The Scientific Temper (2023) Vol. 14 (1): 222-226.
Doi: 10.58414/SCIENTIFICTEMPER.2023.14.1.28
E-ISSN: 2231-6396, ISSN: 0976-8653
https://scientifictemper.com/

RESEARCH ARTICLE

Performance analysis of compressive sensing and reconstruction by LASSO and OMP for audio signal processing applications

Susithra N[1], Rajalakshmi K[2], Ashwath P[3]

Abstract

Audio signal processing is used in acoustic IoT sensor nodes which have limitations in data storage, computation speed, hardware size and power. In most audio signal processing systems, the recovered data constitutes far less fraction of the sampled data providing scope for compressive sensing (CS) as an efficient way for sampling and signal recovery. Compressive sensing is a signal processing technique in which a sparse approximated signal is reconstructed at the receiving node by a signal recovery algorithm, using fewer samples compared to traditional sampling methods. It has two main stages: sparse approximation to convert the signal into a sparse domain and reconstruction through sparse signal recovery algorithms. Recovery algorithms involve complex matrix multiplication and linear equations in sampling and reconstruction, increasing the computational complexity and leading to highly resourceful hardware implementations. This work reconstructs the sparse audio signal using LASSO and orthogonal matching pursuit (OMP) algorithm. OMP is an iterative greedy algorithm involving least square method that takes a compressed signal as input and recovers it from the sparse approximation, while LASSO is L1 norm based with a controlled L2 penalty. The paper reviews the reconstruction and study of sparsity and error obtained for reconstructing an audio signal by OMP and LASSO.

Keywords: *Orthogonal Matching Pursuit, Sparse approximation, Audio Signal Processing, Least Square Method, Compressive sensing,* IoT node, LASSO

Introduction

The important characteristic of sparse signals is the minimum number of non-zero coefficients in one of their transformation domains (Donoho, 2006). They can be reconstructed from reduced linear combinations of sparse coefficients (Elad, 2010). In certain applications, a reduced set of samples result as a consequence of their physical unavailability due to the intentional omitting of high noise or corrupted signals like audio signals (Stankovic, Stankovic and Amin, 2014). In some applications, sparsity is a result of a reduction in the number of observations while preserving the whole information (data compression) ((Donoho, 2006; Baraniuk, 2007).

Compressive sensing is a breakthrough development that allows sparse sampling signals under sub-nyquist rate and reconstructing the signal using a recovery algorithm. Reconstruction algorithms are complex with high computational intensity, and hardware implementation of these algorithms is difficult (Carrillo, Barner and Aysal, (2010). Orthogonal matching pursuit (OMP) is a two-step method involving Least mean square and QR Decomposition. This paper focuses on the simulation of the OMP algorithm using MATLAB for IoT applications.

Literature Survey

In the previous studies, novel design techniques based on the Internet of Things (IoT) are introduced for acquiring data from compressed signals using an OMP algorithm. Technical analysis and data dependence between the various phases of the OMP algorithm provides high throughput. Discrete cosine transform (DCT) along with compressive sampling

[1]Department of Electronics and Communication Engineering, PSG Institute of Technology and Applied Research, Coimbatore, Tamil Nadu, India

[2]Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

[3]PSG College of Technology, Coimbatore, Tamil Nadu, India

***Corresponding Author:** Susithra N, Department of Electronics and Communication Engineering, PSG Institute of Technology and Applied Research, Coimbatore, Tamil Nadu, India, E-Mail: nsusithra90@gmail.com

How to cite this article: Susithra, N., Rajalakshmi, K., Ashwath, P. (2023). Performance analysis of compressive sensing and reconstruction by LASSO and OMP for audio signal processing applications. The Scientific Temper, **14**(1):222-226

Doi: 10.58414/SCIENTIFICTEMPER.2023.14.1.28

Source of support: Nil

Conflict of interest: None.

Received: 15/01/2023 **Accepted:** 13/02/2023 **Published** : 25/03/2023

(CS) techniques, are used for audio signal compression. Using spectral analysis, DCT structures and CS stage, audio signals are represented sparsely in the frequency domain (Bi, Mitra and Li, 2013). Thus, compressive sampling represents the signals with less samples than a traditional sampling method. At the receiver side, reconstruction of the samples is proposed using the OMP algorithm, which constitutes two main processes: QR decomposition (QRD) and least square method (Davis, 1997; Chi, 2015).

Bai, et al. (2012) present FPGA implementations of the OMP and AMP algorithms. It was observed that, AMP is more suitable for less sparse problems while OMP performs faster for recovery problems with a smaller number of non-zero coefficients and more sparsity. The implementations show that CS reconstruction is feasible for real-time applications.

Stanislaus and Mohsenin (2013) provide an architecture for OMP-based reconstruction and has implemented scalable QR decomposition to reduce the processing time by at least 25%. So, OMP can be considered as the reconstruction algorithm for acoustic IoT node applications. There are roughly three types of sparsity promoting algorithms (a) greedy algorithms, (b) iterative shrinkage methods, and (c) convex optimization algorithms (Theodoridis, Kopsinis and Slavakis, 2012). OMP is greedy, and OMP and LASSO (Least Absolute Shrinkage and Selection Operator) are convex optimization problems.

Compressive Sensing

Compressive sensing has been known as a technique that has dawned new possibilities for signal detection and processing. It is a standard sampling theory that tries to minimize the required number of samples for effective signal reconstruction (Rabah et al., 2014) and aims to provide efficient sensing, transmission, and storage methods and facilitate signal processing in situations where certain information is not available. It relies on computational algorithms to solve the data reconstruction problem from a lessor number of samples by examining sparsity and inconsistencies (Zhang et al., 2015). This concept aims to provide an optimal solution in the frequency domain of representation.

The conventional basic method of signal reconstruction from its samples is the Shannon- Nyquist sampling theorem. The theorem states that reconstruction is effective only if the sample size is at least twice the maximum frequency of the signal. For exact reconstruction, the number of observations must be at least equal to the length of the signal. However, the traditional sampling and reconstruction method requires more storage space, sensing time, and many sensors while consuming more energy (Stanković and Daković, 2016).

Compressive sensing (CS) is a novel concept that surpasses the conventional methods. It shows that a sparse signal can be reconstructed from even very few incoherently observed samples (Rabah et al., 2014). Basically, it is assumed that, most physical signals in real-time applications have a sparse representation in a specific transform domain, i.e, only a few data coefficients are significant, and many are zero or nearly zero to be ignored. This kind of signal reduction is a basic requirement for pursuing compressive sampling (Sejdić et al., 2014). Another important requirement is the incoherent nature of the sample coefficients in the signal acquisition domain with outliers that do not fit in. CS aims to provide original signal reconstruction from a small number of incoherent coefficients by exploiting sparsity properties.

Reconstruction Methods

A Sparse signal is a Linear projection of a signal to a known basis. If X is input signal with sample vector length 'n', Y is output obtained by compressed sensing whose length m<<n, then is the sensing/measurement matrix with dimensions mxn, and

$$y = \phi x \tag{1}$$

$$\hat{x} = \psi \lambda \tag{2}$$

where is a basis function to make x sparse and incoherent, it is a nxn matrix. is the co-efficient sequence of length n and is the 'effective' sensing matrix

$$y = \theta\lambda = (\phi\psi)\lambda \tag{3}$$

Now reconstructing is a non-convex l_0 norm problem, which can be solved by following other regularisation schemes like LASSO or OMP with an l1 constraint or least squares evaluation support.

LASSO

LASSO is used to find the the least square solution on the solution vector subject to an l1-norm constraint. LASSO can be rewritten equivalently as an optimization problem

$$\min_x \frac{1}{2}\|y - Ax\|_2^2 + \alpha\|x\|_1 \tag{4}$$

where $\alpha \in [0, \infty)$ is a scalar regularization parameter that handles the trade-off between the mean square error and the l1-norm of the solution vector x. α is tuned to converge the solution for a large number of zeros in the vector x (sparsity).

Orthogonal Matching Pursuit

A number of compressed sample reconstruction methods have been developed in the previous studies, and they are one of the three main modes: convex optimization methods such as basis pursuit and gradient-based algorithms; greedy techniques like matching pursuit and orthogonal matching pursuit. Greedy algorithms bring lower computational complexity than other methods(Zhang et al., 2015). Matching pursuit (MP) is a common form of low-level signal reconstruction, combining greedily with the original signal's equivalent. The MP algorithm identifies a measurement matrix column closely related to the current signal ratio, followed by measuring the enhanced signal ratio. Even though each iteration in in the MP requires a very low computational effort, the number of iterations or repetitions depends largely on the level of sparsity 'k', and as a result, MP is suitable for signals with a high level of sparsity (Candès and Wakin, 2008). 11–minimization algorithm is another recovery algorithm commonly used with better accuracy, but the implementation is quite complex and time-consuming.

In OMP, the Least Square (LS) step significantly reduces the amount of repetition compared to Matching Pursuit, but it results in higher computations with each iteration (Rabah et al., 2014). The large number of inner-products, comparative functions, and matrix conversions attributes this complexity. Moreover, OMP is less complex as it helps in finding the most correlated values in each iteration (Sejdić et al., 2014). The OMP algorithm is best suited for audio signals and hardware implementations (Davis, 1997).

The audio signals are highly correlated signals. The algorithm (OMP) takes the measurement matrix Φ and the measured vector as input and gives the estimate $\tilde{x}$ of the original signal x. At the time of each iteration, it selects one of the columns most closely related to the residue of the measurements y, and then removes the contribution of this column to the new remaining calculation. In each stage, the right solution is found in the form of multiplication and residual update. It also incorporates new real signal limitations; after a number of iterations, the number being determined by the sparsity 'k', the algorithm will produce the final estimate of the original signal. The OMP algorithm is best suited for reconstructing two-dimensional sparse signals based on a few number of sample measurements Candès and Wakin, 2008).

Methodology

This work involves compressive sampling through sparse approximation and reconstruction by least square and QR decomposition method constituting the OMP algorithm as shown in Figure 1 and LASSO method.

Sparse Approximation

Sparse approximation indicates that a signal of length M is represented with a smaller number of samples N with non-zero coefficients where N<<M. It has two main objectives. There are a variety of sampling techniques that force the parameters to go directly to zero. Sparse approximation works like a type of regularization, leading to a simpler model that learns what parameters can be kept, and what parameters could be avoided.

Another reason to use a lesser amount of data is to efficiently reduce the memory usage, for example, when one needs to send the data over the network or store the data on mobile devices, it will consume less memory.

Least Square Method

Least Square Method is the process of finding the most suitable curve or line of proportion that is appropriate for a set of points by reducing the total sum of the squares of the residual points from the curve. During the process of finding the relationship between the two variables, the outcomes are measured quantitatively and the process is called regression analysis (Zhao et al., 2013)

Curve measurement method is a method of retrospective analysis. It is quite clear that the curves of a particular set of data are not always unique. Therefore, finding a curve with a slight deviation from all points of the measured data is necessary. This is known as a best-fitting curve. The least square is the method of approximating the curves to the given data by fitting equations.

QR Decomposition

QR decompression, also known as QR factorization, is a method used to convert matrix A to the form
A = QR.

where A represents the output matrix, Q represents the orthogonal matrix, and R is the upper triangular matrix. Usually, QR decomposition is used to solve the problem of least squares. To decompose a square or rectangular matrix into two parts, Q and R. QR decomposition makes it easier to perform linear operations, thus increasing the performance of a given task.

LASSO

The objective of OMP is to find an approximate solution for the l_0-norm minimization problem with least square method in the algorithm, while LASSO being a linear model trained with l_1-norm for regularising, solves the l_1-norm minimization problem. The 'L1 ratio' denoted by is 1 for zero L2 penalty. But fine-tuning of introduces variation in fitting the reconstruction while introducing L2 penalty towards an ordinary least squares problem. The lasso co-efficient of the Discrete cosine transformed random sparse samples of the original audio signal are determined and these coefficients are sufficient to reconstruct after inverse DCT.

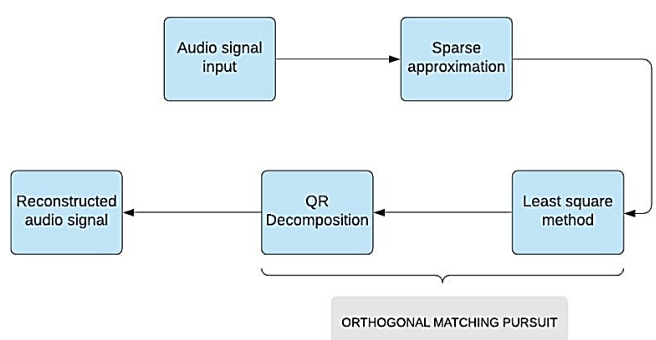

Figure 1: Block diagram of compressive sensing

Since tuning introduces L2 norm, sparsity is reduced for a real-time application like audio signal processing. Figure 2 shows the reconstructed signal when around 12% of the audio is made sparse in the transformed domain. Figure 3 shows the corresponding frequency spectrum. By decreasing the compression ratio, there is an increase in error. The error decrease with a decrease in in the LASSO algorithm (Figure 4).

OMP Algorithm

The matrices and parameters involved in executing the OMP algorithm shown in Table 1 is described below.

The inputs to the algorithm are Transform Matrix Ψ, Measurement Matrix ϕ, Compressive Sensing Matrix A (A=ϕΨ), Measurement vector y and sparsity k. A signal estimate is represented as $\tilde{x}$ and an approximation to the measurements is given by y.a_k is a matrix containing non-zero elements. A residue $r_k = y - a_k$, represents the output matrix of each iteration which are compared with y. Ω_k has the set of positions of non-zero elements of the signal estimate $\tilde{x}$.

Results and Discussion

Real-time audio signals have large data length. Eventually, the number of samples handled are very high. For ease of experimentation and to perceive the plots better, one frame of the audio signal for a duration less than a second is considered without windowing a long sequence of 1D audio signal. Figure 5 shows the original signal, audio signal after the sparse approximation and the reconstructed signal. There are 35841 samples in the audio signal and the sampling rate for the considered input signal is 44100 Hz.

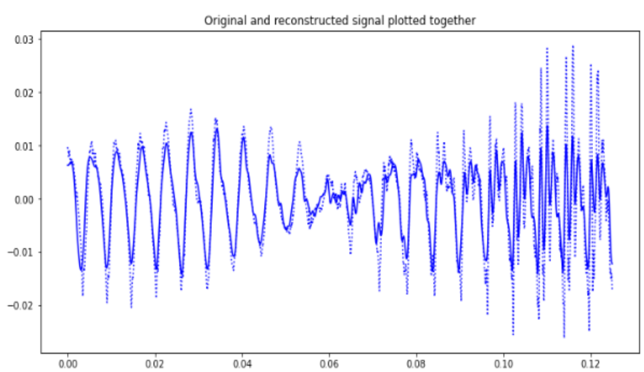

Figure 2: Overlayed original and reconstructed signals by LASSO reconstruction with compression ratio= 40% and 12.4% sparsity and Alpha tuning 0.005

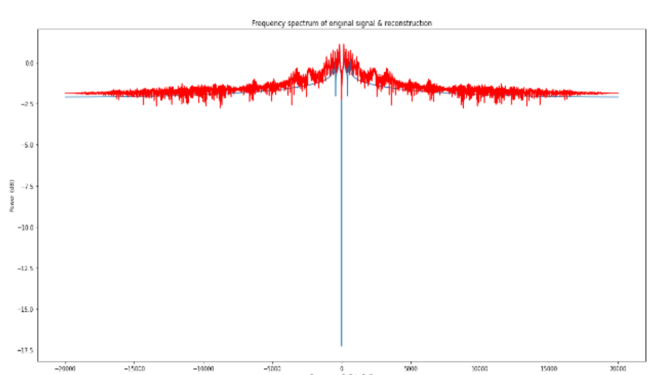

Figure 3: Frequency spectrum of original signal & reconstruction

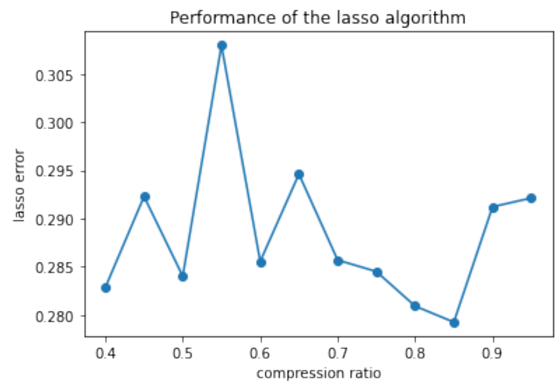

Figure 4: Performance of LASSO Algorithm

Table 1: OMP Algorithm

Initialize $r_0 \leftarrow y, \Omega_0 \leftarrow \phi, \theta_0 \leftarrow [\,]$
for $i = 1, \ldots, K$ $\omega_i \leftarrow \arg\max \lvert (r_{i-1}, A_j) \rvert_{j=1,\ldots N}$ $\Omega_i \leftarrow \Omega_{i-1} \cup \omega_i$ (Update the indices) $\theta_i \leftarrow [\theta_{i-1} A\omega_i]$ (Update the elements) $x_i = \arg\min \lVert r_{i-1} - \theta_i x \rVert_{x2}^2$ $a_i \leftarrow \theta_i x_i$ (New approximation) $r_i \leftarrow y - a_i$ (Update residual)
end for
return x_K, a_K, r_K, Ω_K

For OMP reconstruction, the compression parameter 'k' denoting the sparsity is set to 1500, which was arrived by trial-and-error method. The reconstructed signal has more error when 'k' was initialized to 100. So, it is gradually increased in steps of 500. As the 'k' value is increased the number of iterations also increase proportionally. As a result, there is a trade-off between the sparsity and the computation time to get a reconstructed signal with fewer or no error.

Figures 6 and 7 show the accuracy of the signal when the 'k' value is 100 & 1500 respectively. It can be seen clearly that the reconstructed signal's accuracy is more with least deviations when 'k' value is increased to 1500. Figure 8 shows the plot for the correlation of each sparse reconstructed signal with the original signal. For each value of 'k' the correlation of the reconstructed signal with input is different and is higher than that of smaller 'k' values. But, higher the 'k' value the execution time taken by the algorithm also increases. Hence, 'k' should be maintained optimally for effective reconstruction.

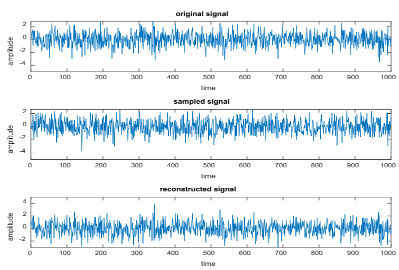

Figure 5: MATLAB plot original signal, sampled signal (after sparse approximation) and Reconstructed signal (after LS and QR Decomposition)

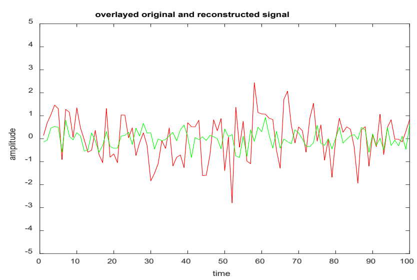

Figure 6: Overlayed original and reconstructed signals when k =100 (Original signal - Red, Reconstructed signal- Green)

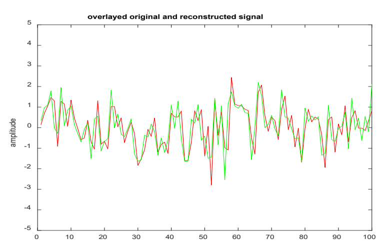

Figure 7: Overlayed original and reconstructed signals when k = 1500 (Original signal - Red, Reconstructed signal- Green)

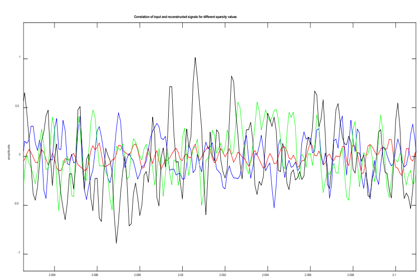

Figure 8: Overlap of correlation of reconstructed signals with the corresponding input at different sparsity values (k=100 - Red, k=500 - Blue k=1000- Green, k=1500 - Black)

Conclusion

This paper presents the execution of compressive sensing using MATLAB and Python and emphasizes the significance of LASSO and Orthogonal Matching Pursuit. In LASSO, optimized tuning of the parameters converges to a better solution with minimum L2 penalty which is a computational overhead for future hardware implementation. Also, efficient sparse signal recovery for varying sparsity is discussed. Since the OMP approach is greedy and iterative, fixing the sparsity judiciously reconstructs the compressed signal efficiently, providing scope for a better FPGA-based architectural implementation. Hardware implementation helps in reducing the computation time and power as well. Thus, these reconstruction methodologies show promising results adding scope for audio signal processing involving feature extraction and denoising in IoT applications where processing power is also a concern.

References

Bai, L., Maechler, P., Muehlberghuber, M., & Kaeslin, H. (2012). High-speed compressed sensing reconstruction on FPGA using OMP and amp. *19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012),* 53-56.

Baraniuk, R. G. (2007). Compressive sensing [lecture notes]. *IEEE signal processing magazine, 24(4),* 118-121.

Bi, G., Mitra, S. K., & Li, S. (2013). Sampling rate conversion based on DFT and DCT. *Signal processing, 93(2),* 476-486.

Candès, E. J., & Wakin, M. B. (2008). An introduction to compressive sampling. *IEEE signal processing magazine,* 25(2), 21-30.

Carrillo, R. E., Barner, K. E., & Aysal, T. C. (2010). Robust sampling and reconstruction methods for sparse signals in the presence of impulsive noise. *IEEE Journal of Selected Topics in Signal Processing, 4(2),* 392-408.

Chi, H. (2015). A discussions on the least-square method in the course of error theory and data processing. *International conference on computational intelligence and communication networks (CICN),* 486-489.

Davis, G. (1997). Greedy adaptive approximation. *Journal of Constructive Approximation, 13(1),* 57-98.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory, 52(4),* 1289-1306.

Elad, M. (2010). Sparse and redundant representations: from theory to applications in signal and image processing, *New York: Springer, 2 (1),* 1094-1097.

Rabah, H., Amira, A., Mohanty, B. K., Almaadeed, S., & Meher, P. K. (2014). FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction. *IEEE Transactions on very large scale integration (VLSI) Systems, 23(10),* 2209-2220.

Sejdić, E., Rothfuss, M. A., Gimbel, M. L., & Mickle, M. H. (2014). Comparative analysis of compressive sensing approaches for recovery of missing samples in implantable wireless Doppler device. *IET Signal Processing, 8(3),* 230-238.

Stanislaus, J. L., & Mohsenin, T. (2013). Low-complexity FPGA implementation of compressive sensing reconstruction. *International conference on computing, networking and communications (ICNC),* 671-675.

Stanković, L., & Daković, M. (2016). On a gradient-based algorithm for sparse signal reconstruction in the signal/measurements domain. *Mathematical Problems in Engineering, 2016,* 1–11.

Stankovic, L., Stankovic, S., & Amin, M. (2014). Missing samples analysis in signals for applications to L-estimation and compressive sensing. *Signal Processing, 94,* 401-408.

Theodoridis, S., Kopsinis, Y., & Slavakis, K. (2012). Sparsity-aware learning and compressed sensing: An overview. *Academic Press Library in Signal Processing.*

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological), 58(1),* 267-288.

Zhang, Z., Xu, Y., Yang, J., Li, X., & Zhang, D. (2015). A survey of sparse representation: algorithms and applications. *IEEE access, 3,* 490-530.

Zhao, L., Bi, G., Wang, L., & Zhang, H. (2013). An improved auto-calibration algorithm based on sparse Bayesian learning framework. *IEEE Signal Processing Letters, 20(9),* 889-892.