in the DCT domain:

$$w_i^{(k)} = p(\varepsilon + (\mathrm{DCT}\, x^{(k)})_i^2)^{p-1}, \quad i = 1, \ldots, N.$$

The MM quadratic surrogate of J leads to the linear system

$$\big(W^\top W + \lambda\ \mathrm{IDCT}(\mathrm{diag}(w^{(k)})\ \mathrm{DCT})\big)x = W^\top m, \tag{2}$$

which is solved approximately by Conjugate Gradients (CG). Each operator acts as:

$$(W^\top W)z = \mathrm{mask} \odot z, \qquad \mathrm{IDCT}(\mathrm{diag}(w^{(k)})\ \mathrm{DCT}\, z) = \mathrm{IDCT}(w^{(k)} \odot (\mathrm{DCT}\, z)).$$

All multiplications are pointwise, and transforms are performed via `dct2` / `idct2` (MATLAB) or `scipy.fftpack.dct` / `idct` (Python).

Algorithm 1 MM–CG for $J(x)$ in (1)

```
Require: observed pixels $m$, mask $M = W^\top W$, DCT/IDCT routines; parameters $\lambda$, $\varepsilon$, $p$.
Initialize $x^{(0)} = W^\top m$ (zero-filled image).
repeat
    Compute $\hat{y}^{(k)} = \mathrm{DCT}(x^{(k)})$.
    Set $w_i^{(k)} = p(\varepsilon + (\hat{y}_i^{(k)})^2)^{p-1}$.
    Define the operator $\mathcal{M}^{(k)}(z) = M \odot z + \lambda\ \mathrm{IDCT}(w^{(k)} \odot \mathrm{DCT}\, z)$.
    Solve $\mathcal{M}^{(k)}(x) = W^\top m$ by CG to tolerance $\mathrm{tol}_{\mathrm{CG}} = 10^{-6}$.
    Stop when relative change $\|x^{(k+1)} - x^{(k)}\| / \|x^{(k)}\| < 10^{-4}$.
until convergence
Output: $\hat{x} = x^{(k)}$, reconstructed image.
```

4. Experimental setup (detailed instructions)

4.1 Model images

Use at least two standard 256×256 grayscale images:

- `cameraman.tif`
- `Barbara` or `Lena`

Normalize pixel values to $[0, 1]$.

4.2 Sampling mask

Create random binary masks of varying sampling percentages:

$$r \in \{0.2,\ 0.3,\ 0.5\},$$

where r is the fraction of pixels retained. Construct M as a 0–1 array with exactly rN ones, randomly distributed.

4.3 Noise model

Add white Gaussian noise to the sampled pixels:

$$m = Wx^\star + \eta, \qquad \eta_i \sim \mathcal{N}(0, \sigma^2),$$

where σ corresponds to an SNR of 30 dB:

$$\mathrm{SNR} = 20 \log_{10} \frac{\|Wx^\star\|_2}{\|\eta\|_2}.$$

4.4 Choice of parameters

- $\varepsilon = 10^{-6}$.
- $p \in \{0.3,\ 0.4,\ 0.5\}$.
- For each experiment, sweep λ over a logarithmic grid:

  $$\lambda \in [10^{-4},\ 10^0],\ \text{ e.g., } \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}.$$

  Choose the λ that yields the best PSNR on the validation image.

4.5 Evaluation metrics

For each reconstruction $\hat{x}$, compute:

$$\mathrm{PSNR}(\hat{x}, x^\star) = 20 \log_{10} \frac{\|x^\star\|_\infty}{\|x^\star - \hat{x}\|_2/\sqrt{N}}.$$

Also report the ℓ_2 relative error $\|\hat{x} - x^\star\|_2/\|x^\star\|_2$.

Visualize:

- Reconstructed images and residual maps $(x^\star - \hat{x})$.
- Histogram of DCT coefficients before and after reconstruction.

4.6 Experiments

For each (r, p) combination, record the best λ, say $\lambda(r, p)$. For at least one combination of (r, p) record PSNR vs λ. For each case of $(r, p, \lambda(r, p))$, record the reconstructed image, image of the mask. For at least one case of $(r, p, \lambda(r, p))$, record:

- Objective value $J(x^{(k)})$ vs iteration index k.
- Relative change $\|x^{(k+1)} - x^{(k)}\|/|x^{(k)}\|$ vs iteration index k.
- Number of CG iterations per MM step vs iteration index k.

5. Deliverables

Submit a concise report (10 pages) including:

- Derivation of MM–CG algorithm (equations (2)–(1)).
- Description of experimental setup.
- Tables/plots of everything asked to record in 4.6.
- Discussion of how λ, p, and sampling rate affect sparsity and reconstruction.

Include commented code (MATLAB or Python) implementing the algorithm exactly as specified. The main submission should be in PDF. Jupyter notebook or any other similar form can only be a supporting submission. The PDF report should contain everything including code.