Comparative Analysis of Sparse Signal Reconstruction Algorithms for Compressed Sensing

Maytee Zambrano N.
Technological University of Panama, Panama, Panama, maytee.zambrano@utp.ac.pa

Fernando X. Arias
Technological University of Panama, Panama, Panama, fernando.arias@utp.ac.pa

Carlos A. Medina C.
Technological University of Panama, Panama, Panama, carlos.medina@utp.ac.pa

Abstract

Compressed sensing (CS) is a rapidly growing field, attracting considerable attention in many areas from imaging to communication and control systems. This signal processing framework is based on the reconstruction of signals, which are sparse in some domain, from a very small data collection of linear projections of the signal. The solution to the underdetermined linear system, resulting from these data, allows us to estimate the original signal. Finding the solution is an optimization problem which is mainly based on the minimization of the l_1-norm. For this purpose, fast algorithms have been developed, and the aim of this paper is to make a comparative analysis of some of these algorithms. Specifically, we study the performance of sparse reconstructions with five commonly used algorithms implemented in Matlab (CVX, L1magic, SPGL1, UnlocBoX and YALL1) in order to determine the viability of implementation of a large number of applications proposed in the last years using CS. The objective is to offer a reference analysis for algorithm selection in CS applications and also provide a methodology for algorithm analysis in sparse reconstruction. Consequently, in order to determine each algorithm's effectiveness and being able to compare them in a standardized manner, three indicators are considered: *execution time*, *percent error* and *CPU usage*, being the first two indicators crucial to any application involving compressed sensing. In addition some general basic concepts on compressed sensing are included.

Keywords: compressed sensing, signal reconstruction, l_1-norm minimization.

Resumen

La detección con datos dispersos (CS - compressed sensing) es un campo de rápido crecimiento, que atrae considerable atención en muchas áreas, desde la producción de imágenes hasta sistemas de comunicación y control. Esta técnica de procesamiento de señales está basada en la reconstrucción de señales, que tienen una representación dispersa en algún dominio, a partir de una cantidad muy pequeña de datos correspondientes a proyecciones lineales de la señal. La solución al sistema lineal subdeterminado, resultante de los datos, permite estimar la señal original. Encontrar la solución es un problema de optimización basado principalmente en la minimización de la norma l_1. Para este propósito, se han desarrollado algoritmos rápidos, y el propósito de este trabajo es realizar un análisis comparativo de algunos de estos algoritmos. Específicamente, se estudia el desempeño de reconstrucciones con datos dispersos con cinco algoritmos comúnmente utilizados implementados en Matlab (CVX, L1magic, SPGL1, UnlocBoX y YALL1), con el fin de determinar la viabilidad de la implementación de una gran cantidad de aplicaciones propuestas en los últimos años usando CS. El objetivo es ofrecer un análisis de referencia para la selección de algoritmos en aplicaciones de CS y también proveer una metodología para el análisis de algoritmos para reconstrucciones con datos dispersos. Por consiguiente, para determinar la eficacia de cada algoritmo y poder compararlos de manera estandarizada, se consideran tres indicadores: tiempo de ejecución, porcentaje de error y uso del CPU, siendo los dos primeros indicadores

cruciales para cualquier aplicación de detección con datos dispersos. Además, se han incluido algunos conceptos básicos generales sobre detección con datos dispersos.

Palabras claves: detección de señales dispersas, reconstrucción de señales, minimización de la norma l_1

1. INTRODUCTION

Compressed sensing or compressive sensing (CS) is a signal processing framework for efficiently sensing and reconstructing a signal. This novel technique reconstructs the original signal via optimization from a relatively small amount of measurements opposite to the traditional methods in signal processing, in which the whole information is collected and then compressed. CS is based on the knowledge that a small collection of linear projections of a sparse signal in some domain contains enough information for its recovery. Thus, there is a potentially large reduction in the sampling and computation requirements of a signal detection or estimation system using CS, which translates into decreasing of energy, bandwidth, and processing time needs (Duarte et al., 2008) (Zambrano et al., 2010) (Fazel et al., 2011). CS is based on the work of Candès, Romberg and Tao (Candès & Romberg, 2005b) (Candès et al., 2006a) and Donoho (Donoho, 2006), who showed that a signal having a sparse representation in some domain can be reconstructed from a small set of non-adaptive, linear measurements if they meet some criteria. CS can be used, not only for signal reconstruction or approximation, but also in making a detection or classification decision or estimation, which makes the CS technique a very powerful approach to many applications and to a wide range of statistical inference tasks (Davenport et al., 2006).

Recently, a large number of signal reconstruction algorithms based on l_1 minimization have been developed as a response to the scientific community and the industry's need for increasingly processor-efficient algorithms for CS. These algorithms exploit some signal sparsity characteristics to reconstruct it from a very small number of samples through the minimization of the l_1-norm (Candès et al., 2005b). Thus, finding a minimum l_1-norm solution to an underdetermined linear system is an important issue in compressed sensing and has lately received a significant attention in the literature (Prakash et al., 2014) (Ai et al., 2014) (Poli et al., 2013) (Viani et al., 2013) (Yang et al., 2013) (Combettes, et al., 2010). This is equivalent to a linear programming, and a broad variety of solution algorithms has been developed, some of them are considered in this work.

When developing an application that employs principles of CS, usually the developer uses reconstruction algorithms that are either popular or familiar to him. This practice leads to not always considering the differences between available algorithms and choosing the most efficient one for a given type of signal or application. Therefore, to study the performance of sparse reconstructions with different algorithms is important in order to determine the viability of implementation of a large number of applications proposed in the last years in this field (Kuriyakose et al., 2013) (Osamy et al., 2013) (Fazel et al., 2011) (Zambrano et al., 2010). The objective is to offer a reference analysis for algorithm selection in CS applications and also provide a methodology for algorithm analysis in sparse reconstruction. In order to accomplish this, we compare the performance of some algorithms for sparse signal reconstruction. Consequently, to determine each algorithm's effectiveness and being able to compare them in a standardized manner, we settled on two main indicators that are crucial to any application involving compressive sensing: *execution time* and *percent error*. In addition, the CPUusage is also considered.

Specifically, we focus in five commonly used algorithms for signal reconstruction in Matlab: CVX (Grant & Boyd, 2013) (Grant & Boyd, 2008), L1magic (Candès & Romberg, 2005a), SPGL1 (Van Den Berg et al., 2007), UnlocBoX (Combettes and Pesquet, 2010) and YALL1 (Zhang et al., 2010). Related to this work there are some publications worth mentioning: (*i*) (Lai et al., 2012) – the authors propose a heuristic optimality check as a general tool for l_1-minimization, and provide a numerical comparison of various l_1-solvers, using a test set with a variety of explicitly given matrices and several right hand sides per matrix reflecting different levels of solution difficulty. They analyzed the results, as well as improvements by the proposed heuristic optimality check to provide an answer to the question which algorithm is the best. (*ii*) (Lorenz et al., 2013) - the authors propose a Projected Proximal Point Algorithm (ProPPA) for solving a class of optimization problems, also provided a convergence analysis theoretically supporting the general algorithm, and then applied it for solving l_1-minimization problems and the matrix completion problem, and compared their proposal with other existing algorithms using numerical experiments, which include some of the algorithms considered in our work. Neither the heuristic optimality check nor the ProPPA have been considered in the present analysis. The comparative analysis presented in these two papers is similar in some aspects: use of various degrees of sub-sampling of the data and use of time and accuracy as comparison quantities. On the other hand, we consider an additional set of commonly used algorithms, and the tests are run for an image reconstruction problem. The test parameters used allow application developers to make a decision based on performance information when choosing a reconstruction algorithm that will allow for computational and/or time constraints (for a specific application) to be met and provide the best possible solution.

In the following sections some signal features and basic concepts on Compressed Sensing and l_1-norm are discussed, followed by the description of the five algorithm considered and the tests performed. Then, comparison of numerical results of the reconstruction process for the five algorithms is discussed, and conclusions based on them are provided. The results are analyzed using different scenarios which have real time response or fidelity of the reconstructed signal as goals.

2. Basic Concepts of Compressed Sensing

Compressed sensing theory mainly exploits two properties of signals that allow them to be subsampled and reconstructed at such great fidelity: *sparsity* and *compressibility*. A signal with a high *sparsity* can be accurately represented using a very small number of nonzero coefficients. The greater the sparsity of the signal, the less varied in magnitude its nonzero coefficients are among themselves and less coefficients are necessary to accurately represent it. Similarly, a *compressible* signal is a signal that can be approximated from a sparse signal with a high degree of precision. In other words, a compressible signal, when transformed, contains most of its energy within a very limited number of coefficients in some basis.

Following (Baraniuk, 2007) let $\mathbf{x}$ be a real-valued, finite-length, one-dimensional, discrete-time signal, which can be viewed as an $N \times 1$ column vector in R^N with elements $x[n]$, $n = 1, \ldots, N$. Any signal in R^N ca be represented in terms of a basis of $N \times 1$ vectors $\{\psi_i\}$, $i = 1, \ldots, N$. For simplicity, assume that the basis is orthonormal. Thus, the signal $\mathbf{x}$ can be expressed as a linear combination of the N basis vectors with their correspondent coefficients $\mathbf{s}$, $\mathbf{x} = \Psi\mathbf{s}$, where $\mathbf{s}$ is the $N \times 1$ column vector of weighting coefficients $s_i = \langle \mathbf{x}, \psi_i \rangle = \psi_i^T \mathbf{x}$, where $\cdot^T$ denotes transposition. If the signal $\mathbf{x}$ is K-sparse, then $\mathbf{x}$ is a linear combination of only K basis vector, that is, only K of the s_i coefficients are nonzero and $(N - K)$ are zero. For CS the interest is in the case when $K << N$, for it is $\mathbf{x}$ a compressible signal. Then, it is possible to have a linear measurement process that computes $M < N$ inner products between $\mathbf{x}$ and a collection of vectors $\{\phi_j\}$, $j = 1, \ldots, M$ as in $y_j = \langle \mathbf{x}, \phi_j \rangle$. Arranging the measurements y_j in an $M \times 1$ vector $\mathbf{y}$ and the measurement vectors ϕ_j^T as rows in an $M \times N$ matrix Φ, we can write

$$\mathbf{y} = \Phi\mathbf{x} = \Phi\Psi\mathbf{s}$$

The basis Φ are fixed and does not depend on the signal $\mathbf{x}$. Then the problem consist of designing (*i*) a stable measurement matrix Φ such that the significant information in any K-sparse signal is not damaged by the dimensionality reduction from $\mathbf{x} \in R^N$ to $\mathbf{y} \in R^M$ and (*ii*) a reconstruction algorithm to recover $\mathbf{x}$ from only $M \approx K$ measurements y. The first condition is not of our concern in this paper; nevertheless, it is worthy to indicate that the condition is satisfied if the matrix has the *restricted isometry property* (RIP) and a related condition, referred to as *incoherence* (Candès et al., 2006a) (Candès et al., 2006b) (Donoho, 2006). As for the signal reconstruction algorithm, it must take the M measurements in the vector $\mathbf{y}$, the matrix Φ, and the basis Ψ and reconstruct the signal $\mathbf{x}$ of length N, or equivalently, its sparse coefficient vector $\mathbf{s}$. This task involves solving an undetermined matrix equation since the number of compressed measurements taken is very small. The constraint that the signal is sparse enables the solution of this underdetermined system of linear equations.

The least –squares solution to such inverse problems of this type is to minimize the l_2-norm; that is, to minimize the energy in the system. Unfortunately, minimization of the l_2-norm almost never finds a K-sparse solution and leads to poor results in practical applications. Instead of using the l_2-norm, which measures signal energy, one can minimize the number of nonzero components of the solution, given the signal sparcity. For this, one can consider an l_0-norm that counts the number of non-zero entries in the vector. Nevertheless, solving this optimization problem is numerically unstable and NP-complete (Baraniuk, 2007). Finally, it was proved that for many problems, optimization based on the l_1-norm, given by

$$\mathbf{s}' = \text{argmin } ||\mathbf{s}||_1 \text{ such that } \mathbf{y} = \Phi\Psi\mathbf{s}$$

is equivalent to the l_0-norm, and allows exact recovery of K-sparse signals and closely approximate compressible signals with high probability (Candès et al., 2006a) (Candès at al., 2006b) (Donoho, 2006). This optimization is much easier to solve as a linear program known as "basis pursuit", for which efficient solution methods exist.

3. ALGORITHMS, EVALUATION AND ANALYSIS

In this section, the algorithms considered are briefly described, a discussion on the evaluation method and parameters taken into account are indicated, and finally the analysis of the results from the experimental tests is given.

3.1 ALGORITHMS

The algorithms compared in this work are the following:

CVX - V2.0, December 2013 - CVX is a Matlab-based modeling system for convex optimization, which makes it easy to construct and solve disciplined convex programs, including l_1 norms. While not specifically aimed at compressive sensing applications, the algorithm provides a way to perform compressive sensing techniques on signals through l_1 minimization.

L1magic - V1.1, April 2011 - Unlike previously mention CVX, L1-MAGIC is a collection of MATLAB routines specifically geared towards signal reconstruction in compressive sensing applications through convex optimization.

SPGL1 - V1.8, May 2013 - SPGL1 is a MATLAB solver which features an l_1 minimization algorithm, which makes it very useful for compressive sensing applications. The code runs natively on MATLAB and relies entirely on vector and matrix operations. This makes it less dependent on varying compiler and system configurations, and therefore, theoretically, provides more stable results along a wider variety of system configurations than algorithms written in other programming languages.

UnlocBoX - V1.2.113, December 2013 - UnlocBox was developed with the objective of becoming a modular, general use convex optimization toolbox for MATLAB that is easily extensible for particular applications. Another of the design objectives was to develop a toolbox that provided repeatable results, which are often difficult to obtain due to the iterative nature of the mathematical processes for performing l_1 minimization on a given signal, which often yield different results when reconstructing a given signal.

YALL1 - V1.3, December 2013 - Your ALgorithms for L1 minimization is a simple MATLAB solver capable of solving eight different l_1 minimization problems. It is designed to require very little user input beyond the specific signal to be processed by the algorithm. However, the solver provides the ability to fine tune a set of customizable parameters for applications that require a specific algorithm behavior, or are otherwise limited.

3.2 EVALUATION

In order to evaluate the performance of the algorithms, several tests were run on a 2GHz T6400 Intel Core2 Duo machine with 4 Gigabytes of DDR2 memory running Windows 7 64-bit operating system. All algorithms were provided for use in MATLAB and evaluated on version R2012a.

We compare the performance in execution time, global error and CPU usage as the percentage of measured samples used in the reconstruction increases. To comparatively test each algorithm's performance, we normalized the evaluation parameters to minimize unnecessary variations in the results.

The execution time is measured starting from the moment after the image signal to be sampled and reconstructed is loaded into memory and ready to be processed until the moment it arrives at a satisfactory solution and is stores in a variable. The reconstruction percent error is calculated by measuring and averaging, pixel by pixel, the difference in magnitude between the input and output signals' coefficients.

The base test signal is a 1024×1024 pixel image shown in Figure 1. Its size, sparsity level and distinct shape make it a very convenient visual aid for the effectiveness of the algorithm in use, even when shrunk to sizes below 32×32 pixels for faster evaluation times. These samples are shown in Figure 2.

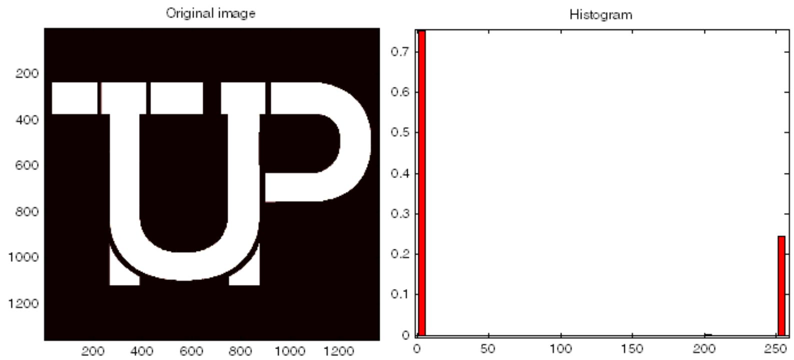


Figure 1: 1024×1024 Image signal designed for reconstruction algorithm analysis and its data histogram.

a) 8×8 image b) 16×16 image c) 32×32 image

Figure 2: Sample image signals used to test the effectiveness of the reconstruction algorithms.

In order to measure each algorithm's effectiveness and efficiency along a wide variety of image signals and operating parameters, we compared the reconstructions of the three image signals presented in Figure 2 when sampled at 15, 30, 45, 60, 75 and 100 percent of its total length. The selected samples are taken considering an i.i.d. standard Gaussian distribution. The tests performed at 100% of the entire length of the given signals have the objective of experimentally determining the inherent reconstruction error introduced by each algorithm, even in scenarios where it is provided with all the signal information and have to perform no actual reconstruction.

We also measured the time required to perform each reconstruction by each algorithm. Execution times and percent errors were measured multiple times and averaged for each case, to eliminate noise in our measurements.

3.3 ANALYSIS

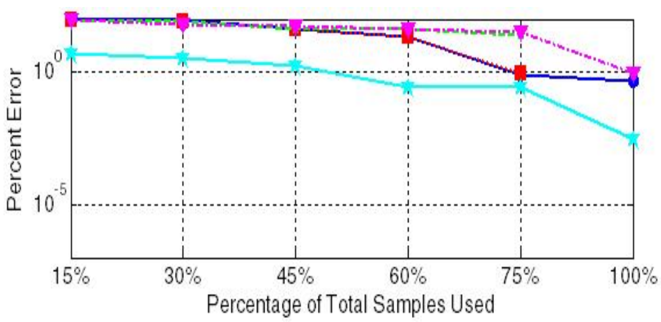
In the following, results from tests using different image sizes and amount of samples are analyzed, comparing the reconstruction percent error, reconstruction effectiveness and reconstruction time for the indicated algorithms.

Figure 3 shows the reconstruction percent error for the chosen algorithms. For the reconstruction of an 8×8 image, all algorithms exhibited roughly the same behavior when sampling below 60% of the total signal information. YALL1 is the exception in the group, and performed consistently better than the rest of the algorithms at reconstructing the 8×8 sample.

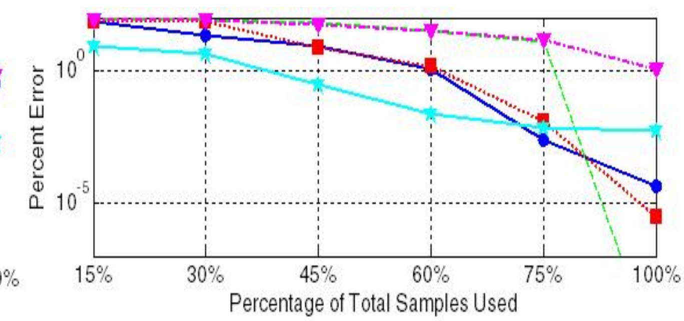
Due to additional data, reconstruction efficacy shows a greater variation among the selected algorithms for 16×16 and 32×32 images, with YALL1 still showing a substantial advantage over the rest of the chosen algorithms when working under the sampling percentages commonly used in compressive sensing applications (15 to 60 percent). When sampling the signal above 60%, L1Magic and CVX were among the algorithms with the least errors.

For SPGL1, UnLocBox and YALL1, the reconstruction error introduced by the algorithm when sampling 100% of the signal length seems unaffected by the size of the signals to be processed. This, however, was not the case with the rest of the algorithms. CVX and L1Magic, as shown in Figure 3, show considerably reduced error when sampling 100% of the signal at larger image sizes.

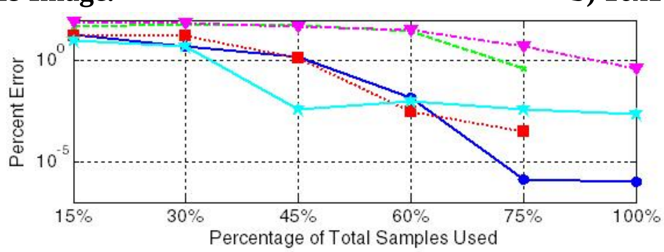
All these finding in this numerical experiment are really important because represent what we can expect when the fidelity of the reconstructed signal is the major objective in the final application. In the case when the speed of processing is an issue like in real time applications, we propose to study the execution time for each algorithm as the size of the reconstructed images increase as well the initial percentage of measurements used to solve the optimization problem.



a) 8x8 Image.



b) 16x16 Image.



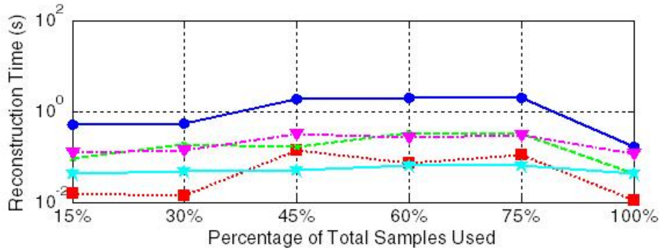
c) 32x32 Image.

Figure 3: Reconstruction error of algorithms for different image sizes.

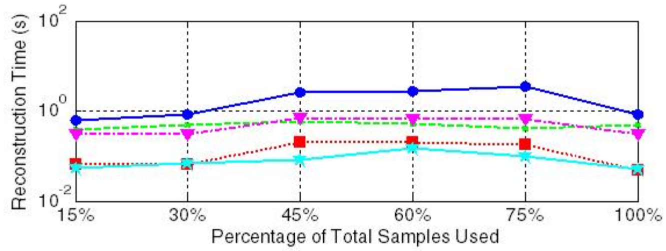
Figure 4 shows the reconstruction time for the chosen algorithms. When working with an 8×8 image, L1Magic and YALL1 were the fastest algorithms, for all sampling percentages, to deliver a reconstructed signal.

With the burden of additional data that comes with 16×16 and 32×32 images, processing times rise for all schemes except for YALL1, and the rate at which these increase must be taken into account when choosing an algorithm if the application requires the reconstruction of 32×32 or larger images. Furthermore, for 32×32 and larger images, the reconstruction times exhibited specifically by CVX become a matter that must be seriously considered for its implementation in any given application, as shown in Figure 4.

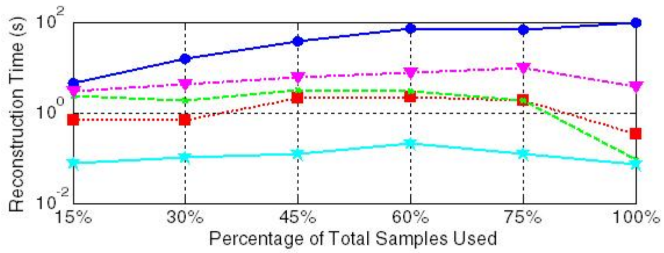
In computationally terms, even though the hardware (extra memory) needed for an application or system is cheap, it is necessary to know how much hard drive memory is used by each algorithm for reconstruction of the signal in order to predict the extra memory needed in the system if others functions or processes have to coexist. This could be really important in some applications which manage high volume of data. For this numerical test we measured the specific CPU usage by the MATLAB process in each algorithm by using the Microsoft Performance Monitor, included by default with every Microsoft Windows 7 installation. This tool allows us to differentiate the specific algorithm's CPU consumption from other processes.



a) 8x8 Image.



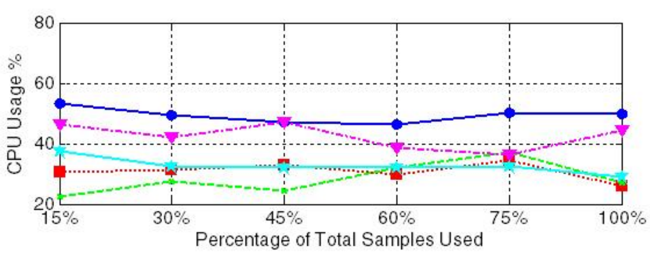
b) 16x16 Image.



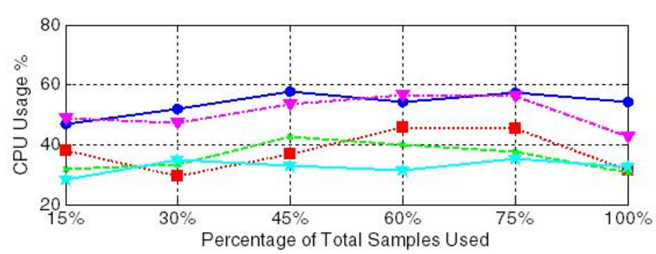
c) 32x32 Image.

Figure 4: Reconstruction time of algorithms for different image sizes.

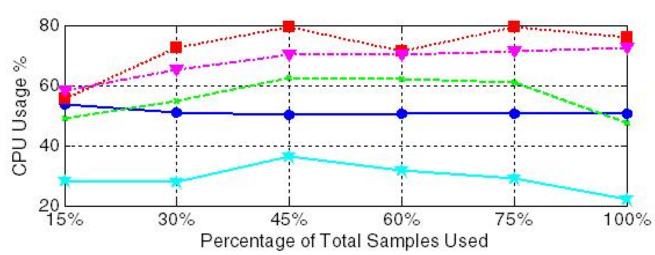
Figure 5 shows the plots of the CPU usage in our particular platform. Nevertheless, similar results are expected in other platforms. We observe that L1 magic present the behavior showing a high rate of increment in the percentage of CPU usage as the size of the reconstructed signal increases. SPGL1 and UnLocBox present similar behavior but less pronounced. Contrarily, CVX and YALL1 keep almost the same percentage of CPU usage within a constant range. Therefore it seems that CVX and YALL1 are not affected by the size of the signal to reconstruct. For all the algorithms we found that depending of the initial amount of measurements used in the reconstruction the percentage changes and does not show a uniform monotonic behavior.



a) 8x8 Image.



b) 16x16 Image.



c) 32x32 Image.

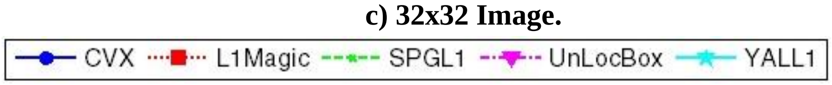


Figure 5: CPU usage of algorithms when reconstructing different image sizes.

Finally, we can make the following observations about each one of the algorithms evaluated according to the obtained results:

CVX - Yields comparatively low reconstruction errors at all image sizes (second best to YALL1). Reconstruction errors decrease with image size. However, reconstruction times are the slowest among all algorithms tested, and may prove to be inconvenient for applications that employ images above 32×32.Algorithm introduces very low reconstruction error when sampling 100% of the provided signal.

L1Magic - While reconstruction errors are greater than CVX, L1Magic still yields comparatively low reconstruction errors at all image sizes. Reconstruction errors decrease with image size. Unlike CVX, reconstruction times are among the lowest of all algorithms (second best to YALL, on average). Algorithm introduces almost no reconstruction error when sampling 100% of the provided signal.

SPGL1 - Together with UnLocBox, SPGL1 yields the highest reconstruction error of all the algorithms evaluated. Reconstruction error is relatively constant among all image sizes tested for all signal portions sampled. Out of all the algorithms evaluated, SPGL1 has the third highest average reconstruction time across all image sizes considered.SPGL1 introduces no reconstruction errors when provided with a sample corresponding to 100% of the signal.

UnLocBox -Relative to all other algorithms evaluated, UnLocBox yields the highest reconstruction error. The reconstruction errors do not seem to be affected by image size. Reconstruction time is the second highest of all the algorithms, but still proves to be considerably faster than CVX. The UnLocBox algorithm introduces the highest reconstruction error of all the evaluated algorithms, even when provided with a sample corresponding to 100% of the signal.

YALL1 - Yields the lowest reconstruction error at small image sizes and small samples by a considerable margin. At bigger sample percentages, other well-performing algorithms such as CVX and L1Magic match its reconstruction error output. The key feature is its reconstruction speed, which is, on average, considerably lower than the rest of the evaluated algorithms across all image sizes. Because of its low reconstruction times, YALL1 might prove to be the ideal choice for applications that require reconstruction of large images, Out of all the evaluated algorithms, YALL1 yields the fourth lowest reconstruction error when provided with the entirety of the signal.

4. CONCLUSIONS

This work compares several popular algorithms implemented in Matlab for solving the l_1-norm minimization problem applied to a sparse signal reconstruction. Specifically, this paper presents a comparative analysis by analizing the efficiency of common algorithms via execution time, percent error and percentage of CPU usage. Several experimental comparisons are reported to validate efficiency and advantage of the algorithms. This can help to select the right algorithm for different practical purposes, and even could help to further improve the available implementations.

This comparative analysis for algorithms in CS applications provides a good tool to analyze algorithms for sparse reconstruction when we have different scenarios. Specifically, we present our findings for applications that have real time and/or high fidelity signal requirements. Also, by considering reconstructed signals that have been sub-sampled to various degrees, allows us to determine both a threshold for each algorithm above which the approximated signal is deemed as satisfactory, and the signal noise introduced by each algorithm. Possible future work includes running the tests presented within this document for additional algorithms and for a variety of larger sizes of signals. Also, complex data and matrices, which prove to be useful for a larger number of potential applications, could be included in the analysis.

REFERENCES

Ai, A., Lapanowski, A., Plan, Y., & Vershynin, R. (2014). "One-bit compressed sensing with non-Gaussian measurements". *Linear Algebra and its Applications*, 441, pp 222-239.

Baraniuk, R. (2007). “Compressive sensing”, *IEEE Signal Processing Magazine*, Vol. 24, No. 4, pp 118-120.
Duarte, M., Davenport, M., Takhar, D., Laska J., Sun, T., Kelly, K., and Baraniuk, R. (2008). “Single-pixel imaging via compressive sampling”, *IEEE Signal Processing Magazine*, Vol. 25, No. 2, pp 83-91.
Candès, E., & Tao, T. (2005). “Decoding by linear programming”, *IEEE Transactions on Information Theory*, 51(12), pp 4203-4215.
Candès, E., & Romberg, J. (2005a). L1-magic: Recovery of sparse signals via convex programming. URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf, 4.
Candès, E., & Romberg, J. (2005b). “Practical signal recovery from random projections”, Draft, From: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.1551&rep=rep1&type=pdf.
Candès, E., & Tao, T. (2006). “Near-optimal signal recovery from random projections: Universal encoding strategies”, *IEEE Transactions on Information Theory*, 52 (12), pp 5406-5425.
Candès, E., & Romberg, J. (2006). “Quantitative robust uncertainty principles and optimally sparse decompositions”. *Foundations of Computational Mathematics*, 6 (2), pp 227-254.
Candès, E., Romberg, J., & Tao, T. (2006a). “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”, *IEEE Transactions on Information Theory*, 52(2), pp 489-509.
Candès, E., Romberg, J., & Tao, T. (2006b). “Stable signal recovery from incomplete and inaccurate measurements”, *Communications on pure and applied mathematics*, 59 (8), pp 1207-1223.
Combettes, P., and Pesquet, J. (2010). “Proximal Splitting Methods in Signal Processing”, arXiv:0912.3522v4 [math.OC]
Davenport, M., Wakin, M., & Baraniuk, R. (2006). Detection and estimation with compressive measurements. Technical Report No. TREE 0610, Rice University, Dept. of ECE.
Donoho, D. L. (2006). “Compressed sensing”, *IEEE Transactions on Information Theory*, 52(4), 1289-1306.
Fazel, F., Fazel, M., & Stojanovic, M. (2011). “Random access compressed sensing for energy-efficient underwater sensor networks”. *IEEE Journal on Selected Areas in Communications*, 29(8), 1660-1670.
Grant, M. & Boyd, S. (2013) CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, September 2013
Grant, M. & Boyd, S. (2008) Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, pages 95-110, Lecture Notes in Control and Information Sciences, Springer, 2008. http://stanford.edu/~boyd/graph_dcp.html
Kuriyakose, N., DivakaraMurthy, H. S., & Jose, A. (2013). “Energy efficiency analysis of Compressed Sensing Video Streaming for Wireless Multimedia Sensors”. IJRCCT, 2 (8), pp 667-672.
Lai, R., and Yuen, P. (2012). “ProPPA: a fast algorithm for l_1 Minimization and Low-rank Matrix Completion” Cornell University Library, submitted, arXiv.org/cs/arXiv:1205.0088
Lorenz, D., Pfetsch, M., and Tillmann, A. (2013). “Solving Basis Pursuit: Heuristic Optimality Check and Solver Comparison”, Optimization Online, URL: http://www.optimization-online.org/DB_HTML/2011/07/3100.html
Osamy, W., Salim, A., & Aziz, A. (2013). “Efficient Compressive Sensing based Technique for Routing in Wireless Sensor Networks”, *INFOCOMP Journal of Computer Science*, 12 (1), pp 1-9.
Perraudin, N., Shuman, D., Puy, G., & Vandergheynst, P. (2014). UNLocBoX A matlab convex optimization toolbox using proximal splitting methods. arXiv preprint arXiv:1402.0779.
Poli, L., Oliveri, G., Rocca, P., & Massa, A. (2013). “Bayesian compressive sensing approaches for the reconstruction of two-dimensional sparse scatterers under TE illuminations”, *IEEE Transactions on Geoscience and Remote Sensing*, 51(5), 2920-2936.
Prakash, J., Dehghani, H., Pogue, B., & Yalavarthy, P. (2014). “Model-Resolution based Basis Pursuit Deconvolution Improves Diffuse Optical Tomographic Imaging”. *IEEE Trans. on Medical Imaging*.
Van Den Berg, E., & Friedlander, M. (2007). SPGL1: A solver for large-scale sparse reconstruction. Online: http://www. cs. ubc. ca/labs/scl/spgl1.
Van Den Berg, E., and Friedlander, M. (2008). “Probing the Pareto frontier for basis pursuit solutions”, *SIAM J. on Scientific Computing*, Vol. 31 No. 2, pp 890-912.
Viani, F., Oliveri, G., & Massa, A. (2013). “Compressive sensing pattern matching techniques for synthesizing planar sparse arrays”, *IEEE Trans. on Antennas and Propagation*, Vol. 61, No. 9, pp 4577–4587.
Yang, Z., Zhang, C., & Xie, L. (2013). “Robust compressive phase retrieval via l_1 minimization with application to image reconstruction”, arXiv preprint arXiv:1302.0081.

Zambrano, M., Marengo, E., & Fisher, J. (2010). “Coherent single-detector imaging system”, *IEEE Workshop on Signal Processing Systems* (SIPS), pp. 111-115.

Zhang Y., Yang, J., and Yin, W. (2010). “User's Guide for YALL1: Your ALgorithms for L1 Optimization”, Versions: 1.0. June 4, 2010, CAAM Technical Report TR09-17.

Authorization and Disclaimer