

A Coordinate Gradient Descent Method for Sparse Optimization with ℓ_1 Regularization

Sangwoon Yun

Singapore-MIT Alliance, National University of Singapore

June 2, 2009

(Joint work with Kim-Chuan Toh (NUS), Paul Tseng (UW))

Outline

- Motivation
- General Problem Model: ℓ_1 -regularized Convex Minimization
- Coordinate Gradient Descent Method
- Convergence Results
- Numerical Experience
- Extension
- Conclusions & Future Work

Motivation

Compressed Sensing: ℓ_1 -regularized linear least squares (ℓ_1 LS) problem

Compressed Sensing is a new area of signal processing whose goal is to capture a signal/image with as few measurements as possible.

Sparse signal $\bar{x} \in \mathbb{R}^n$ can be recovered from relatively few incomplete measurements $b = A\bar{x}$ for a carefully chosen $A \in \mathbb{R}^{m \times n}$ ($m \leq n$ or $m \ll n$) by solving the ℓ_0 -minimization problem:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & \|x\|_0 \\ \text{subject to} & Ax = b. \end{array}$$

Sparse sol. \bar{x} of the above problem can be found by solving the basis pursuit problem (Candés et al. '06, Donoho '06):

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & \|x\|_1 \\ \text{subject to} & Ax = b, \end{array}$$

When \bar{x} is compressible but not sparse, or when measurements b are contaminated with noise, an appropriate norm of the residual $Ax - b$ is desirable

An appropriate model can be formulated as an quadratically constrained ℓ_1 -minimization problem:

$$\begin{aligned} \min_{x \in \mathcal{R}^n} \quad & \|x\|_1 \\ \text{subject to} \quad & \|Ax - b\|_2 \leq \sigma, \end{aligned}$$

where $\sigma > 0$ is related to the noise level.

Another frequently discussed model considers solving the above problem in Lagrangian form:

$$\min_{x \in \mathcal{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \mu \|x\|_1,$$

where $\mu > 0$. This is also known as the basis pursuit denoising problem.

Recent methods

- Homotopy methods (Osborne et al. '00, Donoho & Tsaig '06).
- Interior Point method (Kim et al. '07:l1-ls).
- Iterative Shrinkage/Thresholding (IST) method (Figueiredo and Nowak '03, Daubechies et al. '04).
- Gradient Projection method (Figueiredo et al. '07:GPSR).
- Fixed-Point Continuation (FPC) method (Hale et al. '07).

Classification: ℓ_1 -regularized logistic regression (ℓ_1 LR) problem

The logistic regression problem:

$$\min_{w \in \mathbb{R}^{n-1}, v \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-(w^T a_i + v b_i))),$$

where $a_i = b_i z_i$ and $(z_i, b_i) \in \mathbb{R}^{n-1} \times \{-1, 1\}$, $i = 1, \dots, m$ are a given set of (observed or training) examples, has been used for the feature selection in classification problems.

When m , the number of observations or training examples, is not large enough compared to $n - 1$, the number of feature variables, simple logistic regression leads to over-fit.

A standard technique to prevent over-fitting is regularization.

Recently, ℓ_1 -regularized logistic regression has received much attention:

$$\min_{w \in \mathbb{R}^{n-1}, v \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-(w^T a_i + v b_i))) + \mu \|w\|_1.$$

Recent methods

- Genkin et al. '06.
- Koh et al. '07: Interior Point (l1-logreg) method.
- Wright et al. '07 (SpaRSA).

General Problem Model: ℓ_1 -regularized Convex Minimization

P

$$\min_x F_\rho(x) := f(x) + \rho^T |x|$$

$f : \mathbb{R}^n \rightarrow \mathbb{R}$ is smooth (i.e., continuously differentiable) and convex.

ρ is a given nonnegative weight vector.

$|x|$ denotes obtained from x by taking absolute values.

- For ℓ_1 LS, $\rho_i = \mu$ for all i .
- For ℓ_1 LR, $\rho_i = \mu$ for all $i = 1, \dots, n - 1$ and $\rho_n = 0$.

Coord. Gradient Descent Method

Descent direction.

For $x \in \mathbb{R}^n$, choose $\mathcal{J} (\neq \emptyset) \subseteq \mathcal{N} = \{1, \dots, n\}$ and $H \succ 0_n$, Then solve

$$\min_{d|d_j=0 \forall j \notin \mathcal{J}} \left\{ \nabla f(x)^T d + \frac{1}{2} d^T H d + \rho^T |x + d| - \rho^T |x| \right\}$$

direc.
subprob

Let $d^H(x; \mathcal{J})$ and $q^H(x; \mathcal{J})$ be the opt. soln & obj. value of the direc. subprob.

Properties:

- $d^H(x; \mathcal{N}) = 0 \Leftrightarrow F'_\rho(x; d) \geq 0 \forall d \in \mathbb{R}^n$. stationarity
- H is diagonal $\Rightarrow d^H(x; \mathcal{J}) = \sum_{j \in \mathcal{J}} d^H(x; j)$, $q^H(x; \mathcal{J}) = \sum_{j \in \mathcal{J}} q^H(x; j)$. separab.
- $q^H(x; \mathcal{J}) \leq -\frac{1}{2} d^T H d$ where $d = d^H(x; \mathcal{J})$.

Stepsize: Armijo rule

Choose α to be the largest element of $\{\beta^k\}_{k=0,1,\dots}$ satisfying

$$F_\rho(x + \alpha d) - F_\rho(x) \leq \sigma \alpha q^H(x; \mathcal{J}) \quad (0 < \beta < 1, 0 < \sigma < 1).$$

For the ℓ_1 -regularized linear least squares problem, the minimization rule

$$\alpha \in \arg \min\{F_\rho(x + td) \mid t \geq 0\}$$

or the limited minimization rule

$$\alpha \in \arg \min\{F_\rho(x + td) \mid 0 \leq t \leq s\},$$

where $0 < s < \infty$, can also be used.

Choose \mathcal{J} :

- Gauss-Southwell- r rule:

$$\|d^D(x; \mathcal{J})\|_\infty \geq v \|d^D(x; \mathcal{N})\|_\infty$$

where $0 < v \leq 1$, $D \succ 0_n$ is diagonal (e.g., $D = \text{diag}(H)$).

- Gauss-Southwell- q rule:

$$q^D(x; \mathcal{J}) \leq v q^D(x; \mathcal{N}),$$

Where $0 < v \leq 1$, $D \succ 0_n$ is diagonal (e.g., $D = \text{diag}(H)$).

- This coord. grad. descent approach may be viewed as a hybrid of coordinate descent and IST (or fixed point) method.

- If H is diagonal, then subproblems can be solved in parallel:

$$d^H(x; \mathcal{N})_j = -\text{median}\{(\nabla f(x)_j - \rho_j)/H_{jj}, x_j, (\nabla f(x)_j + \rho_j)/H_{jj}\}.$$

Convergence Results

Global convergence

If

- $0 \prec \underline{\lambda}I \preceq D, H \preceq \bar{\lambda}I,$
- α is chosen by Armijo rule,

then every cluster point of the x -sequence generated by CGD method using GS- r or GS- q rule is a stationary point of F_ρ .

Local convergence rate In addition, if f satisfy **any** of the following assumptions, then the x -sequence generated by CGD method using GS- q rule converges at R-linear rate.

C1: f is strongly convex, ∇f is Lipschitz cont.

C2: f is quadratic.

C3: $f(x) = g(Ex) + q^T x$, where $E \in \mathfrak{R}^{m \times n}$, $q \in \mathfrak{R}^n$, g is strongly convex, ∇g is Lipschitz cont. on \mathfrak{R}^m .

Numerical Experience

1. ℓ_1 -regularized linear least squares problem (Compressed Sensing):

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \bar{\mu} \|x\|_1 \quad (\bar{\mu} > 0) \quad (1)$$

- Implement CGD method in Matlab.
- Choose H as a constant multiple of the identity matrix

$$H = \theta I,$$

Initially, we set $\theta^{\text{init}} = \|Au\|_2^2$ where u is a random unit vector. Then updated as follows:

$$\theta^{\text{new}} = \begin{cases} \max\{\theta/\alpha, 1\} & \text{if } \alpha > 10 \\ \min\{\theta/\alpha, 1\} & \text{if } \alpha < 10^{-1} \\ \theta & \text{otherwise.} \end{cases}$$

- Choose \mathcal{J} by Gauss-Southwell- r rule,

$$\mathcal{J} = \{j \mid |d_j^H(x; \mathcal{N})| \geq v \|d^H(x; \mathcal{N})\|_\infty\}.$$

Or by Gauss-Southwell- q rule,

$$\mathcal{J} = \left\{j \mid q^H(x; j) \leq v \min_i q^H(x; i)\right\}.$$

- Choose α by the minimization rule.

$$\alpha \in \arg \min \left\{ \frac{a_1}{2} t^2 - a_2 t + c \sum_{j \in \mathcal{J}} |x_j + t d_j| + a_3 \mid t > 0 \right\},$$

where $a_1 = \|Ad\|_2^2$, $a_2 = (Ad)^T(b - Ax)$, $a_3 = \frac{1}{2}\|b - Ax\|_2^2$, and $d = d^H(x; \mathcal{J})$.

- Termination Criterion:

$$\|Hd^H(x; \mathcal{N})\|_\infty \leq 10^{-3} \quad \text{or} \quad \frac{\|Hd^H(x; \mathcal{N})\|_\infty}{\max\{1, \|x\|_\infty\}} \leq 10^{-3}$$

- Continuation Strategy: solving a sequence of ℓ_1 LS problems defined by a decreasing sequence $\{\mu^0, \mu^1, \dots, \mu^\ell = \bar{\mu}\}$.

Initially, we set $\mu^0 = 0.01\|A^T b\|_\infty$ and update $\mu^{\text{new}} = \max\{0.25\mu, \bar{\mu}\}$ whenever the following criterion is satisfied:

$$\frac{\|Hd^H(x; \mathcal{N})\|_\infty}{\max\{1, \|x\|_\infty\}} \leq \max\{10^{\lfloor \log(\mu) \rfloor}, 10^{-3}\}.$$

Gaussian matrix:

- $A \in \mathbb{R}^{m \times n}$ is obtained by filling it with independent samples of the Gaussian distribution and then orthonormalizing the rows.

$b = Ax_s + \xi$, where ξ is Gaussian white noise with variance $(0.01\|Ax_s\|_2)^2$.

- Comparison with I1-Is (interior-point), GPSR-BB (gradient projection), and FPC-BB (fixed-point continuation).
- Require only matrix-vector mults. involving A and A^T at each iteration.

Matrix-vector mults. involving A^T cost $O(mn)$ ops. (same as the computational cost of other algorithms).

But matrix-vector mults. involving A cost $O(m|\mathcal{J}|)$ ops. (only need to evaluate Ad).

Sorting is needed to find the stepsize, the cost is $O(|\mathcal{J}| \ln |\mathcal{J}|)$.

Test Results

- To perform this comparison, first ran l1-ls and then each of the others until each reached the same objective value reached by l1-ls (10 random instances).

	l1-ls	CGD-GS-q	CGD-GS-r	GPSR-BB	FPC-BB
$n = 4096, m = 1024, \bar{\mu} = 0.05 \ A^T b\ _\infty$					
mean iterations	12	17	19	33	39
mean nnz(x)	474	214	215	302	323
mean CPU time	6.2e+00	2.6e-01	3.0e-01	8.8e-01	1.0e+00
mean error	1.3e-01	1.5e-01	1.5e-01	1.5e-01	1.4e-01
$n = 4096, m = 1024, \bar{\mu} = 0.01 \ A^T b\ _\infty$					
mean iterations	13	48	44	125	59
mean nnz(x)	559	480	602	600	616
mean CPU time	8.4e+00	7.3e-01	6.6e-01	3.3e+00	1.6e+00
mean error	3.3e-02	4.7e-02	4.5e-02	4.4e-02	3.5e-02
$n = 4096, m = 1024, \bar{\mu} = 0.005 \ A^T b\ _\infty$					
mean iterations	13	57	53	1001	60
mean nnz(x)	569	670	859	4090	788
mean CPU time	9.8e+00	8.8e-01	8.0e-01	2.6e+01	1.6e+00
mean error	2.1e-02	2.8e-02	2.8e-02	7.3e-01	3.0e-02

the original signal x_s contains 160 randomly placed ± 1 spikes.

	l1-ls	CGD-GS-q	CGD-GS-r	GPSR-BB	FPC-BB
$n = 4096, m = 1024, \mu = 0.05 \ A^T b\ _\infty$					
mean iterations	16	9	11	22	21
mean nnz(x)	125	77	75	121	124
mean CPU time	3.8e+00	1.6e-01	1.8e-01	6.0e-01	5.8e-01
mean error	1.2e-01	1.3e-01	1.3e-01	1.2e-01	1.3e-01
$n = 4096, m = 1024, \mu = 0.01 \ A^T b\ _\infty$					
mean iterations	14	11	13	56	32
mean nnz(x)	208	113	114	306	347
mean CPU time	5.2e+00	1.9e-01	2.2e-01	1.5e+00	8.9e-01
mean error	3.1e-02	3.3e-02	3.3e-02	3.8e-02	3.5e-02
$n = 4096, m = 1024, \mu = 0.005 \ A^T b\ _\infty$					
mean iterations	13	13	15	103	34
mean nnz(x)	297	128	145	516	648
mean CPU time	6.1e+00	2.1e-01	2.4e-01	2.7e+00	9.2e-01
mean error	2.0e-02	2.1e-02	2.2e-02	2.7e-02	2.9e-02

the original signal x_s contained 50 randomly placed ± 0.1 spikes, 60 randomly placed ± 1 spikes, and 50 randomly placed ± 10 spikes.

Discrete Cosine Transform:

- $A \in \mathbb{R}^{m \times n}$ is a partial DCT matrix whose m rows were chosen at random (uniformly) from the $n \times n$ discrete cosine transform (DCT) matrix.

the original signal x_s contains 320 randomly placed ± 1 spikes.

$b = Ax_s + \xi$, where ξ is Gaussian white noise with variance $(0.01\|Ax_s\|_2)^2$.

- Comparison with l1-ls, GPSR-BB, and FPC-BB.
- Require only matrix-vector mults. involving A and A^T at each iteration.

By using fast transform operators, such as the FFT, matrix-vector mults. involving A and A^T cost $O(n \ln n)$ ops.

But we lose the advantage of “sparse” multiplication.

Test Results

- To perform this comparison, first ran l1-ls and then each of the others until each reached the same objective value reached by l1-ls (10 random instances).

	l1-ls	CGD-GS-q	CGD-GS-r	GPSR-BB	FPC-BB
$n = 8192, m = 2048, \bar{\mu} = 0.01 \ A^T b\ _\infty$					
mean iterations	13	46	46	119	58
mean nnz(x)	912	830	1090	917	1215
mean CPU time	2.6e+00	4.2e-01	4.1e-01	9.4e-01	4.3e-01
mean error	3.3e-02	4.4e-02	4.3e-02	3.8e-02	3.6e-02
$n = 8192, m = 2048, \bar{\mu} = 0.005 \ A^T b\ _\infty$					
mean iterations	13	55	54	1001	60
mean nnz(x)	1204	1199	1584	8179	1593
mean CPU time	2.9e+00	5.2e-01	4.9e-01	7.9e+00	4.6e-01
mean error	2.2e-02	2.8e-02	2.8e-02	7.2e-01	3.1e-02

2. ℓ_1 -regularized logistic regression problem:

$$\min_{w \in \mathbb{R}^{n-1}, v \in \mathbb{R}} \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-(w^T a_i + v b_i))) + \bar{\mu} \|w\|_1, \quad (\bar{\mu} > 0)$$

- Implement CGD method in Matlab.
- Choose H as a diagonal Hessian approximation

$$H = \text{diag} \left[\min \{ \max \{ \nabla^2 f(x)_{jj}, 10^{-10} \}, 10^{10} \} \right]_{j=1, \dots, n},$$

where $x = ((w)^T, v)^T$.

- Choose \mathcal{J} by Gauss-Southwell- r rule, or by Gauss-Southwell- q rule.

- Choose α by the Armijo rule.

- The CGD method is terminated when

$$\|Hd^H(x; \mathcal{N})\|_\infty \leq 10^{-6}.$$

- Numerical tests on three large two-class (sparse) data classification problems and on (dense) randomly generated problems.

- Comparison with l1-logreg (interior-point) and SpaRSA.

Test Results

- To perform this comparison, first ran l1-logreg and then each of the others until each reached the same objective value reached by l1-logreg.

	l1-logreg (10^{-4})	CGD-GS-q	CGD-GS-r	SpaRSA
leu	$n = 7130, m = 38, \bar{\mu} = 0.01\mu_{\max}$			
iterations	25	92	146	fail
obj value	3.14477e-02	3.14458e-02	3.14212e-02	
CPU time	1.1e+00	6.6e-01	1.0e+00	
rcv1	$n = 47237, m = 20242, \bar{\mu} = 0.01\mu_{\max}$			
iterations	28	118	105	191
obj value	2.12420e-01	2.12420e-01	2.12420e-01	2.12420e-01
CPU time	1.1e+01	7.5e+00	6.9e+00	7.5e+00
real-sim	$n = 20959, m = 72309, \bar{\mu} = 0.01\mu_{\max}$			
iterations	22	71	72	88
obj value	2.14289e-01	2.14289e-01	2.14287e-01	2.14289e-01
CPU time	1.4e+01	7.5e+00	7.9e+00	7.6e+00

$$\mu_{\max} = \frac{1}{m} \left\| \frac{m_-}{m} \sum_{b_i=1} a_i + \frac{m_+}{m} \sum_{b_i=-1} a_i \right\|_{\infty}$$

- Randomly generated problem (Features of positive (negative) examples are independent and identically distributed, dense).

	l1-logreg 10^{-4}	CGD-GS-q 10^{-6}	CGD-GS-r 10^{-6}
10 random	$n = 10001, m = 1000, \bar{\mu} = 0.01\mu_{\max}$		
mean iterations	20	230	260
mean CPU time	2.4e+02	1.2e+01	1.4e+01
10 random	$n = 1001, m = 100, \bar{\mu} = 0.01\mu_{\max}$		
mean iterations	17	187	220
mean CPU time	1.9e-01	2.4e-01	2.9e-01
10 random	$n = 1001, m = 10000, \bar{\mu} = 0.01\mu_{\max}$		
mean iterations	16	82	88
mean CPU time	2.3e+02	4.8e+00	5.2e+00
10 random	$n = 101, m = 1000, \bar{\mu} = 0.01\mu_{\max}$		
mean iterations	14	60	66
mean CPU time	1.9e-01	6.8e-02	9.0e-02

- The computational cost for the search direction of l1-logreg is $O(\min(n, m)^2 \max(n, m))$ ops. per iteration. In contrast, the computational cost of CGD is $O(mn)$ ops. per iteration.

Extension

Sparse Inverse Covariance Estimation

$$\min_{X \in \mathcal{S}_+^n} -\log \det X + \text{tr}(XS) + \mu \|X\|_1$$

$S \in \mathcal{S}_+^n$ is an empirical covariance matrix.

$$\|X\|_1 = \sum_{i,j=1}^n |X_{ij}|, \mu > 0.$$

- $-\log \det X + \text{tr}(XS)$ is strictly convex, cont. diff. on its domain \mathcal{S}_{++}^n , $O(n^3)$ ops. to evaluate. $\|\cdot\|_1$ is convex, nonsmooth. In applications, n can exceed 5000.

The Fenchel dual problem is a bound-constrained convex program:

$$\begin{aligned} \min_{W \in \mathcal{S}_+^n} \quad & -\log \det W \\ \text{s.t.} \quad & |(W - S)_{ij}| \leq \mu, \quad i, j = 1, \dots, n, \end{aligned}$$

- IP method requires $O(n^7 \log(1/\epsilon))$ ops. to find ϵ -optimal soln. Impractical! Nesterov's first-order smoothing method requires $O(n^4/\epsilon)$ ops. (Lu '07).
- Use coordinate descent method to solve the dual problem, cycling thru columns (and corresponding rows) $j = 1, \dots, n$ of W . Each iteration reduces (via determinant property & duality) to

$$\min_{w \in \mathbb{R}^{n-1}} \frac{1}{2} w^T W_{i^c i^c} w - S_{i^c i}^T w + \mu \|w\|_1.$$

Solve this using IP method $O(n^3)$ ops. or coordinate descent method (Friedman '07).

- Can apply CGD method with cyclic rule to either primal or dual problem. Applied to the primal, each iteration entails the following subproblem:

$$\min_{u \in \mathbb{R}^n} \left\{ \text{tr}((X^{-1} + S)D) + \frac{1}{2}u^T H u + \mu \|X + D\|_1 \right\}_{D=ue_j^T + e_j u^T}.$$

For diagonal H , the above problem has a closed form solution.

And each iteration requires 1 gradient (X^{-1}) evaluation to find the direction: this can be updated by using the Sherman-Woodbury-Morrison formula on two rank,

and 1 evaluation for the inverse of the $(n - 1) \times (n - 1)$ submatrix of X to find the stepsize: this can be evaluated by using the Schur complement.

Both computations require $O(n^2)$ operations.

Conclusions & Future Work

1. Numerical results shows that CGD method is efficient not only in minimizing a quadratic convex smooth function with ℓ_1 -regularization but also in minimizing a nonquadratic convex smooth function with ℓ_1 -regularization.
2. The Barzilai-Borwein steps accelerate the convergence of GPSR and FPC. But the performance of CGD method using BB steps was generally worse than that of CGD method using the minimization rule when it was applied for solving ℓ_1 -regularized linear least squares problems. Can the efficiency be further improved by using nonmonotone descent?
3. When CGD method with continuation was applied to solve ℓ_1 -regularized logistic regression problems, its overall performance was worse than that of CGD method without continuation. Can other acceleration techniques be developed for solving ℓ_1 -regularized logistic regression problems?

Thank you!

Tseng, P. and Yun S., A coordinate gradient descent method for nonsmooth separable minimization.

Yun S. and Toh K.-C., A coordinate gradient descent method for ℓ_1 -regularized convex minimization.

(PDF file available at <http://www.math.nus.edu.sg/matys/index.html>)