A Coordinate Gradient Descent Method for Nonsmooth Separable Minimization

Sangwoon Yun
Mathematics, University of Washington
Seattle, WA, USA

ISMP 2006
July 31, 2006
(Joint work with Paul Tseng)

Talk Outline

- Basic Problem Model (Motivation)
- General Problem Model
- Coordinate Gradient Descent Method
- Convergence Results
- Numerical Results
- Conclusions & Future Work

Basic Problem Model (Motivation)

Box-constrained optimization problem

$$\min_{l \le x \le u} \quad f(x),$$

where $f : \Re^N \to \Re$ is smooth, $l \le u$ (possibly with $-\infty$ or ∞ components).

ℓ_1- regularization

Find x so that $Ax - b \approx 0$ and x has "few" nonzeros.

Formulate this as an unconstrained convex optimization problem:

$$\min_{x \in \Re^n} \|Ax - b\|_2^2 + c\|x\|_1 \qquad (c > 0)$$

Ex: the "Basis Pursuit" model for signal denoising.

General Problem Model

P

$$\min_x F_c(x) := f(x) + cP(x) \qquad (c \geq 0)$$

$f : \Re^N \to \Re$ is smooth.

$P : \Re^N \to (-\infty, \infty]$ is proper, convex, lsc, and $P(x) = \sum_{j=1}^n P_j(x_j)$ ($x = (x_1, ..., x_n)^T$).

- $P(x) = \begin{cases} 0 & \text{if } l \leq x \leq u \\ \infty & \text{else} \end{cases}$
- $P(x) = \|x\|_1$

Previous methods

- Fukushima and Mine (81) proposed a proximal gradient descent method which computes a direction $\bar{d}$ as the solution of the subproblem

  $$\min_d \ \nabla f(x)^T d + \tfrac{1}{2}\rho\|d\|^2 + cP(x+d) \qquad (\rho > 0)$$

  and shows local linear convergence under the assumption that $\nabla^2 f(x^*)$ is positive definite where x^* is a stationary point.

- Auslender (78)

- Mine and Fukushima (81)

- Kiwiel (86)

- Fukushima (91)

The above studies did not present numerical results.

Coord. Gradient Descent Method(new)

Descent Direction.

For $x \in \mathrm{dom}P$, choose $J(\neq \emptyset) \subseteq \{1, ..., n\}$ and $H \succ 0_n$, Then solve

$$\min_{d|d_j=0\ \forall i \notin J} \{g^T d + \frac{1}{2} d^T H d + cP(x+d) - cP(x)\}$$

direc. subprob

with $g = \nabla f(x)$.

Facts: Let $d_H(x; J)$ and $q_H(x; J)$ be the opt. soln and obj. value of the direc. subprob.

- $d_H(x; \{1, ..., n\}) = 0 \Leftrightarrow F_c'(x; d) \geq 0\ \forall d \in \Re^N$. stationarity
- H is diagonal $\Rightarrow d_H(x; J) = \sum_{j \in J} d_H(x; j)$, $q_H(x; J) = \sum_{j \in J} q_H(x; j)$. separab.
- $q_H(x; J) \leq -\frac{1}{2} d^T H d$ where $d = d_H(x; J)$

Stepsize: **Armijo rule**

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

$F_c(x + \alpha d) - F_c(x) \leq \sigma \alpha q_H(x; J)$ $(0 < \beta < 1, 0 < \sigma < 1)$.

Choose J:

- Gauss-Southwell-q (new):

$q_D(x; J) \leq \upsilon\, q_D(x; \{1, ..., n\})$ $(0 < \upsilon \leq 1, D \succ 0_n$ is diagonal, e.g., $D = \mathrm{diag}(H))$.

Remarks:

Alternative rules for choice of J

- Gauss-Seidel :

J cycles through $\{1\}, \{2\}, ..., \{n\}$.

- Gauss-Southwell-d :

$\|d_D(x; J)\|_\infty \geq \upsilon \|d_D(x; \{1, ..., n\})\|_\infty$ $(0 < \upsilon \leq 1, D \succ 0_n$ is diagonal).

Advantage

- CGD method is simple, highly parallelizable, and is suited for solving large-scale problems.

- CGD not only has cheaper iterations than exact coordinate descent, it also has stronger global convergence properties.

Convergence Results

Global Convergence If

- $0 < \underline{\lambda} \leq \lambda_i(D), \lambda_i(H) \leq \bar{\lambda}\ \forall i$,
- α is chosen by Armijo rule,
- J is chosen by G-Southwell-q (or G-Seidel or G-Southwell-d),

then every cluster point of the x-sequence generated by CGD method is a stationary point of F_c.

Local Convergence Rate If

- $0 < \underline{\lambda} \leq \lambda_i(D), \lambda_i(H) \leq \bar{\lambda}\ \forall i$,
- α is chosen by Armijo rule,
- J is chosen by G-Southwell-q (or G-Seidel),

in addition, if P and f satisfy any of the following assumptions, then the x-sequence generated by CGD method converges at R-linear rate.

C1 f is strongly convex, ∇f is Lipschitz cont. on $\mathrm{dom}P$.

C2 f is (nonconvex) quadratic. P is polyhedral.

C3 $f(x) = g(Ex) + q^T x$, where $E \in \Re^{m\times N}$, $q \in \Re^N$, g is strongly convex, ∇g is Lipschitz cont. on $\Re^m$. P is polyhedral.

C4 $f(x) = \max_{y\in Y}\{(Ex)^T y - g(y)\} + q^T x$, where $Y \subseteq \Re^m$ is polyhedral, $E \in \Re^{m\times N}$, $q \in \Re^N$, g is strongly convex, ∇g is Lipschitz cont. on $\Re^m$. P is polyhedral.

Notes:

Proof of convergence rate uses a local error bound

- Error Bound

$$\mathrm{dist}(x, X^*) \leq \kappa \|d_I(x)\|_2 \,\mathrm{whenever}\, \|d_I(x)\|_2 \leq \epsilon,$$

for some $\kappa > 0$, $\epsilon > 0$, where X^* denotes the set of stationary points of F_c and $\mathrm{dist}(x, X^*) = \min_{x^* \in X^*} \|x - x^*\|_2$ and $d_I(x) = d_I(x; \{1, ..., n\})$.

Numerical Results

- Implement CGD method in Matlab.
- Diagonal Hessian approximation

$$H = \operatorname{diag}\left[\min\{\max\{\nabla^2 f(x)_{jj}, 10^{-2}\}, 10^9\}\right]_{j=1,\ldots,n}.$$

- The index subset J by the Gauss-Southwell-q rule,

$$J = \left\{ j \mid q_H(x; j) \leq \upsilon \min_i q_H(x; i) \right\}.$$

- The stepsize α is chosen by the Armijo rule.

- The CGD method is terminated when

$$\|Hd_H(x;\{1,...,n\})\|_\infty \leq 10^{-4}.$$

- Numerical tests with f from Moré-Garbow-Hillstrom set, $P(x) = \|x\|_1$, and different c (e.g., $c = .1, 1, 10$).

- Comparison with MINOS 5.5.1 (Murtagh, Saunders '05) and L-BFGS-B (Zhu, Byrd, Nocedal '97) applied to a reformulation of *P* as a bound-constrained smooth optimization problem:

$$\min_{y\geq 0, z\geq 0} \; f(y-z) + c\, e^T(y+z),$$

where e is the vector of 1s.

Description of test functions from Moré-Garbow-Hillstrom set

Name	n	Description
DBV	1000	(28), nonconvex, with sparse Hessian.
ER	1000	(21), nonconvex, with sparse Hessian.
EPS	1000	(22), convex, with sparse Hessian.
LR1	1000	(33), convex, with dense Hessian.
LFR	1000	(32), strongly convex, with dense Hessian.
VD	1000	(25), strongly convex, with dense Hessian.

Acceleration Techniques

First one

- Use an active-set identification strategy (Facchinei, Fischer, and Kanzow '98) to estimate which components of x would be nonzero at a solution and update these components by L-BFGS method.

- The above procedure is invoked 50 times every 50 consecutive CGD iterations.

Second one

- Solve the subproblem with rank-1 Hessian

$$\min_d \; g^T d + \frac{1}{2}(h^T d)^2 + c\|x+d\|_1.$$

  where h satisfies the rank-1 secant equation $(hh^T)s = y$, with $s = x - x^{prev}$ and $y = g - g^{prev}$.

- If an optimal solution exists, then it has at most one nonzero component (computed efficiently using Matlab's vector operations). Let d^{r1} be the optimal solution.

- Update x along the direction d^{r1} by the Armijo rule as in CGD if d^{r1} is a descent direction.

- The above procedure is invoked once every 10 consecutive CGD iterations.

Test results ($x^0 = (1, 1, ..., 1)^T$)

Name	c	L-BFGS-B	MINOS	CGD-GS-q-acc
		♯nz/obj/cpu	♯nz/obj/cpu	♯nz/obj/cpu
DBV	.1	[a]999/83.4557/.01	0/0.00000/51.5	0/0.00000/.5
	10	0/0.00000/.00	0/0.00000/52.5	0/0.00000/.01
ER	1	1000/436.250/.1	1000/436.250/71.5	1000/436.250/.1
	100	0/500.000/.00	0/500.000/52.4	0/500.000/.03
EPS	1	[a]999/352.526/.05	1000/351.146/60.3	1000/351.146/.3
	100	0/1250.00/.01	0/1250.00/51.5	0/1250.00/.01
LR1	.1	[a]1000/424.663/.00	[b]2/249.625/59.7	1/249.625/.1
	10	[a]1000/17753.4/.01	1/249.625/58.0	1/249.625/.05
LFR	.1	1000/98.5000/.00	1000/98.5000/77.2	1000/98.5000/.01
	10	0/1001.00/.00	0/1001.00/53.3	0/1001.00/.01
VD	1	[a]1000/1000.00/.00	1000/937.594/43.0	1000/937.594/.5
	100	[a]996/75135.5/.2	136/55043.1/57.4	[c]1000/55043.1/88.1

[a] L-BFGS-B exited due to the objective value cannot be improved upon.

[b] MINOS exited due to the current point cannot be improved upon.

[c] CGD exited due to the Armijo stepsize in an L-BFGS acceleration step reaching 10^{-30}.

- CGD with acceler. steps is competitive with MINOS in terms of solution accuracy and is generally faster in terms of cpu time (except on VD).

- L-BFGS-B is fast, but often exits when still far from a solution.

Conclusions & Future Work

1. The method may be viewed as a hybrid of gradient-projection and SOR methods, or as a block-coordinate version of descent methods.

2. Numerical results shows the practical efficiency of the method.

3. In our current implementation of the CGD method, we used diagonal H. How about block-diagonal H?

4. How would the CGD method perform on bound-constrained problems?

5. Can the CGD method be extended to handle linear equality constraints (as arises in SVM) or, more generally, smooth equality constraints?

Thank you!