Non-Iterative Reweighted-Norm Least-Squares Local ℓ_0 Minimization for Sparse Solutions to Underdetermined Linear Systems of Equations

Andrew E. Yagle
Department of EECS, The University of Michigan, Ann Arbor, MI 48109-2122

Abstract— **We present a non-iterative algorithm for computing sparse solutions to underdetermined M×N linear systems of equations. The algorithm computes a solution which is a local minimum of the ℓ_0 norm (number of nonzero values) obtained from the ℓ_1 norm (sum of absolute values) minimum. At each step, it uses reweighted-norm least-squares minimization to compute the ℓ_p^p norm for values of p decreasing from 2 to 0. The result is similar to the ℓ_1 solution, but uses less computation (solution of ten M×M systems of equations), and there are no convergence issues.**

Keywords— **Sparse reconstruction**
Phone: 734-763-9810. Fax: 734-763-1503.
Email: aey@eecs.umich.edu. EDICS: 2-REST.

I. INTRODUCTION

A. Problem Statement

The goal is to compute a sparse solution to the underdetermined (M<<N) linear system of equations

$$y = Ax; \quad \mathbf{A}: M \times N; \quad \mathbf{x}: \text{N vector}; \quad \mathbf{y}: \text{M vector}.$$

This can be done by solving any of the following:

$$\mathbf{MIN}\,\{||x||_1\} \quad \mathbf{such} \quad \mathbf{that}\, y = Ax \quad (\mathbf{BP}) \tag{1}$$
$$\mathbf{MIN}\,\{||y - Ax||_2^2 + \lambda||x||_1\} \qquad (\mathbf{LASSO}) \tag{2}$$
$$\mathbf{MIN}\,\{||y - Ax||_2^2 + \lambda \sum_{n=2}^{N} |x_n - x_{n-1}|\} (\mathbf{TV}) \tag{3}$$

where the ℓ_p (and ℓ_1) norms is defined as

$$||x||_p = \sqrt[p]{\sum_{n=1}^{N} |x_n|^p} \quad \text{and} \quad ||x||_1 = \sum_{n=1}^{N} |x_n| \tag{4}$$

and TV is the Total Variation functional in 1-D.

If A has random entries, then ℓ_1 norm minimization of x tends to sparsify the solution x to $y = Ax$ as much as possible. This problem is currently of great interest in compressed sensing, since many real-world signals and images have sparse (mostly zero) representations in an appropriate basis, such as a set of wavelet or curvelet basis functions.

Sparse solutions can be computed using *matching pursuit*, in which the columns of A most highly correlated with $y - A\hat{x}_i$ are successively chosen to minimize the residual $y - A\hat{x}_{i+1}$. Here $\hat{x}_i$ is the approximation to x after i iterations. This produces a sparse solution that approximately satisfies $y = Ax$, but it does not always find the optimally sparse solution.

The solution to (1) can be computed using *basis pursuit (BP)*, in which linear programming is used to minimize the functional (1). This has been done in the geophysics literature since the 1960s, for deconvolution of sparse spike trains in reflection seismology.

The functional (2) is appropriate when zero-mean Gaussian white noise is present in the data. The solution to (2) or (3) can be computed using gradient methods, coordinate descent, thresholded Landweber iteration, or iterative reweighted-norm least-squares.

Many review papers have appeared recently on these problems; we will not attempt to review all of the recent work on all of these problems here.

B. Iterative Reweighted Least Squares (IRLS)

We propose a variation of IRLS that is applicable to all of the above problems. IRLS uses the fact that

$$||x||_p^p = \sum_{n=1}^{N} \frac{1}{|x_n|^{2-p}} x_n^2 \tag{5}$$

and the weighted-least-squares (WLS) problem

$$\mathbf{MIN}\left\{\sum_{n=1}^{N} d_n x_n^2\right\} \quad \mathbf{such} \quad \mathbf{that} \quad y = Ax \tag{6}$$

with $d_n > 0$ has the closed-form solution

$$x = D^{-1}A^T(AD^{-1}A^T)^{-1}y; D = \text{diag}[d_1 \dots d_N] \tag{7}$$

which requires as its main computation the solution of an M×M system of equations (recall M<<N). IRLS iteratively solving the WLS problem (1) with

$$d_n = \frac{1}{|x_n|^{2-p} + \epsilon} \tag{8}$$

where the x_n are from the previous iteration and we ensure $d_n \geq \epsilon$ to avoid problems when x_n=0.

II. New Algorithm

A. Recursion in Decreasing p

We propose to minimize $||x||_p^p$ *recursively*, not iteratively, by slightly reducing the norm order p at each recursion. At each step, we have minimized $||x||_p^p$ and wish to minimize $||x||_{p-\delta}^{p-\delta}$ for some small $\delta = 2/L$, where L is the number of recursions required to compute the local minimum of $||x||_\delta^\delta$. We do this as follows. For convenience we demonstrate this for the BP criterion (1), then modify to LASSO.

At a given recursion we have minimized $||x||_p^p$. This minimum $\hat{x}$ is the solution to WLS problem (6) with

$$d_n = \frac{1}{|\hat{x}_n|^{2-p}}, \quad \hat{x}_n \neq 0 \tag{9}$$

We could not have computed $\hat{x}$ by inserting (9) in (7) since we need to know $\hat{x}$ in order to compute it.

The minimizer of $||x||_{p-\delta}^{p-\delta}$ is the solution to the WLS problem (6) with the weights d_n set to

$$d_n = \frac{1}{|x_n|^{2-(p-\delta)}} = \frac{1}{|x_n|^{2-p}} \frac{1}{|x_n|^\delta}. \tag{10}$$

Since this minimizer $x_n \approx \hat{x}_n$ and $|x_n|^\delta \approx 1$, we have

$$d_n \approx \frac{1}{|\hat{x}_n|^{2-p}} \frac{1}{|x_n|^\delta} \approx \frac{1}{|\hat{x}_n|^{2-p}} \tag{11}$$

Hence we can, to a good approximation, compute the minimizer of $||x||_{p-\delta}^{p-\delta}$ by using the minimizer of $||x||_p^p$ to compute the weights d_n for the WLS problem using (9). This seems to be a trivial observation, but in fact it can be viewed as a type of continuation algorithm, in which an easily-solvable problem is perturbed into the desired problem.

More importantly, there are no covergence issues, and the algorithm terminates after a known and finite number of operations. We have found experimentally that using δ=0.2 as the reduction of norm order p at each recursion seems to work reasonably well, so that a total of $\frac{2}{0.2}$=10 recursions are then required.

B. LASSO Functional

We can apply the same idea to the LASSO functional (2). The minimizer of

$$\textbf{MIN} \left\{ ||y - Ax||_2^2 + \sum_{n=1}^{N} d_n x_n^2 \right\} \tag{12}$$

with $d_n > 0$ has the closed-form solution

$$x = (A^T A + D)^{-1} A^T y; \quad D = \text{diag}[d_1 \dots d_N] \tag{13}$$

which has as its main computation the solution of an N×N system of equations. Since N>>M, we use

$$(A^T A + D)^{-1} A^T = D^{-1} A^T (A D^{-1} A^T + I)^{-1} \tag{14}$$

to rewrite the closed-form solution as

$$x = D^{-1} A^T (A D^{-1} A^T + I)^{-1} y \tag{15}$$

which has as its main computation the solution of an M×M system of equations. Since M<<N, this is a significant computational savings at each recursion.

III. NUMERICAL EXAMPLE

We compare the results of our proposed algorithm with IRLS for a simple problem in which:

- A is a random 25×50 matrix;
- x is an 8-sparse length-50 vector;
- ϵ=0.00001 in (9) for both algorithms.

Results are shown in the figure. Note that result of our algorithm is very close to the solution (mean-square error=0.15) while IRLS is still quite far away (mean-square error=0.96). Similar results were obtained for other random matrices and are not shown.

We note that IRLS eventually converged to the solution, but dozens of iterations could be required before the mean-square error became smaller than our algorithm. Of course, the results of our algorithm could be improved with a couple of final iterations.

Matlab code used to generate this example:

```
clear;rand('seed',0);H=rand(25,50);X(50)=0;
X(3)=1;X(9)=1;X(13)=1;X(23)=1;X(30)=1;X(38)=1;
X(43)=1;X(46)=1;X=X';Y=H*X;
Z1=ones(1,50);B1=[];Z2=ones(1,50);B2=[];for I=1:9
G1=H*diag(abs(Z1)+0.00001);
Z1=G1'*((G1*G1')\Y);B1=[B1 Z1];end;
subplot(211),plot(1:50,X,1:50,Z1),for I=1:9
G2=H*diag(abs(Z2).^(0.2*I)+0.00001);
Z2=G2'*((G2*G2')\Y);B2=[B2 Z2];end
subplot(212),plot(1:50,X,1:50,Z2)
MSE1=sum((X-Z1).^2);
MSE2=sum((X-Z2).^2);[MSE1 MSE2]
```