Blind Compressed Sensing

Ameya Anjarlekar

June 2020

1 Introduction

Commonly used compressed sensing based signal reconstruction consists of representing a vector into typical bases such as DCT, Haar, Fourier, etc which provide a sufficiently sparse representation of the vector. However, there may exist bases which provide a much sparser representation of the vector. Thus, we can try to learn a dictionary for a particular set of signals which can represent the signals in the sparsest way. This is also known as Blind Compressed Sensing. The paper attempts to provide an algorithm for the same.
There have been various algorithms proposed for this task of dictionary learning, one of the popular algorithm being K-SVD. However, these papers mainly focus on the practical aspects of Blind Compressed Sensing (BCS). This papers proposes an algorithm for BCS while also studying its theoretical aspects.

2 Problem statement

Consider signals x_1, $x_2....x_n$. Let the observations corresponding to these signals be y_1, $y_2....y_n$ and the corresponding sensing matrices be Φ_1, $\Phi_2...\Phi_n$. We need to reconstruct the original signals from their respective observations by first finding a common basis for this set of signals which can represent them in the sparsest way possible and thus provide more accurate reconstruction of these signals. Thus, the loss function chosen is

$$L = min_{D,\Theta} \sum_{i=1}^{n} ||y_i - \phi_i D\theta_i||_2^2 + \lambda||\theta_i||_1 \tag{1}$$

where θ is the representation of the signals wrt the dictionary D.

3 Mathematical Analysis of the Results presented in the Paper

A rigorous mathematical analysis is done of the results proposed in the paper. This is mentioned in another document. Some queries regarding the same are highlighted in bold in the document and are also mentioned in another supporting document.

4 Brief overview of the algorithm

The algorithm is based on alternative minimization. We fix either the dictionary or the atoms and minimize the other. The algorithm steps are described below:
1) Initialize the dictionary by a random Gaussian initialization or by taking it as a typical dictionary such as DCT, Haar, Fourier, etc.
2) Estimate the vectors Θ using Lasso. For our implementation we used $l1_ls$ solver. Using the estimated values, calculate sigma given by

$$\sigma_i = \gamma^{-1}||y_i - \phi_i D\theta_i||_2 \tag{2}$$

where γ is a hyperparameter. Lesser the value of γ indicates a more faster but more unstable convergence. It can be better appreciated through the proof of the convergence rates.
3) Divide the observations for each signal into two equal parts. Thus for every signal we can divide sensing matrices as ϕ_i^1 and ϕ_i^2.
4) Using sigma we can solve the primal version of the optimization problem i.e.

$$min_{\theta_i}||\theta_i||_0 \quad s.t.||y_i - \phi_i^1 D\theta_i||_2 \leq \gamma \cdot \sigma_i \tag{3}$$

The paper solves l1 minimization while we are solving l0 minimization. This can be solved using OMP.
5) Now, optimize D. This can be done by performing a gradient descent step on the dual-loss function with ϕ_i replaced by ϕ_i^2.
6) Finally repeat 4 using the updated dictionary matrix and performing it for all samples i.e. replacing ϕ_i^1 by ϕ_i. Also, calculate sigma for each signal in this step.
7) Again go to step 4 if convergence is not reached else return the dictionary D and the original signals $x_i = D\theta_i$.

5 Data used

For the algorithm images - Barbara and Boat were used.

Figure 1: Barbara

Figure 2: Boat

6 Results

The images considered were Barbara and Boat with 50% samples. The code was run for iterations = 250, learning rate = 0.0001, $\lambda = 0.01$ and $\gamma = 0.95$.
1) Randomly initialized dictionary:
Barbara

Figure 3: Reconstructed image on random dictionary: PSNR = 8.8303

Figure 4: Reconstructed image on learned dictionary: PSNR = 11.3257

Boat

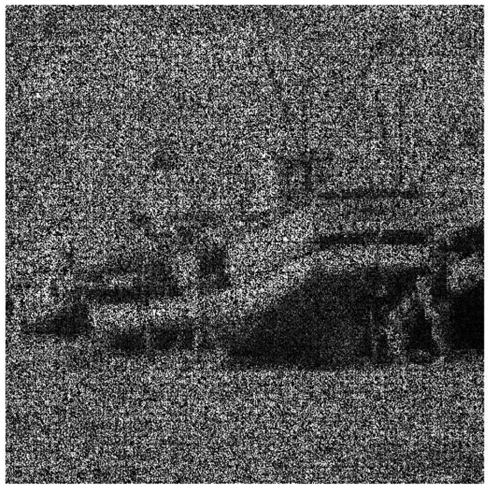

Figure 5: Reconstructed image on random dictionary: PSNR = 8.2154

Figure 6: Reconstructed image on learned dictionary: PSNR = 11.7630

2) DCT wavelet basis as the dictionary:

Figure 7: Reconstructed image on DCT Basis: PSNR = 17.9784

Figure 8: Reconstructed image on learned dictionary: PSNR = 18.3005

3) Using K-SVD learnt dictionary The dictionary was trained on the same image.

Figure 9: Reconstructed image on K-SVD Basis: PSNR = 28.3654

Figure 10: Reconstructed image on learned dictionary: PSNR = 28.8361

This also shows that a PSNR improvement is achieved by using a K-SVD trained dictionary.

4) Using binary sensing matrix (Image inpainting) and random dictionary

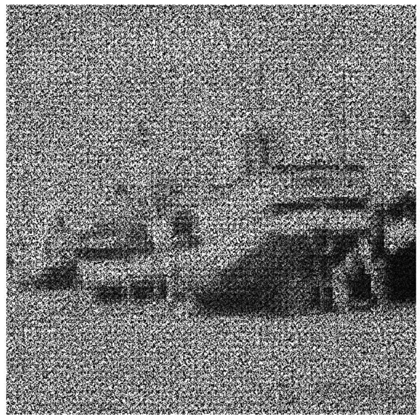

Figure 11: Reconstructed image on random dictionary: PSNR = 7.6507

Figure 12: Reconstructed image on learned dictionary: PSNR = 18.6357

7 Conclusion

The algorithm works best to fine-tune an initialized dictionary to best represent the image. Further, the paper also provides theoretical bounds which support the validity of the algorithm.

Good aspects:
1) Helps to increase the quality of the image.
2) Provides better quality than state of the art K-SVD.

Bad aspects:
1) Doesn't increase the quality much for a specified dictionary.
2) Computational cost is high.