Coordinate Descent for L_1 Minimization

Yingying Li and Stanley Osher

Department of Mathematics
University of California, Los Angeles

Jan 13, 2010

Outline

- Coordinate descent and L_1 minimization problems
- Application to compressed sensing
- Application to image denoising

Basic Idea of Coordinate Descent

Consider solving the optimization problem

$$\tilde{u} = \arg\min_{u \in \mathbb{R}^n} E(u), \quad u = (u_1, u_2, \ldots, u_n).$$

To solve it by coordinate descent, we update one coordinate while fixing all the others every step and iterate.

$$\tilde{u}_j = \arg\min_{x \in \mathbb{R}} E(u_1, \ldots, u_{j-1}, x, u_{j+1}, \ldots, u_n)$$

The way we choose the updated coordinate j is called the sweep pattern, which is essential to the convergence.

Models Including L_1 Functions

We consider energies including an L_1 term

$$E(u_1, u_2, \ldots, u_n) = F(u_1, u_2, \ldots, u_n) + \sum_{i=1}^{n} |u_i|, \tag{1}$$

and also energies with a total variation (TV) term

$$E(u_1, u_2, \ldots, u_n) = F(u_1, u_2, \ldots, u_n) + \sum_{i=1}^{n} |\nabla u_i|, \tag{2}$$

where F is convex and differentiable and ∇ is a discrete approximation of gradient.

Models Including L_1 Functions

- Compressed Sensing

$$u = \arg\min |u|_1 \text{ s.t. } Au = f \tag{3}$$

- TV-base image processing, like ROF (Rudin-Osher-Fatemi) denoising and nonlocal ROF denoising (Guy Gilboa and Stanley Osher).

$$u = \arg\min |u|_{\text{TV}} + \lambda \|u - f\|_2^2 \tag{4}$$

$$u = \arg\min |u|_{\text{NL-TV}} + \lambda \|u - f\|_2^2 \tag{5}$$

where λ is a scalar parameter and f is input noisy image.

Why Coordinate Descent?

Advantages

- Scalar minimization is easier than multivariable minimization
- Gradient free
- Easy implementation

Disadvantages

- Hard to take advantage of structured relationships among coordinates (like a problem involving the FFT matrix)
- The convergence rate is not good; sometimes it gets stuck at non-optimal points (for example, fused lasso $E(u) = \|Au - f\|_2^2 + \lambda_1 |u|_1 + \lambda_2 |u|_{\mathrm{TV}}$)

Coordinate Descent Sweep Patterns

Sweep Patterns

- Sequential: $1, 2, \ldots, n, 1, 2, \ldots, n, \ldots$
  or $1, 2, \ldots, n, n-1, n-2, \ldots, 1, \ldots$
- Random: permutation of $1, 2, \ldots, n$, repeat

Essentially Cyclic Rule

Suppose the energy is convex and the nondifferentiable parts of the energy function are separable. Then coordinate descent converges if the sweep pattern satisfies the essentially cyclic rule, that every coordinate is visited infinitely often.

However, no theoretical results for the convergence rate.

Compressed Sensing (CS)

$$\min_{u \in \mathbb{R}^n} E(u) = |u|_0, \quad \text{subject to } Au = f.$$

If the matrix A is incoherent, then it is equivalent to its convex relaxation:

$$\min_{u \in \mathbb{R}^n} |u|_1, \quad \text{subject to } Au = f.$$

Restricted Isometry Condition (RIC): A measurement matrix A satisfies RIC with parameters (s, δ) for $\delta \in (0, 1)$ if we have

$$(1 - \delta)\|v\|_2 \leq \|Av\|_2 \leq (1 + \delta)\|v\|_2 \qquad \textit{for all } s\textit{-sparse vectors.}$$

When δ is small, the RIC says that every set of s columns of A is approximately an orthonormal system.

Bregman Iterative Method

The constrained problem:

$$\min_{u \in \mathbb{R}^n} |u|_1 \quad \text{subject to } Au = f.$$

The unconstrained problem:

$$\min_{u \in \mathbb{R}^n} \; E(u) = |u|_1 + \lambda \|Au - f\|_2^2,$$

Bregman Iterative Algorithm

1: Initialize: $k = 0$, $u^0 = \mathbf{0}$, $f^0 = f$.
2: while $\frac{\|Au-f\|_2}{\|f\|_2} >$ tolerance do
3: Solve $u^{k+1} \leftarrow \arg\min_u \ |u|_1 + \lambda\|Au - f^k\|_2^2$ by coordinate descent
4: $f^{k+1} \leftarrow f + (f^k - Au^{k+1})$
5: $k \leftarrow k + 1$
6: end while

Solving the Coordinate Subproblem

$$\min_{u_j} E(u) = \lambda \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} u_j - f_i \right)^2 + \sum_{i=1}^{n} |u_i|$$

$$= \lambda \left[\left(\sum_{i=1}^{p} a_{ij}^2 \right) u_j^2 - 2 \left(\sum_{i} a_{ij} (f_i - \sum_{k \neq j} a_{ik} u_k) \right) u_j + \sum_{i=1}^{p} f_i^2 \right] + \sum_{i=1}^{n} |u_i|$$

$$= \lambda \left(\|a_j\|_2^2 u_j^2 - 2\beta_j u_j + \|f\|_2^2 \right) + |u_j| + \sum_{i \neq j} |u_i|,$$

where $\beta_j = \sum_i a_{ij}(f_i - \sum_{k \neq j} a_{ik} u_k)$. The optimal value for u_j is

$$\tilde{u}_j = \frac{1}{\|a_j\|_2^2} \operatorname{shrink}\left(\beta_j, \tfrac{1}{2\lambda}\right).$$

So this formula corrects the jth component of u, which strictly decreases the energy function $E(u)$.

Pathwise Coordinate Descent

Algorithm (Pathwise Coordinate Descent):

While "not converge"

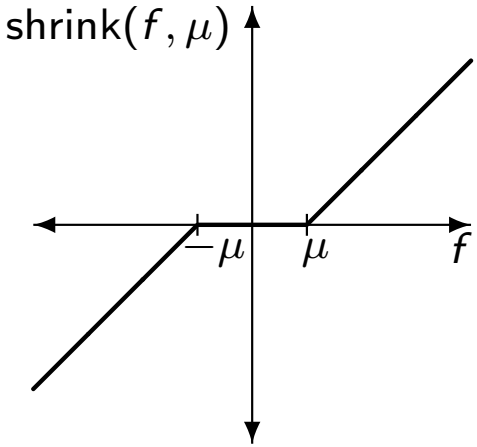
For $i = 1, 2, \ldots, n$,

$\beta_j \leftarrow \sum_i a_{ij}(f_i - \sum_{k \neq j} a_{ik} u_k)$

$u_j \leftarrow \frac{1}{\|a_j\|_2^2} \text{shrink}\left(\beta_j, \frac{1}{2\lambda}\right)$

where the shrink operator is:

$$\text{shrink}(f, \mu) = \begin{cases} f - \mu, & \text{if} \quad f > \mu; \\ 0, & \text{if} \quad -\mu \leq f \leq \mu; \\ f + \mu, & \text{if} \quad f < -\mu. \end{cases}$$



Adaptive Greedy Sweep

Consider the energy decrease by only updating the jth coordinate at once:

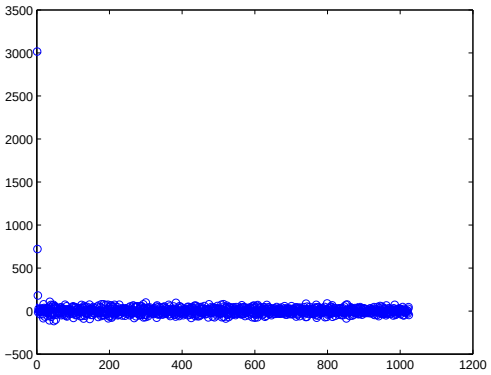
$$\begin{aligned}\Delta E_j &= E(u_1, \ldots, u_j, \ldots, u_n) - E(u_1, \ldots, \tilde{u}_j, \ldots, u_n) \\ &= \lambda \|a_j\|_2^2 \left(u_j - \frac{\beta_j}{\|a_j\|_2^2} \right)^2 + |u_j| - \lambda \|a_j\|_2^2 \left(\tilde{u}_j - \frac{\beta_j}{\|a_j\|_2^2} \right)^2 - |\tilde{u}_j| \\ &= \lambda \|a_j\|_2^2 (u_j - \tilde{u}_j) \left(u_j + \tilde{u}_j - \frac{2\beta_j}{\|a_j\|_2^2} \right) + |u_j| - |\tilde{u}_j|.\end{aligned}$$

We select the coordinate to update so as to maximize the decrease in energy,

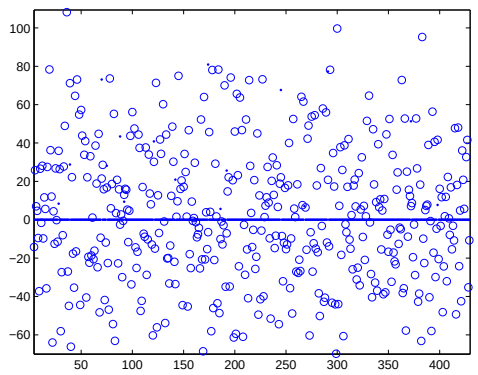
$$j^* = \arg\max_j \Delta E_j.$$

Greedy is Better

After one sequential sweep



zoom in



adaptive sweeps

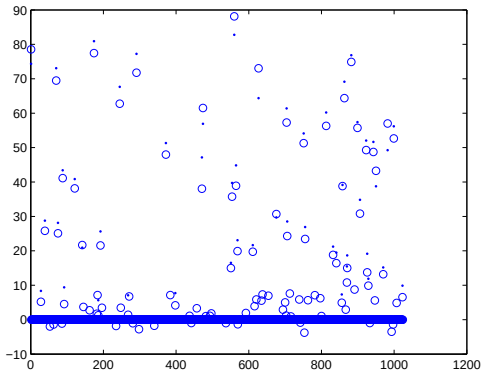


Figure: The dots represent the exact solution of the unconstrained problem at $\lambda = 1000$ and the circles are the results after 512 iterations. The left figure shows the result after one sequential sweep (512 iterations), the middle one is its zoom-in version and the right one shows the result by using an adaptive sweep.

Gradient-based Adaptive Sweep

For the same unconstrained problem, Tongtong Wu and Kenneth Lange update u_j giving the most negative value of directional derivatives along each forward or backward directions. For instance, if e_k is the coordinate direction along with u_k varies, the the objective function $E(u)$ has directional derivatives:

$$d_{e_k}E = \lim_{\sigma \downarrow 0} \frac{E(u+\sigma e_k) - E(u)}{\sigma} = 2\lambda(\|a_k\|_2^2 u_k - \beta_k) + \begin{cases} 1, & u_k \geq 0; \\ -1, & u_k < 0. \end{cases}$$

$$d_{-e_k}E = \lim_{\sigma \downarrow 0} \frac{E(u-\sigma e_k) - E(u)}{\sigma} = -2\lambda(\|a_k\|_2^2 u_k - \beta_k) + \begin{cases} -1, & u_k > 0; \\ 1, & u_k \leq 0. \end{cases}$$

We choose the updating coordinate

$$j^* = \arg\min_j \{d_{e_j}E, d_{-e_j}E\}.$$

Update Difference Based

We use the following way for choosing the updating coordinate based on the update difference:

$$j^* = \arg\max_j |u_j - \tilde{u}_j|.$$

Greedy Selection Choices

- Energy function based: $j = \arg\max_i \Delta E_i$;
- Gradient-based: $j = \arg\min_i \{d_{e_i} E, d_{-e_i} E\}$;
- Update difference based: $j = \arg\max_i |u_i - \tilde{u}_i|$

Computational Tricks

Our algorithm involves a vector β,

$$\beta_j = (A^T f)_j - a_j^T Au + \|a_j\|_2^2 u_j.$$

Since u changes in only one coordinate each iteration, there is a computationally efficient way to update β without entirely recomputing it,

$$\beta^{k+1} - \beta^k = (u_p^{k+1} - u_p^k)(\|a_p\|_2^2 I - A^T A)e_p.$$

Algorithm, Coordinate Descent with a Refined Sweep

Precompute: $w_j = \|a_j\|_2^2$;

Normalization: $A(\cdot, i) = A(\cdot, i)/w_i$;

Initialization: $u = 0$, $\beta = A^T f$;

Iterate until convergence:

$\tilde{u} = \text{shrink}(\beta, \frac{1}{2\lambda})$;

$j = \arg\max_i |u_i - \tilde{u}_i|$,

$u_j^{k+1} = \tilde{u}_j$;

$\beta^{k+1} = \beta^k - |u_j^k - \tilde{u}_j|(A^T A)e_j$,

$\beta_j^{k+1} = \beta_j^k$.

Convergence of Greedy Coordinate Descent

Consider minimizing functions of the form

$$\min_{u_i \in \mathbb{R}} H(u_1, u_2, \ldots, u_n) = F(u_1, u_2, \ldots, u_n) + \sum_{i=1}^{n} g_i(u_i),$$

where F is differentiable and convex, and g_i is convex. Tseng (1988) proved pathwise coordinate descent converges to a minimizer of H.

Theorem

If $\lim_{u_i \to \infty} H(u_1, u_2, \ldots, u_n) = \infty$ *for any* i, *and* $|\frac{\partial^2 F}{\partial u_i \partial u_j}|_\infty \leq M$, *then the greedy coordinate descent method based on the selection rule:* $j = \arg\max_i \Delta H$ *converges to an optimal solution.*

Convergence of Greedy Coordinate Descent

Lemma

Consider function $H(x) = (ax - b)^2 + |x|$, *where* $a > 0$. *It is easy to know that* $\tilde{x} = \arg\min_x H(x) = \text{shrink}(\frac{b}{a}, \frac{1}{2a^2})$. *Now we claim that for any* x,

$$|x - \tilde{x}| \leq \frac{1}{a}\big(H(x) - H(\tilde{x})\big)^{1/2}.$$

Theorem

The greedy coordinate descent method with the selection rule: $j = \arg\max_i |u_i - \tilde{u}_i|$ *converges to an optimal solution of the following problem*

$$\min_{u \in \mathbb{R}^n} H(u) = |u|_1 + \lambda \|Au - f\|_2^2,$$

Comparison of Numerical Speed

$$\min_{u \in \mathbb{R}^n} \ |u|_1 + \lambda \|Au - f\|_2^2. \tag{6}$$

λ	Pathwise	(1)	(2)	(3)
0.1	43	0.33	0.24	0.18
0.5	42	0.31	0.22	0.17
1	49	0.26	0.19	0.16
10	242	0.56	0.42	0.33
100	1617	4.0	2.7	2.2

Table: Runtime in seconds of pathwise coordinate descent and adaptive sweep methods (1), (2), (3) for solving (6) when they achieve the same accuracy.

Another Application to Image Denoising

Exact

Noisy

Denoised

Solving ROF

The ROF (Rudin-Osher-Fatemi) model for image denoising is the following: find u satisfying

$$u = \arg\min_u \int_\Omega |\nabla u| \, dx + \lambda \|u - f\|_2^2$$

where f is the given noisy image and $\lambda > 0$ is a parameter.

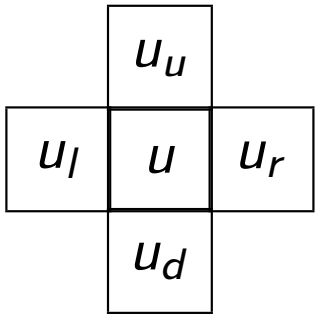
Methods for solving ROF:

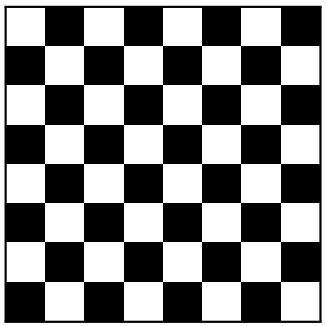
- Gradient descent (slow, time step restriction)
- Graph cuts (fast, but does not parallelize)
- Coordinate descent

Solving in Parallel by Coordinate Descent

$$\min_{u\in\mathbb{R}} E(u) = |u-u_l| + |u-u_r| + |u-u_u| + |u-u_d| + \lambda(u-f)^2 \qquad (7)$$

$$u_{opt} = \text{median}\left\{u_l, u_r, u_u, u_d, f+\frac{2}{\lambda}, f+\frac{1}{\lambda}, f, f-\frac{1}{\lambda}, f-\frac{2}{\lambda}\right\} \qquad (8)$$





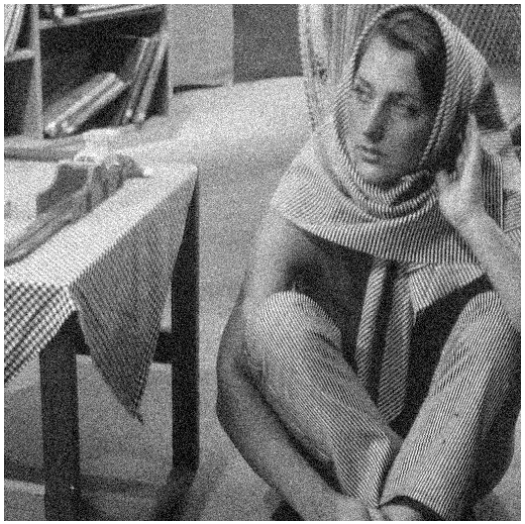
While not converged,

apply (8) to pixels in the black pattern in parallel ;

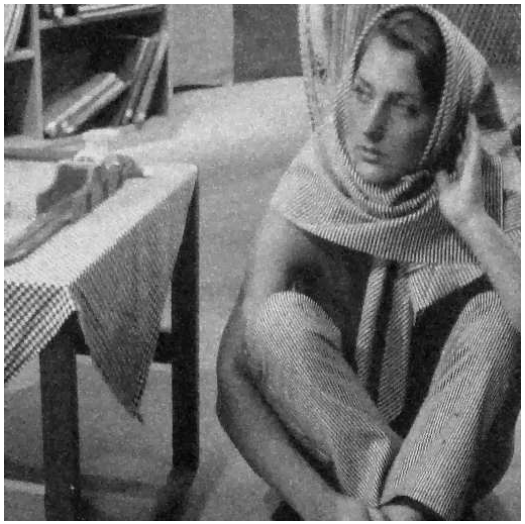
apply (8) to pixels in the white pattern in parallel.

Numerical Results

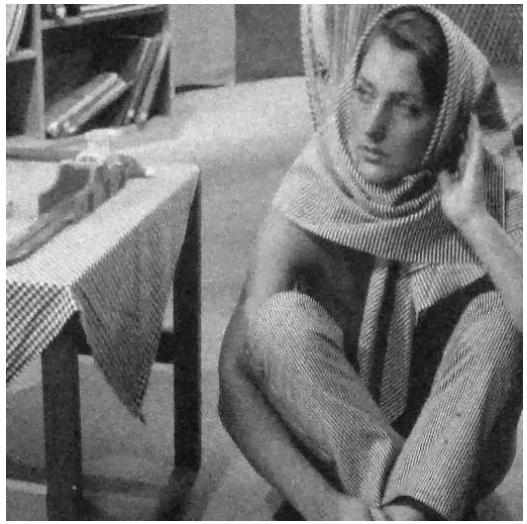
Input: SNR 8.8



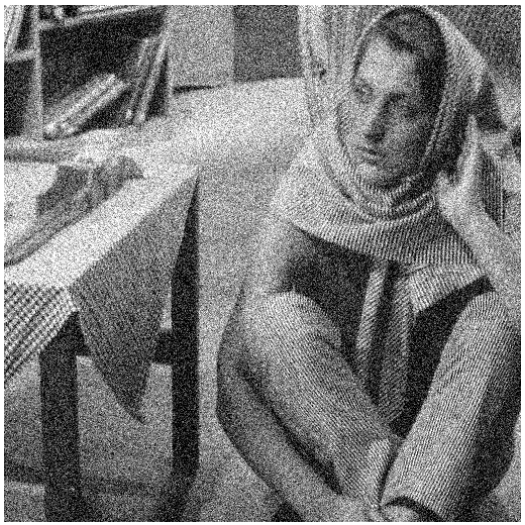
SNR 13.2 (1.10s)



SNR 13.5 (1.15s)



Input: SNR 3.1



SNR 9.5 (1.30s)

SNR 10.3 (1.27s)

Nonlocal Means (Buades, Coll and Morel)

Every two pixels x and y have a weight $w(x, y)$ used to evaluate the similarity of their patches. Define

$$w(x,y) = \exp\Big(-\frac{1}{h^2}\int_\Omega G_a(t)\,|f(x+t) - f(y+t)|^2\,dt\Big), \tag{9}$$

where G_a is a Gaussian with standard deviation a.

For computational efficiency, the support of $w(x, y)$ is often restricted to a "search window" $|x - y|_\infty \le R$ and set to zero outside.

Here, we consider the Nonlocal-ROF model,

$$\arg\min_u E(u, f) = \int w(x,y)\,|u(x) - u(y)|\,dx\,dy + \lambda\int (u-f)^2\,dx. \tag{10}$$

Generalization of the Median Formula

$$\begin{aligned}
&\arg\min_x \sum_{i=1}^{n} w_i|x-u_i| + \lambda|x-f|^\alpha \\
&= \text{median}\big\{u_1,\ u_2,\ \ldots,\ u_n,\ f+|w_n+\cdots+w_1|^p\,\mu, \\
&\quad f+\text{sign}(w_n+\cdots+w_2-w_1)|w_n+\cdots+w_2-w_1|^p\mu, \\
&\quad f+\text{sign}(w_n+\cdots+w_3-w_2-w_1)|w_n+\cdots+w_3-w_2-w_1|^p\mu, \\
&\quad \ldots,\ f+\text{sign}(w_n-w_{n-1}-\cdots-w_1)|w_n-w_{n-1}-\cdots-w_1|^p\mu, \\
&\quad f-|w_n+w_{n-1}+\cdots+w_1|^p\mu\big\}, \qquad (11)
\end{aligned}$$

where $p=\frac{1}{\alpha-1}$ and $\mu=(\frac{1}{\alpha\lambda})^p$. A computational savings is that the p_i do not depend on the u_i, so with all else fixed they only need to be computed once.

Numerical Results

5^2 search window(SNR 10.4)

7^2 (SNR 10.8)

11^2 (SNR 10.9)

The runtimes are 14s with the 5×5 search window and 20s with the 7×7 search window and 34s with 11×11 search window. (Parameters: $\lambda = 5 \times 10^{-3}$, $a = 1.25$, $h = 10.2$.)

Thanks for your attention!