Snapshot Compressed Sensing

CS 754 - Advanced Image Processing

Kumar Ashutosh 16D070043
Sarthak Consul 16D100012

Aim

- To investigate the performance of the Snapshot Compression Algorithms - both theoretically and experimentally.

- To observe the performance using different sensing matrices, and to come up with theoretical explanations and proofs.

Algorithms - CbPGD and CbGAP

Algorithm 1 CbPGD for Snapshot CS Recovery

Require: $\mathbf{H}$, $\boldsymbol{y}$.
Initial $\mu > 0$, $\boldsymbol{x}^0 = 0$.
for $t = 0$ to Max-Iter **do**
  Calculate: $\boldsymbol{e}^t = \boldsymbol{y} - \mathbf{H}\boldsymbol{x}^t$.
  Projected gradient descent: $\boldsymbol{s}^{t+1} = \boldsymbol{x}^t + \mu \mathbf{H}^\top \boldsymbol{e}^t$.
  Projection via compression: $\boldsymbol{x}^{t+1} = g(f(\boldsymbol{s}^{t+1}))$.
end for
Output: Reconstructed signal $\hat{\boldsymbol{x}}$.

Algorithm 2 CbGAP for Snapshot CS Recovery

Require: $\mathbf{H}$, $\boldsymbol{y}$.
Initial $\mu > 0$, $\boldsymbol{x}^0 = 0$.
for $t = 0$ to Max-Iter **do**
  Calculate: $\boldsymbol{e}^t = \boldsymbol{y} - \mathbf{H}\boldsymbol{x}^t$.
  Euclidean projection: $\boldsymbol{s}^{t+1} = \boldsymbol{x}^t + \mu \mathbf{H}^\top \mathbf{R}^{-1} \boldsymbol{e}^t$.
  Projection via compression: $\boldsymbol{x}^{t+1} = g(f(\boldsymbol{s}^{t+1}))$.
end for
Output: Reconstructed signal $\hat{\boldsymbol{x}}$.

- R = HH^T in CbGAP.
- g(f(.)) is quantization function

Data Compression

- The algorithm uses data compression for further compression. We define two functions:

$$f: \ R^{nB} \longrightarrow \{1,2,...,2^{nBr}\}$$

And inverse $g: \{1,2,....,2^{nBr}\} \longrightarrow R^{nB}$

where r is the compression rate.

- g(f(x)) denotes quantization of the signal and this incurs quantization loss.
- The compression error is additive to the reconstruction loss due to compressive sensing and hence can be tuned separately.

Theoretical Performance Bounds

- Both algorithms will converge with high probability

$$\frac{1}{nB}\left\|\tilde{x}-x^t\right\|_2^2 \leq \delta, \quad \text{or} \quad \frac{1}{\sqrt{nB}}\|x^{t+1}-\tilde{x}\|_2 \leq \frac{2\lambda}{\sqrt{nB}}\left\|x^t-\tilde{x}\right\|_2+4\sqrt{\delta},$$

With probability

- Cb-PGD: $$1-2^{4nBr}e^{-\left(\frac{\delta}{2K\rho^2}\right)^2\lambda^2 n}-(2^{2nBr}+1)e^{-n\left(\frac{\delta}{2K\rho^2}\right)^2}.$$
- Cb-GAP: $$1-2^{4nBr}e^{-\frac{\lambda^2\delta^2 n}{2B\rho^4}}-2^{2nBr}e^{-\frac{n\delta}{2\rho^2B^2}}.$$
- When white-noise is added to the signal, there is an additional error term and the error probability will also depend on noise

Metric

- To evaluate the performance we use Peak Signal-to-Noise Ratio (PSNR)

$$PSNR(P, Q) = 10 \log_{10} \left(\frac{I_{max}}{MSE(P, Q)} \right)$$

where MSE is the mean square error between images P and Q.

I_{max} is the maximum intensity in the input images.

Simulation Results

- We simulate and verify the two algorithms CbPGD and CbGAP.
- We need to downsample the images in order to fit the capacity of CPU.
- The algorithm is fast since it is based on projections.
- Some of the parameters including the sensing matrix H and mean parameter μ is unspecified in the paper. Hence we choose suitable parameters.

Dataset

- We use sample videos from traffic dataset.
- For the purpose of this demonstration we consider a subset of 51 frames of one of the video samples.
- Contains RGB images.
- Here are two sample images 5 frames apart.

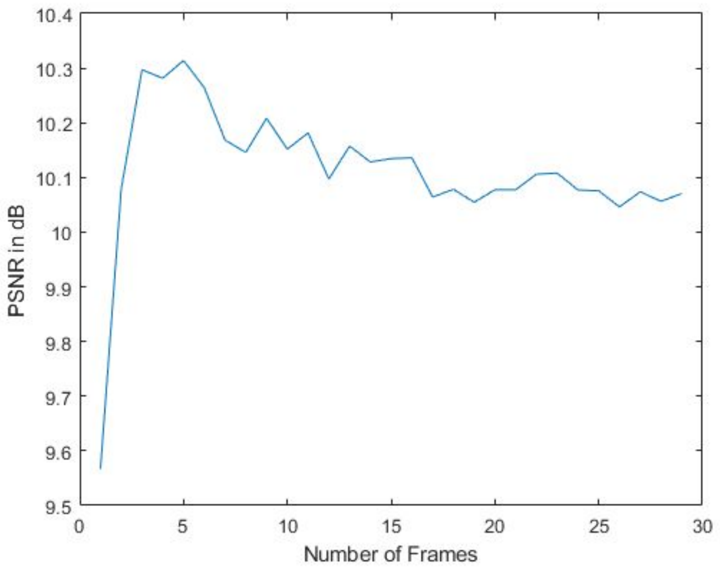
Quantitative Results

- We obtain good reconstruction results with a variety of sensing matrices H.
- We tried following choices of the sensing matrix H: random diagonal matrix, identity matrix and multiple of identity matrix.
- In all the cases, we obtained good reconstruction results. Although, the parameter μ needs to be adjusted for each case.

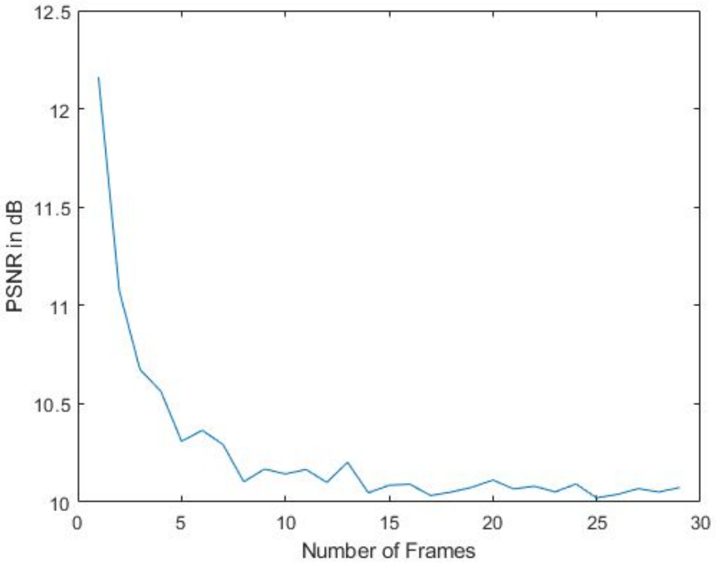
(Top) Ground truth image and (Bottom) Reconstructed image

Increasing number of frames

- We use this method for reconstruction of video frames simultaneously.



Algorithm 1 - **CbPGD**



Algorithm 2 - **CbGAP**

Inference

- We see that the reconstruction has a PSNR ~ 10dB which implies good reconstruction.
- This algorithm can take multiple frames as input and perform faithful reconstruction.
- The PSNR tends to decrease when we increase the number of simultaneous input frames.
- The two algorithms show good results for different values of parameter μ (0.08 and 1.5 for the above simulation, respectively)

Drawbacks of the Algorithms

- The performance of the algorithm is highly sensitive to the choice of parameter μ.
  - Increasing μ leads to divergent solutions.
  - Having very small μ leads to local optima
  - How to choose the parameter is not explicitly described in the paper.
- The structure of sensing matrix is not mentioned except for it being a diagonal matrix. The same is not presented in the simulation results as well.
- The size of the sensing matrix H is huge and the algorithm involves several multiplications including HH^T.

Thank You