10/10

Reading Report of Gorodnitsky et al.

(ECE 695, Reading Assignment 05, Spring 2018)

Overview of the Paper

In order to estimate the solution to problems with limited data or estimate sparse signals, this paper introduces the FOcal Underdetermined System Solver (FOCUSS). In particular it hopes to inexpensively solve sparse problems in a nonparametric manner.

Prior Work

While the author mentions that there are many other methods to solving these sorts of problems, their goal in developing FOCUSS was to compensate for the weaknesses of these prior methods. Sparse signals can be estimated with greedy algorithms that searches A to find the solution, so can Linear Programming methods that solve L_1 or L_2 minimization. But the author feels the solutions with these methods are arbitrary with respect to the real signal. They also discuss the use of parametric methods, acknowledging they are good in some situations, but say they are interested in times where the signal doesn't have a good parametric model.

Key Ideas of the Paper

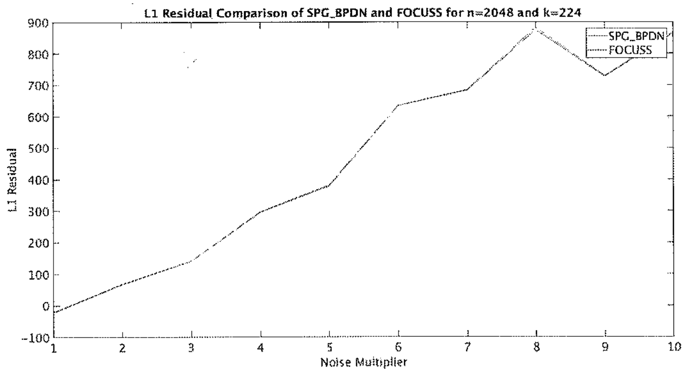
The author makes an effort to point out the algorithm is nonparametric because there are parametric methods for estimating sparse signals but they can't be used in all cases. There are cases where a parametric model can't accurately represent the signal. Also the author feels there are some steep limitations to using these methods that can't be overcome in some applications. There are two parts to the problem: finding an initial low resolution estimate of the signal, then iteratively improving this estimate.
The paper doesn't give much information on how to determine the low resolution estimate, though they do recommend "beamforming" and say that sparse initial estimate should be blurred. Then, an Affine Scaling Transformation (AST) is applied to constrain the sparse signal representation. If our problem is to solve for x in $Ax = b$, then AST can be represented as $q = X_{k-1}^{+}x$, with $X_{k-1} = diag(x_{k-1})$. This allows us to change our problem to optimizing for q. With this "reduced" optimization problem, we can now minimize it using the operations performed in the iterative step
FOCUSS's iterative step involves finding the L_2 weighted-norm minimization of q with the constraint $W_{pk} = X_{k-1}$. The author devotes a section of the paper explaining this formulation. They explain that the typical solution to L_2 minimization ($x_{mn} = A^{+}b$) doesn't provide a sparse solution. They introduce minimized weighted-norm formulation as $x = W(AW)^{+}b$ and modify W and the constraint so it is possible to find every solution. Mathematically, this means after applying the AST we solve for $x_k = W_{pk}q_k$, where $q_k = (AW_{pk})^{+}b$. The idea is to update the weights for our minimization based on the solution found in the previous iteration. While the author shows a Basic FOCUSS algorithm for conceptually understanding, they extend it to the General FOCUSS algorithm so it can be used as an optimization algorithm. This extension includes raising x_{k-1} to some power l and using an additional weight matrix W_{ak}. Unlike W_{pk}, W_{ak} is calculated once and is therefore independent of x_{k-1}. The general form is therefore to calculate $x_k = W_{ak}W_{pk}q_k$, where $W_{pk} = diag(x_{k-1}^{l}$ and $q_k = (AW_{ak}W_{pk})^{+}b$. The author then goes on to provide analysis on the convergence rate of the algorithm.

Comments

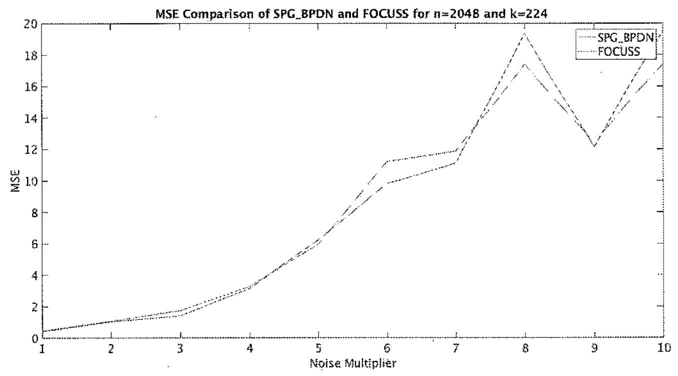
I decided to compare the effectiveness of this method with that of a previous algorithm we have studied. In our 3rd assignment we read about solving basis pursuit denoising (BPDN) using spectral projected-gradient (SPG) in Van de Berg et al. I created artificial sparse signal b in MATLAB and tried to recover x_0 with both FOCUSS and SPG_BPDN. While initially I used MSE to compare the similarity between the true signal and the estimate, I was concerned because with sparse signal reconstruction you often get the lowest MSE when an algorithm has smaller coefficients. These coefficients are not necessarily a good representation of the signal, but because most of the coefficients are zero, it will yield a better MSE. At the suggestion of a

friend, I decided to use the difference between L_1 norms of the true and estimated signals as a metric (I tried out both metrics to see how they compare).
The experiments I performed involved increasing the size of x_0 as well as the number of nonzeros in x_0. For a given x_0 and number of nonzeros k, I increased the "noise multiplier" a, which is a scalar that is multiplied by the random noise vector v in our experiment as $b = Ax_0 + va$. Over all sizes of x, both the MSE and L_1 residual increased as the a increased, which isn't surprising as the signal would be harder to reconstruct as the amount of noise increases. For the first experiment I doubled the size of x_0 as I doubled k, for my second experiment I left the size of x_0 fixed but doubled k. Of course, x_0 would not be sparse as k increases, but I wanted to see if the algorithms performed well regardless of sparsity.
For the first experiment (some of which is shown in figure 1), the performance of FOCUSS and SPG_BPDN is roughly the same when looking at the L_1 residual, though FOCUSS has a slightly larger residual overall. When looking at MSE, it is not entirely clear which one is better. For smaller values of the noise multiplier SPG_BPDN is better, but for larger values FOCUSS is better. Perhaps SPG_BPDN performs better with less noise because it uses a gradient method to find the signal, but its harder to get an accurate gradient calculation with all the noise. FOCUSS does not involve computing the gradient to minimize the solution. For our second experiment We see in figure 2 that both methods are both similar when looking at L_1 residual, but for MSE, FOCUSS performs worse as the noise increases. However, if the vector is not very sparse ($k = 448$ when $n = 512$), FOCUSS actually performs better as the noise increases compared to SPG_BPDN. However, both methods perform worse in this less sparse case. So SPG_BPDN appears to perform better on sparser data, but has a bigger drop in performance as the data becomes less sparse. I'm not entirely sure why this is, but my guess is it is because the minimization technique used in FOCUSS is much simpler computationally than in SPG_BPDN (which uses SPG) so it is less likely to be affected by a change in sparsity.

Very good!

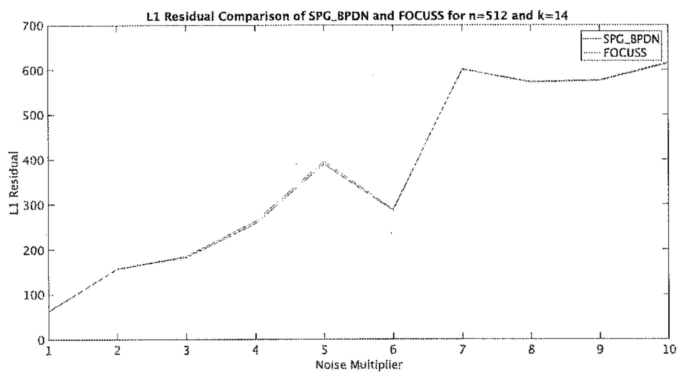


(a) L1 Metric

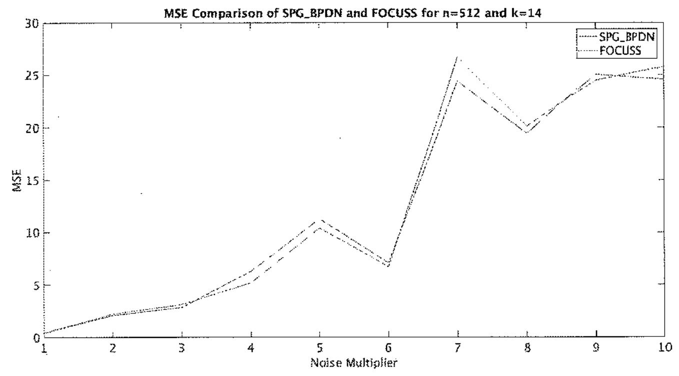


(b) MSE Metric

Figure 1: Comparison of Recovered Signal with FOCUSS and SPG_BPDN (k is proporational to size of x_0)



(a) L1 Metric



(b) MSE Metric

Figure 2: Comparison of Recovered Signal with FOCUSS and SPG_BPDN

Reading Report of Gorodnitsky et al.

(ECE 695, Reading Assignment 5, Spring 2018)

10/10

Overview of the Paper

This paper presents an algorithm for solving underdetermined system $Ax = b$ where $A \in \mathbb{R}^{m*n}$ and $m < n$. The algorithm introduced yields a sparse solution in general. The authors proved the convergence of the algorithm under certain circumstances and gave two examples for showing possible application of their algorithm.

Prior Work

Prior to this paper, many different algorithms have been designed for solving underdetermined system of equations: various forms of greedy algorithms (e.g. matching pursuit), genetic algorithms, linear programming methods etc.

Key Ideas of the Paper

The first key idea of the paper is about the uniqueness of sparse solution. Under the relatively strong constraint of Unique Representation Property (URP), the solution is guaranteed to be unique if it has a solution with dimension less than $\frac{m}{2}$.

The second key idea of the paper is to use the affine scaling transformation (AST) for pruning the solution space.

The third key idea of the paper is about the fixed point convergence. Given a sparse solution of FOCUSS algorithm, there exists a neighborhood that always converges to this sparse solution. Given a non-sparse solution of FOCUSS algorithm, the solution is a saddle point along certain trajectory in solution space. More importantly, the FOCUSS algorithm converges to non-sparse solution/ saddle points with 0 probability.

The fourth key idea is to regularize the the pseudo-inverse of AW with either Tikhonov regularization or with truncated SVD.

Comments

In the proof of uniqueness section, the Unique Representation Property is actually a very strong constraint on the structure of matrix A. What it requires is essentially that any single column of A must not live in the $m-1$ dimensional subspace spanned by any other columns of A. In other words, all $m-1$ dimensional subspaces spanned by any $m-1$ columns of A are unique; they cannot be spanned by other $m-1$ columns in A. This further implies that any k-dimensional subspace of F^m, where F is any valid field, spanned by k columns of A is unique in that dimension. From this, we can deduce that given a solution $\tilde{x}$ with k non-zero entries, for which $k \geqslant \frac{m}{2}$, the most sparse solution is not unique if and only if the vector b lies in the intersection between two different k-dimensional subspaces of F^m. Two interesting questions arise here: 1) given a sparse solution with k non-zero entries, can we quickly verify whether it is the most sparse solution if $k \geqslant \frac{m}{2}$? 2) given a solution with k ($k \geqslant \frac{m}{2}$) non-zero entries and that the solution is not unique, can we quickly find all other solutions with same sparsity (same number of non-zero elements)?

In my experiment, I tried to address a weaker version of the second question, in which I enforce A to satisfy URP, using FOCUSS. Assume that vector $\mathbf{x}_1 \in \mathbb{R}^n$ with k non-zero elements is a solution to $Ax = b, A \in \mathbb{R}^{m*n}$, and $k \geqslant \frac{m}{2}$ ($k < \frac{m}{2}$ case can be ignored since in that case, we know that the solution is unique). If $\mathbf{x}_1$ is not a unique most sparse solution, we have at least one other solution $\mathbf{x}_2$. From previous paragraph, we know that the columns in A corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ must be two disjoint sets (if they are not disjoint, then the subspace of these two subspaces will not be unique, or in other words, the columns will not be linearly independent). Hence, we can rewrite A as $A = A_1 + A_2$ where A_1 contains columns corresponding to $\mathbf{x}_1$ and all other columns equal to zero. We obtain following equations:

$$\begin{aligned}
&Ax_1 = b, Ax_2 = b \\
&A(x_1 + x_2) = (A_1 + A_2)(x_1 + x_2) = 2b \\
&A_1x_1 + A_2x_2 = 2b \\
&A_2x_2 = c, where, c = 2b - A_1x_1
\end{aligned}$$

Line2 to line 3 is based on the fact that $A_2x_1 = A_1x_2 = 0$. It can be seen that the last line is solving a smaller version of the original sparse solution finding problem. The number of columns of A_2 equal $n - k$. Therefore, we can iteratively use equation above to find all solutions with dimension k. The maximum number of iterations is upper bounded by $\frac{2n}{m}$(we remove k columns from A each time, then we can at most remove $\frac{n}{k}$ times, k is greater than or equal to $\frac{m}{2}$, so iteration is less than or equal to $\frac{2n}{m}$).

Experimental setup: In my experiment, I chose A to be a 3x32 matrix. $b = [0, 0, 1]$. Columns of A are generated by rotating unit vector [1,0,0] in x-y plane (rotate $\frac{\pi}{16}$ radians each time) and then add [0,0,1]. Hence, the columns of A form a cone and no three columns are on the same plane (enforcing the linear independence requirement of A). Using the procedures described by equations above, I was able to obtain all 16 different sets of solutions in 17 iterations. This experiment can be easily adapted to higher dimensional space; we can fix a vector and rotate another vector and then sum the two to form a hyper-cone which ensures that no three vectors live on the same hyper-plane.

Reference

I. F. Gorodnitsky and B. D. Rao, Sparse signal reconstruction from limited data using FOCUSS: A re-weighted norm minimization algorithm, IEEE Trans. Signal Processing, 45, pp. 600616, March 1997.

Reading Report of Sparse Signal Reconstruction from Limited Data using FOCUSS

(ECE 695, Reading Assignment V, Spring 2018)

10/10

Overview of the Paper

This paper describes the mechanics of non-parametric iterative re-weighted least squares algorithm (FOCUSS) that is sparsity inducing. The algorithm is absolutely convergent. The authors provide detailed mathematical analysis of the conditions under which this algorithm works and two applications of this algorithm to a Neuroimaging inverse problem and direction of arrival estimation have been discussed.

Prior Work

Other sparsity inducing approaches that exist are matching pursuit for basis selection, genetic search algorithms with a sparsity constraint, Bayesian restoration using Gibbs prior and l_p norm minimization techniques.

Key Ideas of the Paper & Assumptions

1. FOCUSS algorithm solves the following problem $min \ \ \|\boldsymbol{W}^{-1}\boldsymbol{x}\|_2 \ \ s.to \ \ \boldsymbol{Ax} = \boldsymbol{b}$.The premise of this algorithm is the following property :

$$\text{Solving } \|\boldsymbol{x}\|_p = \left(\sum_{n=1}^{N} |\boldsymbol{x}(n)|\right)^{1/p} \iff \text{Solving } \|\boldsymbol{x}\|_p = \left(\sum_{n=1}^{N} \frac{|\boldsymbol{x}(n)|^2}{\boldsymbol{w}(n)^2}\right)^{1/2}$$

   Note $\|\boldsymbol{x}\|_p$ is the l_p norm of $\boldsymbol{x}$. So FOCUSS is re-weighted least square algorithm that tries to approximate the l_p norm solution of $\boldsymbol{x}$. If we write the above optimization problem as an augmented lagrangian and solve for $\boldsymbol{w}(n)$, we get $\boldsymbol{w}(n) = \boldsymbol{x}(n)^{\frac{2-p}{2}}$. In [1], the authors define p as the diversity measure. The authors claim that in the range $0 < p \leq 1$, the algorithm is sparsity inducing. The choice of p = 1 is equivalent to minimizing the l_1 norm of the solution $\boldsymbol{x}$.

2. The methodology of scaling the iterates at every iteration by a power of the previous iterate is also called generalized Affine Scaling Transformation (AST).

3. The Unique Representation Property (URP) gives us some intuition about the sparsity of the solution. If the sensing matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}, m < n$ has $2\boldsymbol{k}$ linearly independent columns, then we can obtain a unique solution using the re-weighted minimum norm approach only if the true solution has sparsity $\leq k$. In other words $\|\boldsymbol{x}\|_0 \leq \boldsymbol{k}$.

4. Convergence of the algorithm is greatly affected by the choice of p and the quality of the initial point. This is justified using the concept of basins of attraction.

Comments

Experimental Setup: The code in its present form is very primitive. For simplicity I chose a well-conditioned random matrix $\boldsymbol{A} \in \mathbb{R}^{100 \times 256}$, with elements iid Gaussian, and modified the code to run the un-regularized version of the algorithm. The columns of the sensing matrix $\boldsymbol{A}$ have been to avoid issues (affinity to a sub-optimal stable fixed point) that arise due to bias in $\boldsymbol{A}$.The code is structured such that after 50% of the prescribed iterations are completed, the code will truncate the current iteration by picking the k highest components of $\boldsymbol{x}$, where k is the desired sparsity in the solution. To make the stopping criterion more robust, 2 conditions were set 1. if the sparsity of the current iterate meets the desired sparsity 2. if the relative error $\left(\frac{\|r\|}{\|y\|} \leq tol\right)$. There were 2 main simulations run.The goal of the first simulation was to study the URP constraint for a range of diversity measures p = 0.2,0.5 and 1 and for a range of k = 0 : 100

(k $= \|\boldsymbol{x}\|_0 =$ number of non-zeros elements). The initial point for all the conditions was kept the same $\boldsymbol{x}_{init} = (1/n)\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}$.There was no noise added to the measurements. The MSE error was computed between the $\boldsymbol{x}_{est}$ and $\boldsymbol{x}_{true}$. The results of this simulation are shown in Figure 1. The MSE displayed was computed by averaging over 100 runs to provide a more accurate picture.
Another simulation was done to study the convergence properties of the algorithm. Here, k = 20 was chosen as the diversity of the solution because the algorithm seems to converge to a sparse solution, for the same 3 diversity measures (p = 0.2,0.5,1). Since the convergence is heavily influenced by the initial point, I chose 2 initial points for this simulation, $\boldsymbol{x}_{init}^{(1)} = (1/n)\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}$ and $\boldsymbol{x}_{init}^{(2)} = 5 * rand(n, 1)$.The second initialization point was carefully chosen to be far away from the solution.The results for this simulation exercise are shown in Figure 2.
Observations - URP Although the MSE is not the best metric for accuracy, considering it suffers from bias, the graph of the MSE does portray a trend which is in agreement with our understanding of the algorithm. For the case p = 1, which corresponds to the l_1 norm, FOCUSS is able to recover a solution with low MSE when the number of non-zeros < 30. For $n > 30$, we start seeing a degradation in the MSE. This can be attributed to the URP which requires atleast $2 \times nnz$ columns of $\boldsymbol{A}$ to be linearly independent. The rank of matrix $\boldsymbol{A}$ was found to be 100,hence good performance for nnz< 30 implies that a basis for the solution exists. In the presence of noise, we see a much more rapid degradation in noise. The un-regularized IRLS is no match for problems with poor SNR in the measurements.

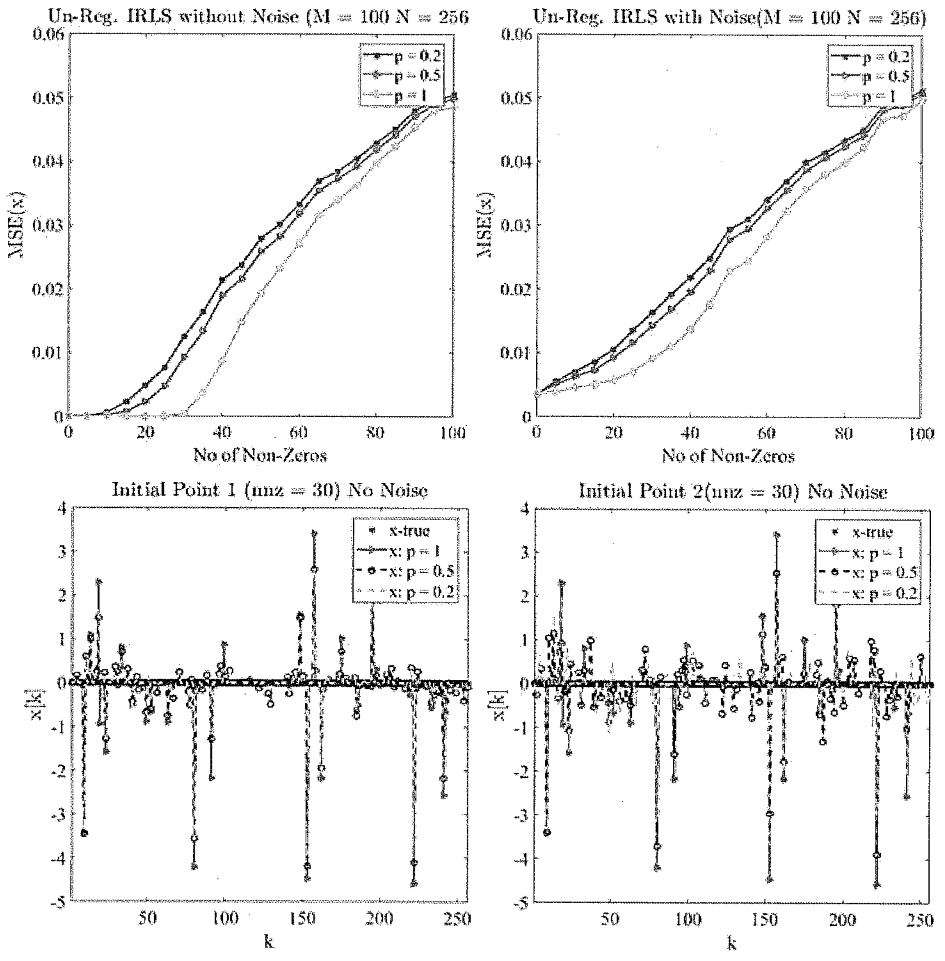


Figure 1:

Observations - Convergence - Convergence is better when p = 1 as compared to $p < 1$. If we look at the objective function for $p < 1$, the objective function is piecewise convex and has a lot of local minimas. For each of these local minima's, we have a basin of attraction. Depending on which basin we pick, FOCUSS converges to the minima corresponding to that basin. Figure 2 clearly explains this phenomenon. For $0 < p < 1$, two different initialization points lead to 2 different solutions. But for p = 1, for both initializations, we arrive at the same minima. One key takeaway from this simulation was that if we don't have apriori knowledge of our solution, it is very hard to initialize the algorithm close to the basin that corresponds to the correct solution. The basins of attraction are tightly interlinked to the diversity measure p and the matrix $\boldsymbol{A}$.

yes. very good comment

References

[1] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, no. 2, pp. 349–396, 2003.

10/10

Reading Report of Gorodnitsky et al.

(ECE 695, Reading Assignment 05, Spring 2018)

Overview of the Paper

The paper proposed an non-parametric iterative method for solving inverse linear problems in general signal processing or image processing. The problem addressed is under-determined and no prior knowledge about the signal we want to recover is assumed. The method requires a initial estimation, and then the iterative process is performed to refined the result. The refinement procedure is built upon a weighted norm minimization concept such that the result is sparse. The weight is determined based on the result of the previous iteration. The proposed algorithm is a general tool for estimation and can be used across different applications. Proof of difference convergence properties and convergence rate is also included in the paper. Although the basic version of the proposed FOcal Underdetermined System Solver(FOCUSS) does not requires any prior information of the data we want to recover, the more general version of FOCUSS can makes use of a priori term in the iterative refinement process.

Prior Work

Many existing algorithms can be used to solve under-determined systems. Common techniques include exhaustive search methods like greedy algorithm,evolutionary searches such as a generic algorithm with sparsity constraint. However, these methods does not makes of additional information other than the sparsity. So the results are often not well constrained. l norm minimization and Linear Programing approaches also exist, but these methods introduces cost functions that is arbitrarily related to the real signal we want to recover. As a result, their solution to the problem is often arbitrary as well. When the regularization term is l_1, then there are a lot of competitive methods. Some notable ones are MOSEK, LARS, SEDUMI, PDCO, l1ls and SPGL. The FISTA method from the previous assignment can also be used in this case.

Key Ideas of the Paper

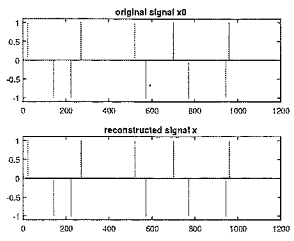
The key concept of FOCUSS algorithm is actually pretty straight forward. It's the use of solution of the previous iteration as the weight for norm minimization for the current iteration. The basic form of FOCUSS is shown below

$$\begin{aligned} \boldsymbol{W}_{pk} &= (diag(\boldsymbol{x}_{k-1})) \\ \boldsymbol{q}_k &= (\boldsymbol{A}\boldsymbol{W}_{pk})^{+}\boldsymbol{b} \\ \boldsymbol{x}_k &= \boldsymbol{W}_{pk}\boldsymbol{q}_k \end{aligned}$$

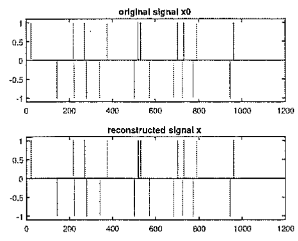
It is obvious that the zero elements in the initial estimate $\boldsymbol{x}_0$ will stay zero. The weight matrix $\boldsymbol{W}$ essentially serves to prune the solution. If the entry for $\boldsymbol{x}_{k-1}$ is large, then the corresponding diagonal entry in $\boldsymbol{W}$ will be large, which will then reduce the weight for this entry when minimizing the cost. And this will then lead a large entry for $\boldsymbol{x}_k$. On the other hand, if an entry in $\boldsymbol{x}_[k-1]$ is small, then the corresponding entry of $\boldsymbol{W}$ is small which means a higher weight when minimizing the weighted norm. And this will then produce a smaller value for the entry of $\boldsymbol{x}_k$. So essentially the weight matrix serves as a soft thresholding operation that resembles the iterative shrinkage method ISTA. good analysis

The general version of FOCUSS allows the user to raise the power of $\boldsymbol{x}_{k-1}$ from 1 to some arbitrary number l so that the thresholding effect could be even stronger. Another weight matrix is also added so that the algorithm can be more flexible to be used for solving many different applications. This is especially convenient when we want to input some a priori information in the FOCUSS algorithm for helping refine the solution.

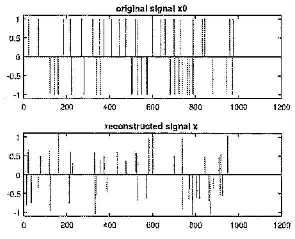
A cumulative form of FOCUSS is also discussed, where one can use the product of solutions from several previous iterations as the weight instead of using only the precedent single iteration. The this form is shown to produce more robust solution near the initialization.



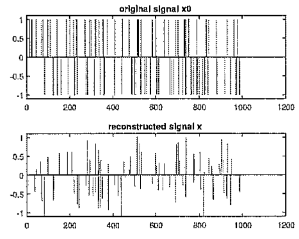
case 1



case 2



case 3



case 4

Comments

I ran the code provided by the author for different experimental setups. For a signal of dimension 1000, the number of measurements is 128. I ran the test for ground truth signals with different number of spikes ranging from $10, 20, 50, 100$. Apparently for this relatively short signal length 1000, the best performance is achieved when the number of spikes are small. From the four pairs of figures above, we can see the cases for 10 and 20 spikes the recovery is pretty good. However, as when the number of spikes is increased the performance is not good any more as in case 3 and 4. This is very reasonable, since the problem is solved under the assumption of sparsity of the signal. When the signal has a lot of spikes, the sparsity property is lost, and the solution would be bad.

One thing interesting is that the code provided by the author has a parameter called 'targetdiv', which corresponds to the number of spikes in the our tests. When the targetdiv is set to the correct number of spikes, the performance is good, however if the parameter is set to a values very different from the correct number of spikes in the ground truth signal, the result is quite different from the ground truth. This is expected as further investigation into the code reveals the code simply keep the maximum number of values of the solution for each iteration. I don't think this a good for the algorithm, as the sparsity of the ground truth signal is often unknown, and this makes the algorithm not very general. All previous algorithms we learned from earlier reading assignments however does not require the user to input a sparsity measure. Although this paper claims to not rely on designing cost function, but it requires a given sparsity. And I personally think this is not a good idea.

Yes. This is a downside of FOCUSS

Another thing worth testing is the power used for producing the weight matrix from the solution of the previous iteration. When we use a power of 1 or 2, the solution is pretty standard. However, if we use a values that is close to 0, then the solution would have values that is closer to 0. This is probably not surprising since the signal's magnitude is smaller than 1, and with a power close to zero this will makes the weights large, and thus smaller spikes are produced in the solution. It is also mentioned in the paper that the power is often chosen as positive integers so that aforementioned issues could be ignored.

It is also worth mentioning the initial estimator of the algorithm. In the code provided by the author, the initial estimated is chosen to be a vector of ones. It might be attempting to try using a vector of zeros as the initial guess, however due to the design of the algorithm, a zero value in the solution of any iteration, would cause a zero value in the solution for the all following iterations. This makes one to expect the initial guess would have big impact on the final solution of FOCUSS, however this does not seem to be true based on experiments as long as zeros are carefully handled in the initial estimate. In general, I think this method can be applied to general inverse problem with sparsity assumption, however it does seem to be more restrictive compared to methods likes FISTA and weird in the sense that it requires the user to input the number of spikes in the ground truth in order to have a better performance.

There is a basin of attraction for the convergence