A Comparison of Basis Pursuit, Orthogonal Matching Pursuit, and Subspace Matching Pursuit and their Performance Differences

Pengxiang Hu,Robert Lewis
Department of Computer Science
Tufts University
Medford, MA 02155
pengxiang.hu@tufts.edu; Robert.Lewis@tufts.edu

Abstract

This paper aims to discuss different compressive sensing techniques such as Basis Pursuit, Orthogonal Matching Pursuit, and Non-convex Compressive Sensing, including their theoretical differences and performance difference in terms of accuracy and speed during algorithm simulation in MATLAB program.

1 Introduction

Ever after David Donoho coined the term "Compressed sensing" (or "Compressive Sensing") (1), the research in the field of Compressive Sensing has exploded in terms of both the theoretical research and practical utilization of this technique. In Emmanual Candés and Terence Tao's paper in 2006 (2), they proposed l_1 minimization of the equation $A * x = y$ where A is the sampling matrix, x is the sparse signal vector, and y is the observation vector. By minimizing the l_1 norm of signal vector x, they proposed that one can recover most of the signal information with very little sampling. However this is not the only technique. Joel Tropp and Anna Gilbert proposed a iterative method, Orthogonal Matching Pursuit, which utilized a different algorithm to recover the signal with different performance features (3).

1.1 The Problem

Compressive Sensing in general tries to solve the equation $A * x = y$ where A is an $M * d$ matrix with $M << d$. Since M is much smaller than d, y has much smaller dimension than x and thus cannot be used to find x exactly. However, with the information of y we can solve via pseudo-inverse and find a large space of vectors, x such that, $A * x = y$. Conventionally we try to find the smallest such x over some norm, ideally the l_0 norm.

2 The CS Aproach

Inverting Matrices is a computationally costly task. Compressive Sensing techniques seek to solve the problem in other ways. Compressive Sensing considers only the case where x, the vector being solved for, is sparse. Then the problem can be solved by finding the collumns of A over which y is a linear combination, otherwise known as the support of y (4).

2.1 OMP

Orthogonal Matching Pursuit, or OMP, is an algorithm developed by Tropp (3). It uses a greedy approach to determine the support of y. For a S-sparse input signal, x, it has $O(S\log(d)$ complexity

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

in the worst case and can recover signals with very high confidence. OMP tries tominimize x over the l_1 norm since minimizing over the l_0 norm is NP-HARD (5). OMP takes $O(S \ln d)$ time. It needs $N \geq QS \log(\frac{d}{\delta})$ measurements for $P_{err} < 2\delta$

2.2 Basis Pursuit

Basis Pursuit tries to solve the same L_1 optimization problem. Basis Pursuit uses a dictionary based approach where the signal is approximated via its sparsest representation by dictionary elements. Linear programming methods, usually simplex method, are used to solve. It is a polynomial time algorithm. Sometimes Basis Pursuit refers to the general l_1 minimization problem discussed in this paper (6).

2.3 Non-Convex

Non-Convex optimization tries to solve $A * x = y$ for the smallest x according to an l_p norm for $0 < p < 1$. Using non-convex optimization, sparse signals can be recovered with fewer measurements than l_1 optimization techniques (7).

3 Performance Comparison of Basis Pursuit, OMP, and Non-Convex Optimization

In this section, we are going to examine the average accuracy and run time performance of each Compressive Sensing algorithm.

3.1 Methodology

There are three major inputs for all algorithms, the sparse vector x, the sampling matrix A, and the observation vector y. To be more specific, the number of measurements m (a.k.a the number of rows in A) determines how much information will be observed in vector y and sparsity of signal x determines the complexity of the signal of interest.

Thus, for accuracy performance comparison, we are going to plot the change of measurements m relative to the change of sparsity S, to see how much measurements does each algorithm need to recover different sparse signals. Note the maximum number of measurements is capped at the length of signal x, because one only need a N by N linearly independent matrix to determine a vector x of size N.

Moreover, we are going to include Monte Carlo Simulation technique to simulate each reconstruction algorithm for 50 times to balance out accidental random small probability events.

3.2 Basis Pursuit

For Basis Pursuit, we just specify the sparsity S of signal x, the number of measurements m of a random Gaussian matrix A, and we can get $b = A * x$. The specific implementation algorithm is achieved through the CVX MATLAB toolbox written by Stephen Boyd from Stanford University.

As you can see the performance figure above shows a close-to-linear relationship between the accuracy minimization of different sparsities and the number of measurements. In general, more measurements can recover better signal. We should also note that with $m \geq 2 * S$ the Basis Pursuit can already recover signal with very high accuracy. However, the run time is 159 seconds which takes way too long. I think it has to do with how CVX toolbox is written because every time I call *cvx_begin* and *cvx_end*, the function would print out all the iteration of ℓ_1 minimization which slows down the program significantly.

We can circumvent this dilemma by utilizing a different toolbox ASP (8) written by Michael Friedlander and Michael Saunders. This toolbox implements the ℓ_1 minimization problem through a faster approach

$$\text{minimize}_x \quad \lambda ||x||_1 + \frac{1}{2}||Ax - b||_2^2$$

covered by David Donoho in his paper (9) from 2007.

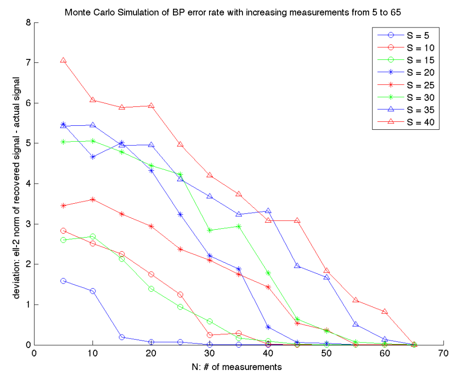

Figure 1: Basis Pursuit with CVX solving $\ell_1 minimization$ with MATLAB run time 159 seconds

3.3 Orthogonal Matching Pursuit

Note that *Theorem 2* of (3) states that if we fix $\delta \in (0, 0.36)$ and $m \geq cS\ln(N/\delta)$, where N is the number of columns in sampling matrix, c is a constant and can be reduced to 4 for large S, OMP can reconstruct the signal with probability exceeding $1 - 2\delta$. First of all, this equation leaves us with great relaxation area: when we set c = 4, N = 64, with the minimum m required, we can recover the signal with only N measurements but this equation went over and required 800 measurements, which is a huge waste of running time (shown in Figure 2). Secondly, we are notj interested in δ, we just need to pick a reasonable δ and keep it consistent during run time.

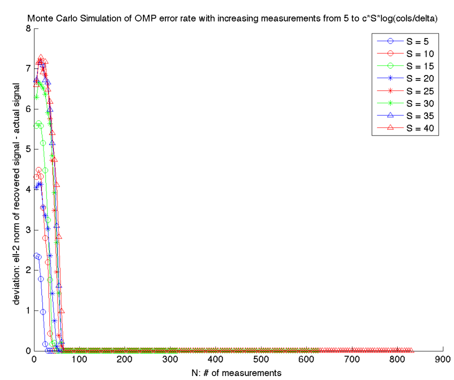

Figure 2: OMP with $\delta = 0.36$

The actual result with measurements capped at N is shown in Figure 3. As you can see, as the number of sparsity increases, it generally takes more measurements to ensure low error rate. One thing to note is that with 35 measurements, the sparsity 5 and 10 have very similar accuracy performance.

3.4 Non-convex Compressive Sensing

Non-convex Compressive Sensing technique is arguing that with smaller support (sparsity), this iterative reweighted algorithm can take fewer measurements than convex optimization to produce exact reconstruction of the original signal. This technique may be really useful with prior knowledge of the sparsity of the signal of interest, but because we are only comparing the performance of the numerical experiments. We can just set $p = 1$ and 0.5 as benchmarks for this algorithm and see if its iterative approach changes any aspects of result.

As shown in Figure 4 and Figure 5, test with $p = 1$ has more accuracy in almost all sparsity test and is faster in run time as well. I'm guess that this is because the sparsity isn't small enough to show the

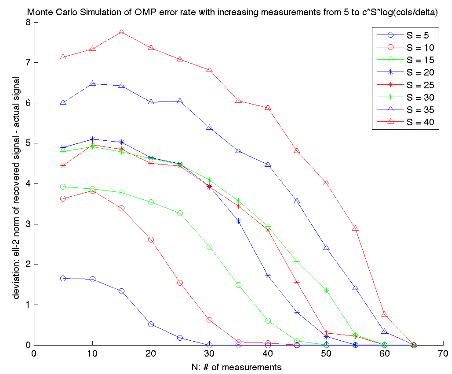

Figure 3: OMP with $\delta = 0.36$. MATLAB run time is 9.7468 seconds

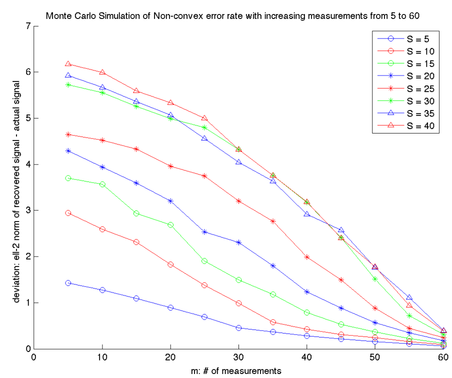

Figure 4: Non-convex sensing with p = 1, Epsilon = 10^{-3} MATLAB run time is 21.0862 seconds

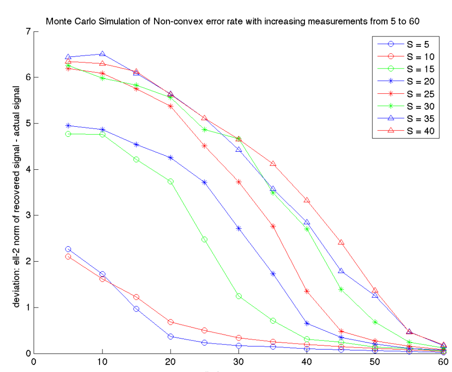

Figure 5: Non-convex sensing with p = 0.5, Epsilon = 10^{-3} MATLAB run time is 28.4333 seconds

computational advantage of this greedy algorithm. Thus let's just use the performance of non-convex compressive sensing with $p = 1$ for comparison.

Moreover, because the way the algorithm is written, *lp_re.m* has a threshold controller *epsilon* which determines if this greedy algorithm needs to run more optimization loops before returning the recovered signal. This gives us control to trade off running time with the accuracy.

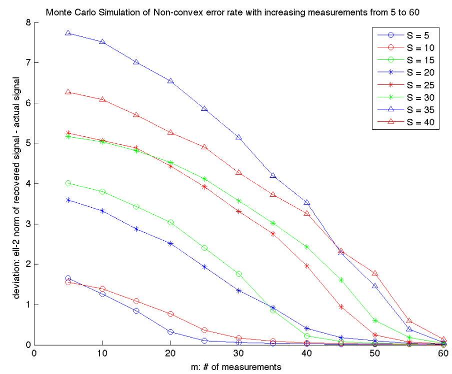

Figure 6: Non-convex sensing with p = 1, Epsilon = 10^{-5} MATLAB run time is 80.8334 seconds

4 Discussion and Conclusion

As you can see above, in terms of the general accuracy, Basis Pursuit is higher than Orthogonal Matching Pursuit. From Figure 1, the shape of BP reconstruction result is close to linear regression but Figure 3 indicated that OMP has a concave shape, and when the signal sparsity is $S \leq \frac{N}{3}$, BP performs unanimously better than OMP.

One can infer that with a Gaussian measurement matrix, BP can, with high probability, recover all sparse signals. In the same setting, OMP recovers each sparse signal with high probability but with high probability fails to recover all sparse signals. One may also infer the latter statement from [J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation, Theorem 3.10] along with a somewhat involved probability estimate.

As paper (3) points out, since OMP is inherently more difficulty to analyze than BP. Right now, we understand the stability of BP much better than the stability of OMP. More research in this direction would be valuable.

Thus, it's safe to say that Greedy algorithms have advantages when we are concerned about computational cost. In certain cases, OMP is faster than standard approaches for completing the minimization (BP). However, when the signal is not very sparse, OMP may be a poor choice because the cost of orthogonalization increases quadratically with the number of iterations.

When we compare the non-convex compressive sensing technique to BP, Figure 4 shows that with small sparsity and a small amount of measurements, this Iterative Reweighted Algorithm(IRA) is having similar error performance. However, once the sparsity increases to a significant portion of the signal, the Basis Pursuit approach generally does better in accuracy. Moreover, it's important to note that, the non-convex IRA method never really minimize ℓ_2 error estimation to trivial. All of the recovery still hovers around 0.2. However, as we set the *epsilon* to one hundredth of previous test, we get new Figure 6, which gives better accuracy than before, but still has the problem of leaving too much error slack even with large number measurements.

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[3] J. A. Tropp and A. C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[4] W. Dai and O. Milenkovic, "Subspace Pursuit for Compressive Sensing Signal Reconstruction," *arXiv:0803.0811 [cs, math]*, Mar. 2008, arXiv: 0803.0811. [Online]. Available: http://arxiv.org/abs/0803.0811

[5] B. Natarajan, "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, Apr. 1995. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/S0097539792240406

[6] S. Chen, D. Donoho, and M. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, Jan. 2001. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/S003614450037906X

[7] R. Chartrand, "Exact Reconstruction of Sparse Signals via Nonconvex Minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, Oct. 2007.

[8] "ASP – A Matlab solver for sparse optimization." [Online]. Available: http://www.cs.ubc.ca/~mpf/asp/

[9] D. L. Donoho and Y. Tsaig, "Fast Solution of -Norm Minimization Problems When the Solution May Be Sparse," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.