Gauss-Seidel for Constrained Nonsmooth Optimization and Applications

Paul Tseng
Mathematics, University of Washington
Seattle

UBC
April 10, 2007

(Joint works with Sylvain Sardy (EPFL) and Sangwoon Yun (UW))

Talk Outline

- (Block) Coordinate Minimization
- Application to Basis Pursuit
- Block Coordinate Gradient Descent
  - ⋆ Convergence
  - ⋆ Numerical Tests
- Applications to group Lasso regression and SVM
- Conclusions & Future Work

Coordinate Minimization

$$\min_{x=(x_1,\ldots,x_n)} f(x)$$

$f : \Re^n \rightarrow \Re$ is convex, cont. diff.

Given $x \in \Re^n$, choose $i \in \{1, ..., n\}$. Update

$$\boxed{x^{\text{new}} = \arg\min_{u | u_j = x_j \ \forall j \neq i} f(u).}$$

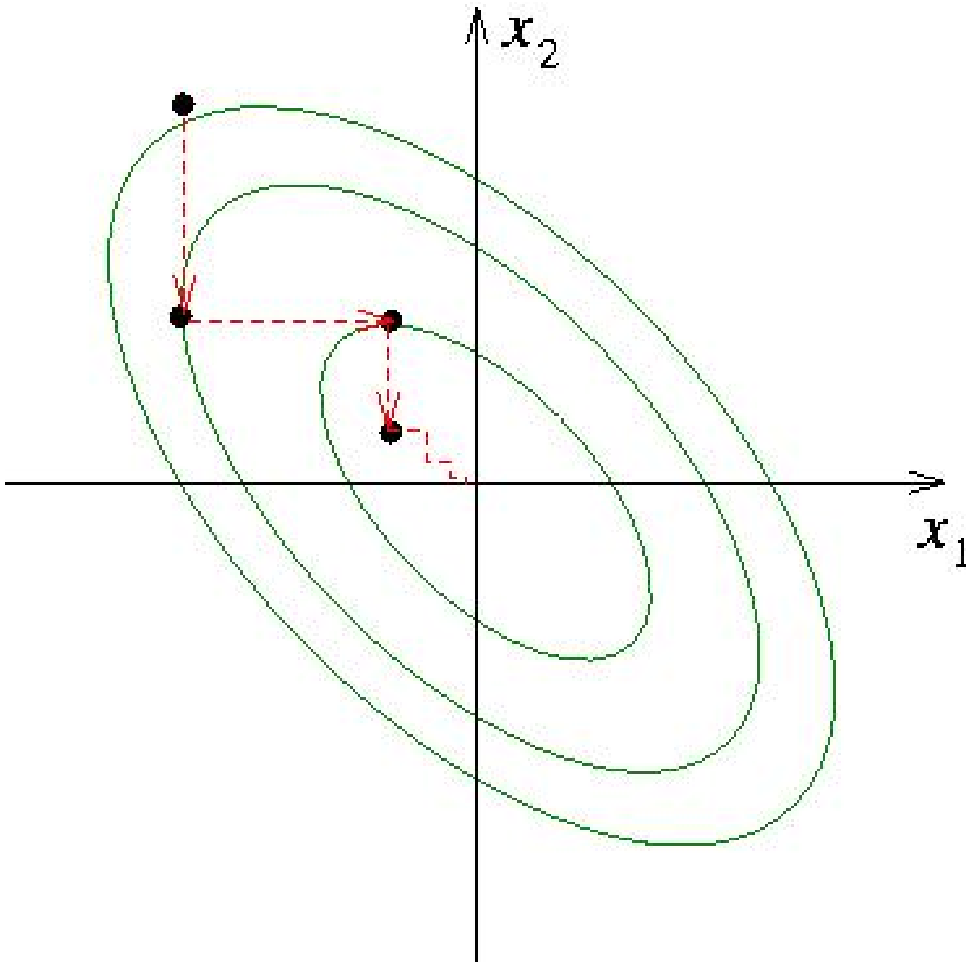
Repeat until "convergence".

Gauss-Seidel: Choose i cyclically, 1, 2,..., n, 1, 2,...

Gauss-Southwell: Choose i with $|\frac{\partial f}{\partial x_i}(x)|$ maximum.

Example:

$$\min_{x=(x_1,x_2)} (x_1+x_2)^2 + \frac{1}{4}(x_1-x_2)^2$$



- This method extends to update a block of coordinates at each iteration.

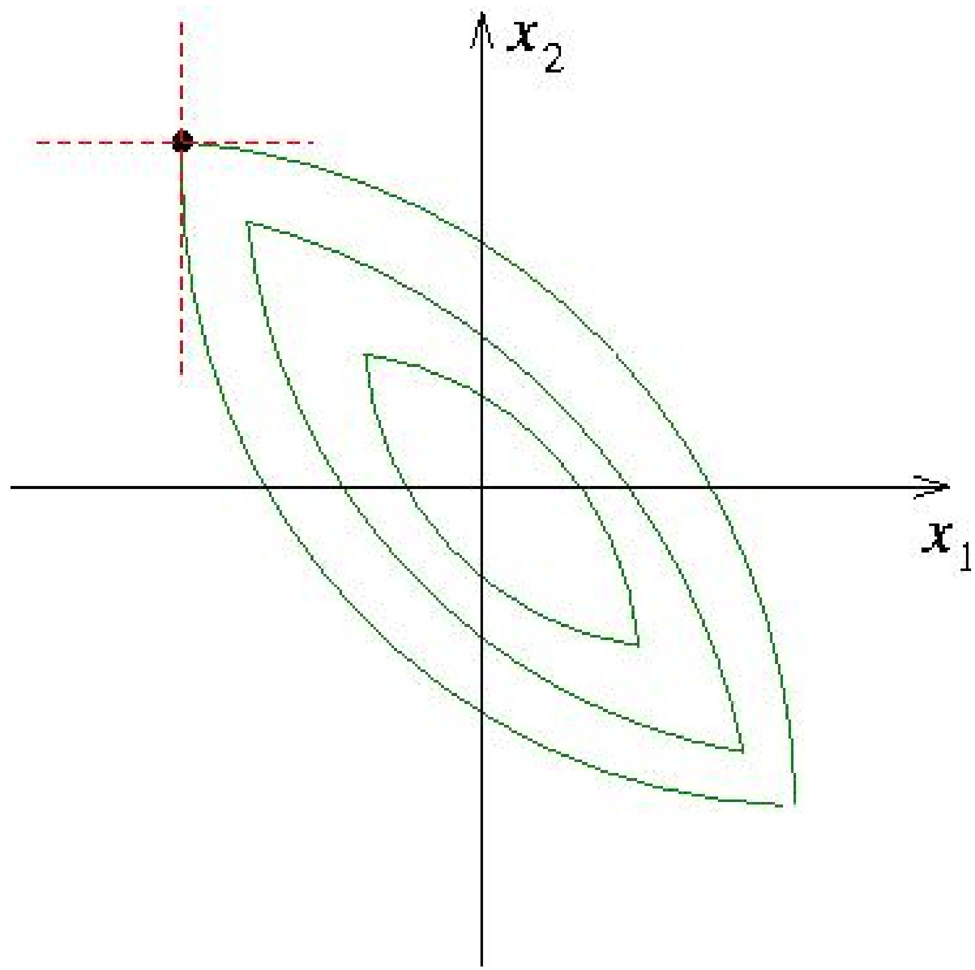
- It is simple, and efficient for large “weakly coupled” problems (off-diagonals of $\nabla^2 f(x)$ not too large).

- Every cluster point of the x-sequence is a minimizer. Zadeh ’70

- If f is nonconvex, then G-Seidel can cycle Powell ’73 but G-Southwell still converges.

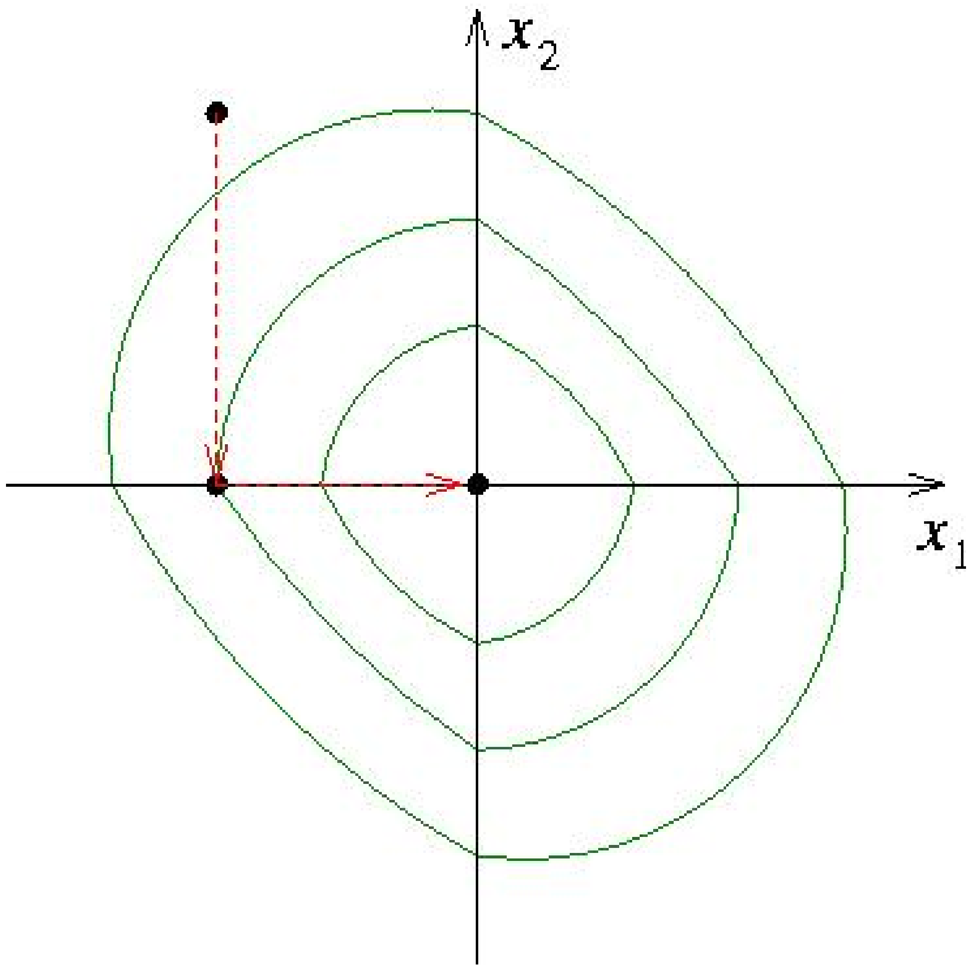
- Can get stuck if f is nondifferentiable.

Example: $$\min_{x=(x_1,x_2)} (x_1+x_2)^2 + \frac{1}{4}(x_1-x_2)^2 + |x_1+x_2|$$



But, if the nondifferentiable part is *separable*, then convergence is possible.

Example: $$\min_{x=(x_1,x_2)} (x_1+x_2)^2 + \frac{1}{4}(x_1-x_2)^2 + |x_1| + |x_2|$$



Block Coord. Minimization for Basis Pursuit

$$\boxed{\min_x F_c(x) := \|Ax - b\|_2^2 + c\|x\|_1}$$

"Basis Pursuit"

$A \in \Re^{m \times n}$, $b \in \Re^m$, $c \geq 0$.

- Typically $m \geq 1000$, $n \geq 8000$, and A is dense. $\|\cdot\|_1$ is nonsmooth.
- Can reformulate this as a convex QP and solve using an IP method. Chen, Donoho, Saunders '99

Assume the columns of A come from an overcomplete set of basis functions associated with a fast transform (e.g., wavelet packets).

BCM for BP:

Given x, choose $\mathcal{I} \subseteq \{1, ..., n\}$ with $|\mathcal{I}| = m$ and $\{A_i\}_{i\in\mathcal{I}}$ orthog. Update

$$\boxed{x^{\text{new}} = \operatorname*{arg\,min}_{u_i = x_i \ \forall i \notin \mathcal{I}} F_c(u)}$$

$\longleftarrow$ has closed form soln

Repeat until "convergence".

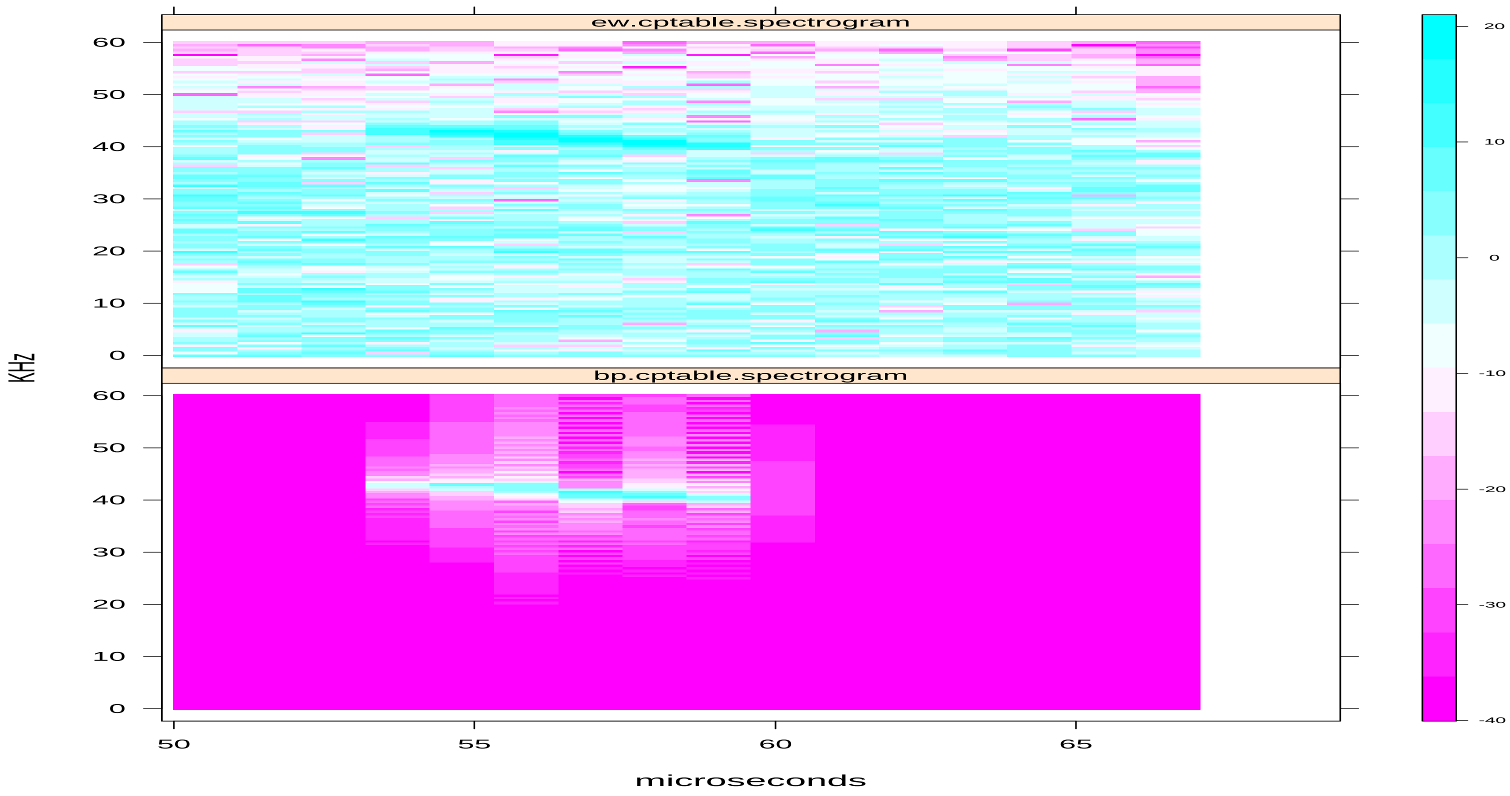
Gauss-Southwell: Choose $\mathcal{I}$ to maximize $\min_{v \in \partial_{x_\mathcal{I}} F_c(x)} \|v\|_2$.

- Finds $\mathcal{I}$ in $O(n + m \log m)$ opers. by algorithm of Coifman & Wickerhauser.
- The x-sequence is bounded & each cluster point is a minimizer. Sardy, Bruce, T '00

Convergence of BCM depends crucially on

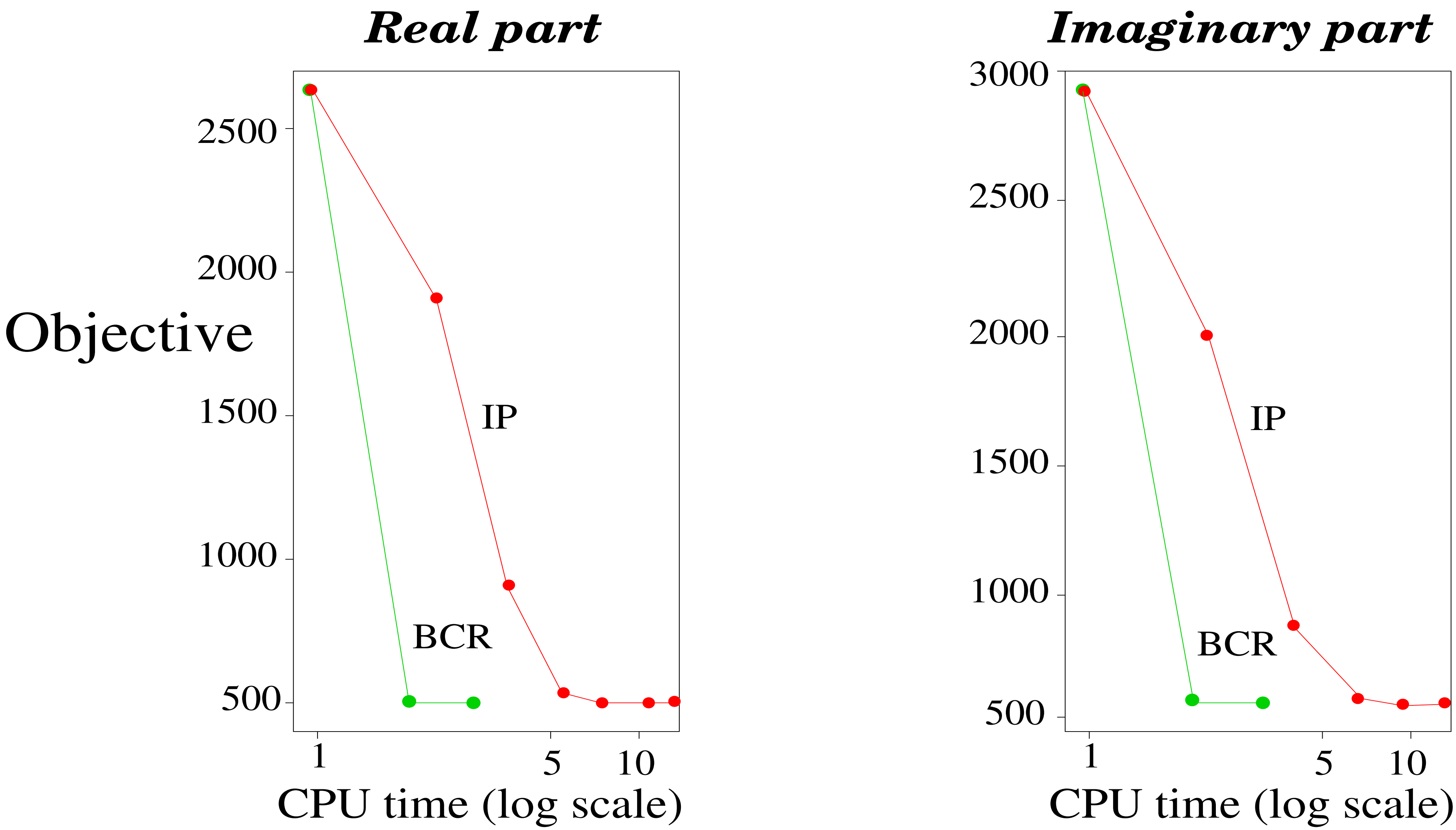
- differentiability of $\|\cdot\|_2^2$
- separability of $\|\cdot\|_1$
- convexity $\Rightarrow$ global minimum

Application: Electronic surveillance



$m = 2^{11} = 2048$, $c = 4$, local cosine transform, all but 4 levels

Method efficiency:



Comparing CPU times of IP and BCM (S-Plus, Sun Ultra 1).

Generalization to ML Estimation

$$\boxed{\min_x -\ell(Ax; b) + c \sum_{i \in \mathcal{J}} |x_i|}$$

ℓ is log likelihood, $\{A_i\}_{i \notin \mathcal{J}}$ are lin. indep "coarse-scale Wavelets", $c \geq 0$

- $-\ell(y; b) = \frac{1}{2}\|y - b\|_2^2$ Gaussian noise
- $-\ell(y; b) = \sum_{i=1}^{m} (y_i - b_i \ln y_i) \quad (y_i \geq 0)$ Poisson noise

Can solve this problem by adapting IP method. But IP method is slow (many CG steps per IP iteration).

Adapt BCM method?

General Problem Model

P1

$$\boxed{\min_{x=(x_1,\ldots,x_n)} F_c(x) := f(x) + cP(x)}$$

$f : \Re^N \to \Re$ is cont. diff. ($N \geq n$). $c \geq 0$.

$P : \Re^N \to (-\infty, \infty]$ is proper, convex, lsc, and block-separable, i.e., $P(x) = \sum_{i=1}^n P_i(x_i)$ ($x_i \in \Re^{n_i}$).

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

Block Coord. Gradient Descent Method for *P1*

Idea: Do BCM on a quadratic approx. of f.

For $x \in \mathrm{dom}P$, $\mathcal{I} \subseteq \{1, ..., n\}$, and $H \succ 0_N$, let $d_H(x;\mathcal{I})$ and $q_H(x;\mathcal{I})$ be the optimal soln and obj. value of

$$\boxed{\min_{d|d_i=0\ \forall i \notin \mathcal{I}} \{\nabla f(x)^T d + \frac{1}{2} d^T H d + cP(x+d) - cP(x)\}}$$

direc. subprob

Facts:

- $d_H(x;\{1, ..., n\}) = 0 \ \Leftrightarrow F_c'(x;d) \geq 0 \ \forall d \in \Re^N$. stationarity
- H is diagonal $\Rightarrow d_H(x;\mathcal{I}) = \sum_{i \in \mathcal{I}} d_H(x;i)$, $q_H(x;\mathcal{I}) = \sum_{i \in \mathcal{I}} q_H(x;i)$. separab.

BCGD for P1:

Given $x \in \mathrm{dom}P$, choose $\mathcal{I} \subseteq \{1, ..., n\}$, $H \succ 0_N$. Let $d = d_H(x; \mathcal{I})$.

Update $$\boxed{x^{\text{new}} = x + \alpha d \qquad (\alpha > 0)}$$

until "convergence." ($d_H(x; \mathcal{I})$ has closed form when H is diagonal and $P(\cdot) = \|\cdot\|_1$.)

Gauss-Southwell-d: Choose $\mathcal{I}$ with $\|d_D(x; \mathcal{I})\|_\infty \geq \upsilon \|d_D(x; \{1, ..., n\})\|_\infty$ ($0 < \upsilon \leq 1$, $D \succ 0_N$ is diagonal, e.g., $D = I$ or $D = \mathrm{diag}(H)$).

Gauss-Southwell-q: Choose $\mathcal{I}$ with $q_D(x; \mathcal{I}) \leq \upsilon\, q_D(x; \{1, ..., n\})$.

Inexact Armijo LS: α = largest element of $\{s, s\beta, s\beta^2, ...\}$ satisfying

$$F_c(x + \alpha d) - F_c(x) \leq \sigma \alpha q_H(x; \mathcal{I})$$

($s > 0$, $0 < \beta < 1$, $0 < \sigma < 1$)

Convergence Results: (a) If $0 < \underline{\lambda} \le \lambda_i(D), \lambda_i(H) \le \bar{\lambda}\ \forall i$, then every cluster point of the x-sequence generated by BCGD method is a stationary point of F_c.

(b) If in addition P and f satisfy any of the following assumptions and $\mathcal{I}$ is chosen by G-Southwell-q, then the x-sequence 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.

- BCGD has stronger global convergence property (and cheaper iteration) than BCM.

Numerical Tests:

- Implement BCGD, with additional acceleration steps, in Matlab.

- Numerical tests on $\min_x f(x) + c\|x\|_1$ with f from Moré-Garbow-Hillstrom set (least square), and different c (e.g., $c = .1, 1, 10$). Initial $x = (1, \ldots, 1)$.

- Compared with L-BFGS-B (Zhu, Byrd, Nocedal '97) and MINOS 5.5.1 (Murtagh, Saunders '05), applied to a reformulation of *P1* with $P(x) = \|x\|_1$ as

$$\min_{\substack{x^+ \geq 0 \\ x^- \geq 0}} f(x^+ - x^-) + c\, e^T(x^+ + x^-).$$

- BCGD seems more robust than L-BFGS-B and faster than MINOS on avg (on a HP DL360 workstation, Red Hat Linux 3.5). However, MINOS is general NLP solver. L-BFGS-B is a bound constrained NLP solver.

$f(x)$	n	**Description**
BAL	1000	Brown almost-linear func, nonconvex, dense Hessian.
BT	1000	Broyden tridiagonal func, nonconvex, sparse Hessian.
DBV	1000	Discrete boundary value func, nonconvex, sparse Hessian.
EPS	1000	Extended Powell singular func, convex, 4-block diag. Hessian.
ER	1000	Extended Rosenbrook func, nonconvex, 2-block diag. Hessian.
LFR	1000	$f(x) = \sum_{i=1}^{n} \left(x_i - \frac{2}{n+1} \sum_{j=1}^{n} x_j - 1 \right)^2 + \left(\frac{2}{n+1} \sum_{j=1}^{n} x_j + 1 \right)^2,$ strongly convex, quad., dense Hessian.
VD	1000	$f(x) = \sum_{i=1}^{n} (x_i - 1)^2 + \left(\sum_{j=1}^{n} j(x_j - 1) \right)^2 + \left(\sum_{j=1}^{n} j(x_j - 1) \right)^4,$ strongly convex, dense ill-conditioned Hessian.

Table 1: Least square problems from Moré, Garbow, Hillstrom, 1981

		MINOS	L-BFGS-B	BCGD-GS-q-acc
$f(x)$	c	♯nz/objec/cpu	♯nz/objec/cpu	♯nz/objec/cpu
BAL	1	1000/1000/43.9	1000/1000/.02	1000/1000/.1
	10	1000/9999.9/43.9	1000/9999.9/.03	1000/9999.9/.2
	100	1000/99997.5/44.3	1000/99997.5/.1	1000/99997.5/.1
BT	.1	1000/71.725/100.6	1000/84.00/.02	1000/71.74/.9
	1	997/672.41/94.7	981/668.72/.2	1000/626.67/42.4
	10	0/1000/56.0	0/1000/.01	0/1000/.01
DBV	.1	0/0/51.5	999/83.45/.01	0/0/.5
	1	0/0/50.8	0/0/.01	2/0/.3
	10	0/0/52.5	0/0/.00	0/0/.01
EPS	1	1000/351.14/60.3	999/352.52/.05	1000/351.14/.3
	10	243/1250/44.2	250/1250/.01	249/1250/.1
	100	0/1250/51.5	0/1250/.01	0/1250/.01
ER	1	1000/436.25/71.5	1000/436.25/.1	1000/436.25/.1
	10	0/500/50.2	500/1721.1/.00	0/500/.3
	100	0/500/52.4	0/500/.00	0/500/.03
LFR	.1	1000/98.5/77.2	1000/98.5/.00	1000/98.5/.03
	1	1000/751/73.8	0/751/.01	0/751/.01
	10	0/1001/53.3	0/1001/.01	0/1001/.01
VD	1	1000/937.59/43.0	1000/1000.0/.00	1000/937.66/.5
	10	413/6726.80/56.9	974/5.8$\cdot 10^{12}$/2.3	1000/6726.81/60.3
	100	136/55043/57.4	996/75135/.2	1000/55043/88.1

Table 2: Performance of MINOS, LBFGS-B and BCGD, with $n = N$, $x^{\text{init}} = (1, \ldots, 1)$

BCGD was recently applied to Group Lasso for logistic regression (Meier et al '06)

$$\min_{x=(x_1,\ldots,x_n)} -\ell(x) + c\sum_{i=1}^{n} \omega_i \|x_i\|_2$$

$c > 0$, $\omega_i > 0$.

$\ell : \Re^N \to \Re$ is the log-likelihood for linear logistic regression.

Extension to constraints?

Linearly Constrained Problem

P2

$$\boxed{\min_{x \in X} f(x)}$$

$f : \Re^n \to \Re$ is cont. diff.

$X = \{x = (x_1, ..., x_n) \in \Re^n \mid l \le x \le u,\ Ax = b\}$, $A \in \Re^{m \times n}$, $b \in \Re^m$, $l \le u$.

Special case. Support Vector Machine QP (Vapnik '82)

$$\min_{0 \le x \le Ce,\ a^T x = 0} \frac{1}{2} x^T Q x - e^T x$$

$C > 0$, $a \in \Re^n$, $e^T = (1, ..., 1)$, $Q_{ij} = a_i a_j K(z_i, z_j)$, and $K : \Re^p \times \Re^p \to \Re$.

$K(z_i, z_j) = z_i^T z_j$ Linear

$K(z_i, z_j) = \exp(-\gamma \|z_i - z_j\|^2)$ ($\gamma > 0$) Gaussian

Block Coord. Gradient Descent Method for *P2*

For $x \in X$, $\mathcal{I} \subseteq \{1, ..., n\}$, and $H \succ 0_n$, let $d_H(x; \mathcal{I})$ and $q_H(x; \mathcal{I})$ be the optimal soln and obj. value of

$$\boxed{\min_{d | x+d \in X,\ d_i = 0\ \forall i \notin \mathcal{I}} \{\nabla f(x)^T d + \frac{1}{2} d^T H d\}}$$

direc. subprob

BCGD for P2:

Given $x \in X$, choose $\mathcal{I} \subseteq \{1, ..., n\}$, $H \succ 0_n$. Let $d = d_H(x; \mathcal{I})$.

Update $$\boxed{x^{\text{new}} = x + \alpha d \qquad (\alpha > 0)}$$

until “convergence.”

- d is easily calculated when $m = 1$ and $|\mathcal{I}| = 2$.

Gauss-Southwell-q: Choose $\mathcal{I}$ with

$$q_D(x;\mathcal{I}) \le \upsilon \, q_D(x;\{1,...,n\})$$

($0 < \upsilon \le 1$, $D \succ 0_n$ is diagonal, e.g., $D = I$ or $D = \mathrm{diag}(H)$).

For $m = 1$, such $\mathcal{I}$ with $|\mathcal{I}| = 2$ can be found in $O(n)$ ops by solving a continuous quadratic knapsack problem and finding a “conformal realization” of the solution.

Inexact Armijo LS: α = largest element of $\{s, s\beta, s\beta^2, ...\}$ satisfying

$$x + \alpha d \in X, \qquad f(x + \alpha d) - f(x) \le \sigma \alpha q_H(x;\mathcal{I})$$

($s > 0$, $0 < \beta < 1$, $0 < \sigma < 1$).

Convergence Results: (a) If $0 < \underline{\lambda} \leq \lambda_i(D), \lambda_i(H) \leq \bar{\lambda} \ \forall i$, then every cluster point of the x-sequence generated by BCGD method is a stationary point of P2.

(b) If in addition f satisfies any of the following assumptions and $\mathcal{I}$ is chosen by G-Southwell-q, then the x-sequence converges at R-linear rate.

C1 f is strongly convex, ∇f is Lipschitz cont. on X.

C2 f is (nonconvex) quadratic.

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 $\Re^m$.

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 $\Re^m$.

• For SVM QP, BCGD has R-linear convergence (with no additional assumption). Similar work as decomposition methods (Joachims '98, Platt '99, Lin et al, ...)

Numerical Tests:

- Implement BCGD in Fortran for SVM QP (two-class data classification).
- $x^{\text{init}} = 0$. Cache most recently used columns of Q.
- On large benchmark problems

`a7a`	$(p = 122, n = 16100)$,
`a8a`	$(p = 123, n = 22696)$,
`a9a`	$(p = 123, n = 32561)$,
`ijcnn1`	$(p = 22, n = 49990)$,
`w7a`	$(p = 300, n = 24692)$

and using nonlinear kernel, BCGD is comparable in CPU time and solution quality with the C++ SVM code LIBSVM (Lin et al). Using linear kernel, BCGD is much slower (it doesn't yet do variable fixing as in LIBSVM).

Conclusions & Future Work

1. For ML estimation, ℓ_1-regularization induces sparsity in the solution and avoids oversmoothing the signals.

2. The resulting estimation problem can be solved effectively by BCM or BCGD, exploiting the problem structure, including nondiffer. of ℓ_1-norm. Which to use? Depends on problem.

3. Applications to denoising, regression, SVM..

4. Improve BCGD speed for SVM QP using linear kernel? Efficient implementation for $m = 2$ constraints?