Coordinate Descent for Sparse Solutions of Underdetermined Linear Systems of Equations

Andrew E. Yagle (revised Nov. 2009)
Department of EECS, The University of Michigan, Ann Arbor, MI 48109-2122

Abstract— **The problem of computing sparse (mostly zero) or sparsifiable (by linear transformation) solutions to greatly underdetermined linear systems of equations has applications in compressed sensing and reduced-exposure imaging. We show that coordinate descent, in which a single variable is varied while all others are held constant, is applicable to a wide variety of such problems. The method is useful for very large problems having dense system matrices.**

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 minimize the cost functional

$$F=\sum_{i=1}^{M}\left(y_i-\sum_{j=1}^{N}A_{ij}x_j\right)^2+$$
$$\sum_{i=1}^{N}\lambda_i\left|\sum_{j=1}^{N}B_{ij}x_j\right|+\sum_{i=1}^{N}\gamma_i^2\left(\sum_{j=1}^{N}C_{ij}x_j\right)^2 \qquad (1)$$

- A is an M×N system matrix with M<<N;
- B is an N×N invertible sparsifying matrix;
- C is an N×N matrix, but need not be invertible.
- y is an M-vector of known observations (data);
- x is an N-vector of unknown variables to find;
- Bx is desired to be sparse (mostly zero-valued);
- λ_i and γ_i are penalty term weights.

B. Discussion of the Cost Functional

The ℓ_1-norm penalty term $||Bx||_1$ penalizes small deviations of the elements of Bx from zero. A considerable amount of research since 2000 has proved what the geophysical community has observed since the 1960s: The ℓ_1 norm produces sparse solutions. In fact, the ℓ_1 norm produces the maximally sparse solution if the number K of nonzero elements has $K\log K <$M and A is a matrix of random elements.

The ℓ_2-norm penalty term $||Cx||_2$ does not penalize small deviations, since its slope is zero at zero. But it does penalize large deviations more heavily than the ℓ_1 norm, and thus stabilizes the solution in the presence of noise. Use of an ℓ_2 norm penalty term for this purpose is called Tikhonov regularization.

Thus the cost functional finds a solution to the underdetermined linear system $y = Ax$ such that:

- Bx is sparse (mostly zero);
- Cx does not become large;
- Zero-mean white Gaussian noise in the data y.

The cost functional admits the stochastic interpretation of MAP estimation by minimizing the log-likelihood function when the data y includes white Gaussian noise and the unknown x is modelled by a mixture of independent zero-mean Gaussian and Laplacian (two-sided exponential) random variables.

C. The Sparsifying Linear Transformation B

The choice of B is important, since this specifies the variables to be sparsified. One choice is an orthogonal or biorthogonal wavelet transform. Another useful choice is the successive difference operator

$$B=\begin{bmatrix}1&-1&0&\cdots&0\\0&1&-1&\cdots&0\\\ddots&\ddots&\ddots&\ddots&\ddots\\0&0&0&\cdots&1\end{bmatrix} \qquad (2)$$

$$B^{-1}=\begin{bmatrix}1&1&1&\cdots&1\\0&1&1&\cdots&1\\\ddots&\ddots&\ddots&\ddots&\ddots\\0&0&0&\cdots&1\end{bmatrix} \qquad (3)$$

useful for piecewise-constant signals or images.

The difference operator can also be used for C to impose smoothness without flatness. For images, the Kronecker product $B\bigotimes B$ is applied to the image by rows or columns. The weights λ_i and γ_i are usually chosen to be constant over i, but different pixels can be penalized differently if a priori information about different regions of the signal or image is available.

D. Reformulation as LASSO functional

Defining $z = Bx$, the cost functional (1) can be rewritten as the familiar LASSO cost functional

$$F=\sum_{i=1}^{M+N}\left(\tilde{y}_i-\sum_{j=1}^{N}\tilde{A}_{ij}z_j\right)^2+\sum_{i=1}^{N}\lambda_i|z_i| \qquad (4)$$

$$\tilde{A}=\begin{bmatrix}A\\ \text{DIAG}[\gamma_i]C\end{bmatrix}B^{-1};\quad \tilde{y}=\begin{bmatrix}y\\0\end{bmatrix} \tag{5}$$

- The system is now overdetermined;
- If C=0 it is still underdetermined;
- B=0 gives least-squares regularization;
- Strangely, the variable change $z=Bx$ seems to be unfamiliar in much of the statistics literature.

II. Coordinate Descent Algorithms

A. Overview

Coordinate descent, sometimes called "one-at-a-time" algorithms, minimize a cost functional by minimizing it over a single variable while holding all other variables constant. There are two approaches:

- Cyclic algorithms cycle through all of the variables;
- Greedy algorithms choose the variable that reduces the cost by the largest amount in each iteration.

Coordinate descent has these advantages over methods that deal with the entire problem each iteration:

- The minimization problem for a single variable is much simpler to compute than for all variables;
- Each problem has size M instead of MN;
- Guaranteed global convergence to minimum F.

Cyclic coordinate descent is guaranteed to converge to the global minimum of the cost functional $f(x)$ from any initial point if the cost functional $f(x)$ is:

- Convex: f(ax+(1–a)y)<af(x)+(1–a)f(y); and
- Continuously differentiable: continuous ∇f(x); or
- LASSO (which is convex but not differentiable).

Roughly, this is because ∇f(x) must have a nonzero component in some direction unless x is the global minimum, and so coordinate descent will reduce f(x).

B. Application to LASSO

Define the partial residual r_{ip} in LASSO (4) as

$$r_{ip}=y_i-\sum_{\substack{j=1\\j\neq p}}^{N}A_{ij}x_j\quad p\text{ denotes } j \text{ skipped} \tag{6}$$

r_{ip} is related to the error e_i in LASSO (4) by

$$e_i=y_i-\sum_{j=1}^{N}A_{ij}x_j=r_{ip}-A_{ip}x_p \tag{7}$$

Setting the partial derivative $\frac{\partial}{\partial x_p}$(LASSO)=0 gives

$$-2\sum_{i=1}^{M+N}A_{ip}(r_{ip}-A_{ip}x_p)+\lambda_p\text{SIGN}[x_p]=0 \tag{8}$$

Solving this equation for x_p and defining T_p as

$$T_p=\sum_{i=1}^{M+N}r_{ip}A_{ip}=\sum_{i=1}^{M+N}A_{ip}\left(y_i-\sum_{\substack{j=1\\j\neq p}}^{N}A_{ij}x_j\right) \tag{9}$$

and the ℓ_2-norm of the p^{th} column of matrix A as

$$\sum_{i=1}^{M+N}A_{ip}^2=||A_p||^2>0\,\text{unless}\,A_{ip}=0\,\text{for all}\,i \tag{10}$$

gives the thresholded inner product computation

$$\hat{x}_p=\begin{cases}(T_p-\lambda_p/2)/||A_p||^2 & \text{if } T_p>+\lambda_p/2;\\ (T_p+\lambda_p/2)/||A_p||^2 & \text{if } T_p<-\lambda_p/2;\\ 0 & \text{if } |T_p|<\lambda_p/2.\end{cases} \tag{11}$$

The smaller λ_p is, the smaller the interval in which $\hat{x}_p$ is thresholded to zero. Also note the following:

- To minimize fewer than N linear combinations of x in (1), choose λ_i=0 for the undesired terms of $||Bx||_1$ (note that B must be N×N to be invertible);
- λ_i=0 computes the weighted least squares solution. This is called "relaxation" and dates to the 1940s;
- If $\hat{x}_p$=0, r_{ip} isn't updated, saving computation;
- $\tilde{A}$ in (5) should be a random matrix for ℓ_1-norm minimization to sparsify the solution.

III. Relation to Thresholded Landweber

If we make the following two alterations:

- Pre-normalize the matrix A so that $||A_i||$=1;
- Update the residual r_i only after a complete pass;

then the update (9) can be written as

$$T_p=x_p+\sum_{i=1}^{M+N}A_{ip}\left(y_i-\sum_{j=1}^{N}A_{ij}x_j\right) \tag{12}$$

since $||A_p||$=1. Collecting these for all p gives

$$\vec{T}=\vec{x}+A^T(\vec{y}-A\vec{x}) \tag{13}$$

followed by thresholding each element of $\vec{T}$.

We recognize this as a thresholded *Landweber iteration* (also called the Van Cittert iteration). This means that *thresholded Landweber iteration is equivalent to coordinate descent without updating*, with the matrix normalized so that each column has unity norm. Unless the matrix A has Toeplitz structure, enabling fast computation of $A\vec{x}$ and $A^T\vec{r}$, there is therefore no point in using thresholded Landweber.

A. Review of Landweber Iteration

The Landweber iteration is a Neumann series:

$$(I-B)^{-1}=I+B+B^2+\ldots;B=I-A^TA \quad (14)$$

x=$(A^TA)^{-1}A^Ty$ can be computed recursively as

$$x_{\text{NEW}}=Bx+A^Ty=(I-A^TA)x+A^Ty \quad (15)$$

which can be rewritten as the Landweber iteration. Hence the Landweber iteration converges only if

$$||B||=||I-A^TA||<1\leftrightarrow\sigma_A<2 \quad (16)$$

where σ_A are the singular values of A. Note that normalization of the columns of A does not guarantee this, so thresholded Landweber, and coordinate descent without updating, may both be unstable.

IV. Small Example

We apply the algorithm to reconstruct a 10×10 block "E" from a 49×100 linear transformation of it. We use for B the Kronecker product of the difference operator defined in (2) with itself. This sparsifies the block letter to its 12 nonzero corners. Since B is invertible, the block letter can be quickly computed from its sparsified version, since B is invertible.

Reconstruction using 49 observations is shown in Figure #1, and is visually excellent. The unwrapped sparsified letter and its reconstruction are both plotted in Figure #2 and match closely, despite the close proximity of several nonzero values of opposite sign.

Reconstruction is using 47 (two fewer) observations is shown in Figure #3. Note that just two fewer observations significantly degrades the reconstruction. This is a property of the ℓ_1-norm minimization itself.

The Matlab program is given below.

```
clear;rand('seed',0);A=rand(49,100);L=0.1;
X(10,10)=0; %49 works well; 47 doesn't
X(2:3,2:9)=ones(2,8);X(8:9,2:9)=ones(2,8);
X(5:6,2:9)=ones(2,8);X(2:9,2:3)=ones(8,2);
B=toeplitz([1 zeros(1,9)],[1 -1 zeros(1,8)]);
IB=toeplitz([1 zeros(1,9)],ones(1,10));
A=A*kron(B,B);Y=A*X(:);W=kron(B,B)*X(:);
%GOAL:Recover block letter E from data Y
%W is sparsified block letter for reference
%Need system matrix A*kron(B,B) so A random
A=A*kron(IB,IB);AP(1:100)=sum(A.^2);R=Y;
Z=zeros(100,1);for I=1:1000;for J=1:100;
R=R+A(:,J)*Z(J);T=R'*A(:,J);
if(T>+L/2);Z(J)=(T-L/2)/AP(J);end
if(T<-L/2);Z(J)=(T+L/2)/AP(J);end
if(abs(T)<L/2);Z(J)=0;end
R=R-A(:,J)*Z(J);end;end;
XHAT=IB*reshape(Z,10,10)*IB;
figure,plot(1:100,W,1:100,Z)
figure,imagesc(XHAT),colormap(gray)
```

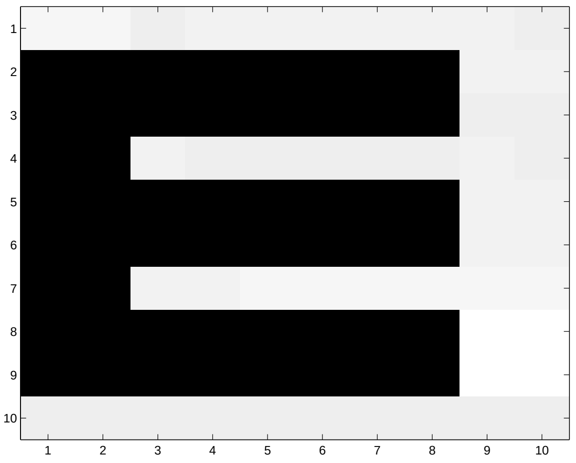
Fig. 1. Reconstruction using 49 observations

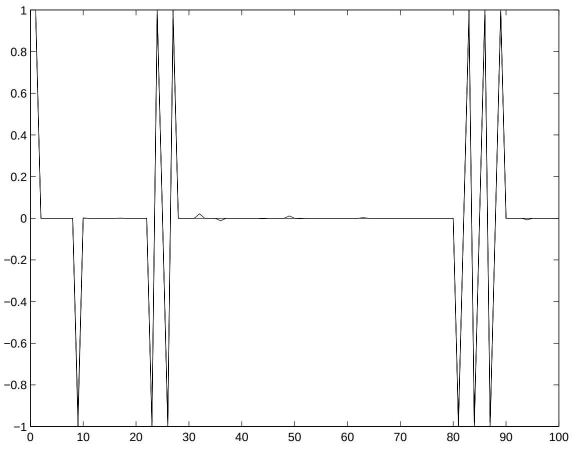
Fig. 2. Unwrapped original and reconstruction

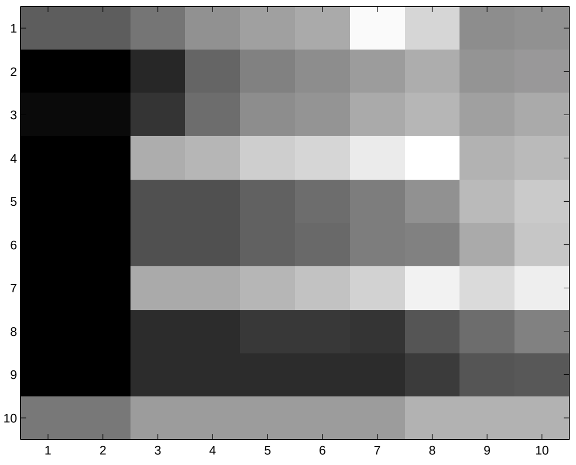
Fig. 3. Reconstruction using 47 observations