A Coordinate Gradient Descent Method for Structured Nonsmooth Optimization

Sangwoon Yun
Mathematics, National University of Singapore

May 21, 2008

Outline

I. Nonsmooth Separable Minimization

II. Linearly Constrained Smooth Minimization

III. Extensions

I. Nonsmooth Separable Minimization

Outline

- Bound-constrained Optimization $\&$ ℓ_1-regularized Convex Minimization
- General Problem Model: Nonsmooth Separable Minimization
- Coordinate Gradient Descent Method
- Convergence Results
- Numerical Experience on ℓ_1-regularized Convex Minimization

Bound-constrained Optimization & ℓ_1-regularized Convex Minimization

Bound-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).

Can be reformulated as the following unconstrained optimization problem:

$$\min_{x} \quad f(x) + P(x),$$

where $P(x) = \begin{cases} 0 & \text{if } l \le x \le u \\ \infty & \text{else} \end{cases}$.

ℓ_1- regularized convex minimization problem

1. ℓ_1-regularized linear least squares problem

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

Formulate this as an unconstrained convex optimization problem:

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

2. ℓ_1-regularized logistic regression problem

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

where $a_i = b_i z_i$ and $(z_i, b_i) \in \Re^{n-1} \times \{-1, 1\},\ i = 1, ..., m$ are a given set of (observed or training) examples.

General Problem Model: Nonsmooth Separable Optimization

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

$$\min_d \ \nabla f(x)^T d + \tfrac{1}{2}\rho\|d\|_2^2 + cP(x+d) \qquad (\rho > 0)$$

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

- Auslender (78)
- Mine and Fukushima (81)
- Kiwiel (86)
- Fukushima (91)

The above studies did not present numerical results.

Coord. Gradient Descent Method

Descent direction.

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

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

direc. subprob

Let $d_H(x; \mathcal{J})$ and $q_H(x; \mathcal{J})$ be the opt. soln and obj. value of the direc. subprob.

Facts:

- $d_H(x; \mathcal{N}) = 0 \ \Leftrightarrow F_c'(x; d) \geq 0 \ \forall d \in \Re^n$. 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 $d = d_H(x; \mathcal{J})$.

This coord. grad. descent approach may be viewed as a hybrid of gradient-projection and coordinate descent. In particular,

- if $\mathcal{J} = \mathcal{N}$ and $P(x) = \begin{cases} 0 & \text{if } l \le x \le u \\ \infty & \text{else} \end{cases}$, then $d_H(x; \mathcal{N})$ is a scaled gradient-projection direction for bound-constrained optimization.

- if f is quadratic and we choose $H = \nabla^2 f(x)$, then $d_H(x; \mathcal{J})$ is a (block) coordinate descent direction.

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

- If $P \equiv 0$, then $d_H(x)_j = -\nabla f(x)_j / H_{jj}$.

- If $P(x) = \begin{cases} 0 & \text{if } l \le x \le u \\ \infty & \text{else} \end{cases}$, then $d_H(x)_j = \text{median}\{l_j - x_j, -\nabla f(x)_j / H_{jj}, u_j - x_j\}$.

- If P is the 1-norm, then $d_H(x)_j = -\text{median}\{(\nabla f(x)_j - c)/H_{jj}, x_j, (\nabla f(x)_j + c)/H_{jj}\}$.

Stepsize: **Armijo rule**

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

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

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

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

or the limited minimization rule

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

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

Choose $\mathcal{J}$:

- Gauss-Seidel rule:

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

- Gauss-Southwell-r rule:

$$\|d_D(x;\mathcal{J})\|_\infty \geq \upsilon \|d_D(x;\mathcal{N})\|_\infty$$

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

- Gauss-Southwell-q rule:

$$q_D(x;\mathcal{J}) \leq \upsilon \; q_D(x;\mathcal{N}),$$

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

Coordinate Descent Method

When $P \equiv 0$. Given $x \in \Re^n$, Choose $i \in \mathcal{N}$. Update

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

Repeat until convergence.

- Gauss-Seidel rule: Choose i cyclically, $1,\ 2, ...,\ n,\ 1,\ 2, ...$
- Gauss-Southwell rule: Choose i with $|\frac{\partial f}{\partial x_i}(x)|$ maximum.

Properties:

- If f convex, then every cluster point of the x-sequence is a minimizer.
- If f nonconvex, then G-Seidel can cycle (Powell '73) but G-Southwell still converges.
- Convergence is possible when $P \not\equiv 0$ (Tseng '01).

Advantage of CGD

- 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

- $0 \prec \underline{\lambda} I \preceq D,\ H \preceq \bar{\lambda} I$,
- $\mathcal{J}$ is chosen by G-Seidel, G-Southwell-r, G-Southwell-q rule,
- α is chosen by Armijo rule,

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

Local convergence rate If

- $0 \prec \underline{\lambda} I \preceq D,\ H \preceq \bar{\lambda} I,$
- $\mathcal{J}$ is chosen by G-Seidel or Gauss-Southwell-q rule,
- α is chosen by Armijo rule,

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 Lipschitzian error bound

- Error Bound

$$\text{dist}(x, X^*) \leq \kappa \|d_I(x; \mathcal{N})\|_2 \quad \text{whenever } \|d_I(x; \mathcal{N})\|_2 \leq \epsilon,$$

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

Numerical Experience on ℓ_1-regularized Convex Minimization

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

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

- 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,

$$\mathcal{J} = \{j \mid \|d_H(x;j)\|_2 \geq \upsilon \|d_H(x;i)\|_\infty\}.$$

Or by Gauss-Southwell-q rule,

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

- Choose α by the minimization rule.

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

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

- The CGD method is terminated when

$$\|Hd_H(x;\mathcal{N})\|_\infty \leq 10^{-3}.$$

- $A \in \Re^{m \times n}$ is obtained by first filling it with independent samples of the standard Gaussian distribution and then orthonormalizing the rows ($m = 2048, n = 8192$).

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

the measurement Ax is corrupted by the noise ξ, $b = Ax + \xi$, where ξ is Gaussian white noise with variance $(0.01\|Ax\|)^2$.

- Comparison with l1-ls (Kim et al '07, interior-point), GPSR-BB (Figueiredo, Nowak and Wright '07, gradient projection), and FPC (Hale, Yin and Zhang '07, fixed-point continuation).

- Require only matrix-vector mults. involving A and A^T (dominant computations) at each iteration.

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

But matrix-vector mults. involving A cost $O(m|\mathcal{J}|)$ opers. (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 run l1-ls and then each of the others until each reaches the same objective value reached by l1-ls (10 random instances).

	l1-ls	CGD-GS-q	CGD-GS-r	GPSR-BB	FPC
	$n = 8192, m = 2048, c = 0.05\|\|A^T b\|\|_\infty$				
mean iterations	12	17	20	28	83
mean nnz(x)	846	402	442	534	574
mean CPU time	1.9e+01	8.3e-01	1.0e+00	2.7e+00	7.6e+00
mean error	1.4e-01	1.5e-01	1.5e-01	1.5e-01	1.5e-01
	$n = 8192, m = 2048, c = 0.01\|\|A^T b\|\|_\infty$				
mean iterations	13	36	44	104	120
mean nnz(x)	1043	779	1110	968	954
mean CPU time	2.6e+01	1.7e+00	2.0e+00	9.0e+00	1.0e+01
mean error	3.4e-02	4.6e-02	4.6e-02	4.2e-02	3.6e-02
	$n = 8192, m = 2048, c = 0.005\|\|A^T b\|\|_\infty$				
mean iterations	13	63	73	1001	118
mean nnz(x)	1234	1383	1947	8178	1567
mean CPU time	2.9e+01	3.0e+00	3.5e+00	8.7e+01	1.0e+01
mean error	2.2e-02	3.4e-02	3.3e-02	7.1e-01	3.0e-02

2. ℓ_1-regularized logistic regression problem:

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

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

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

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

- Comparison with l1-logreg (Koh, Kim and Boyd '07, interior-point) and SpaRSA (Wright, Nowak and Figueiredo '07, iterative method).

Test Results

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

	l1-logreg (10^{-4})	CGD-GS-q	CGD-GS-r	SpaRSA
leu	$n = 7130, m = 38, \mu = 0.01\mu_{\max}$			
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	$n = 47237, m = 20242, \mu = 0.01\mu_{\max}$			
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	$n = 20959, m = 72309, \mu = 0.01\mu_{\max}$			
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

- 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	$n = 10001, m = 1000, c = 0.01 c_{\max}$		
mean iterations	20	230	260
mean CPU time	2.4e+02	1.2e+01	1.4e+01
10 random	$n = 1001, m = 100, c = 0.01 c_{\max}$		
mean iterations	17	187	220
mean CPU time	1.9e-01	2.4e-01	2.9e-01
10 random	$n = 1001, m = 10000, c = 0.01 c_{\max}$		
mean iterations	16	82	88
mean CPU time	2.3e+02	4.8e+00	5.2e+00
10 random	$n = 101, m = 1000, c = 0.01 c_{\max}$		
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 $O(\min(n-1, m)^2 \max(n-1, m))$ opers. per iteration. In contrast, the computational cost of CGD is $O(mn)$ opers. per iteration.

II. Linearly Constrained Smooth Minimization

Outline

- Support Vector Machine (Primal and Dual Optimization Problem)
- General Problem Model: Linearly Constrained Smooth Minimization
- Coordinate Gradient Descent Method
- Convergence Results
- Complexity Bound
- Index Subset Selection
- Numerical Experience on SVM QP

Support Vector Machines

Support Vector Classification

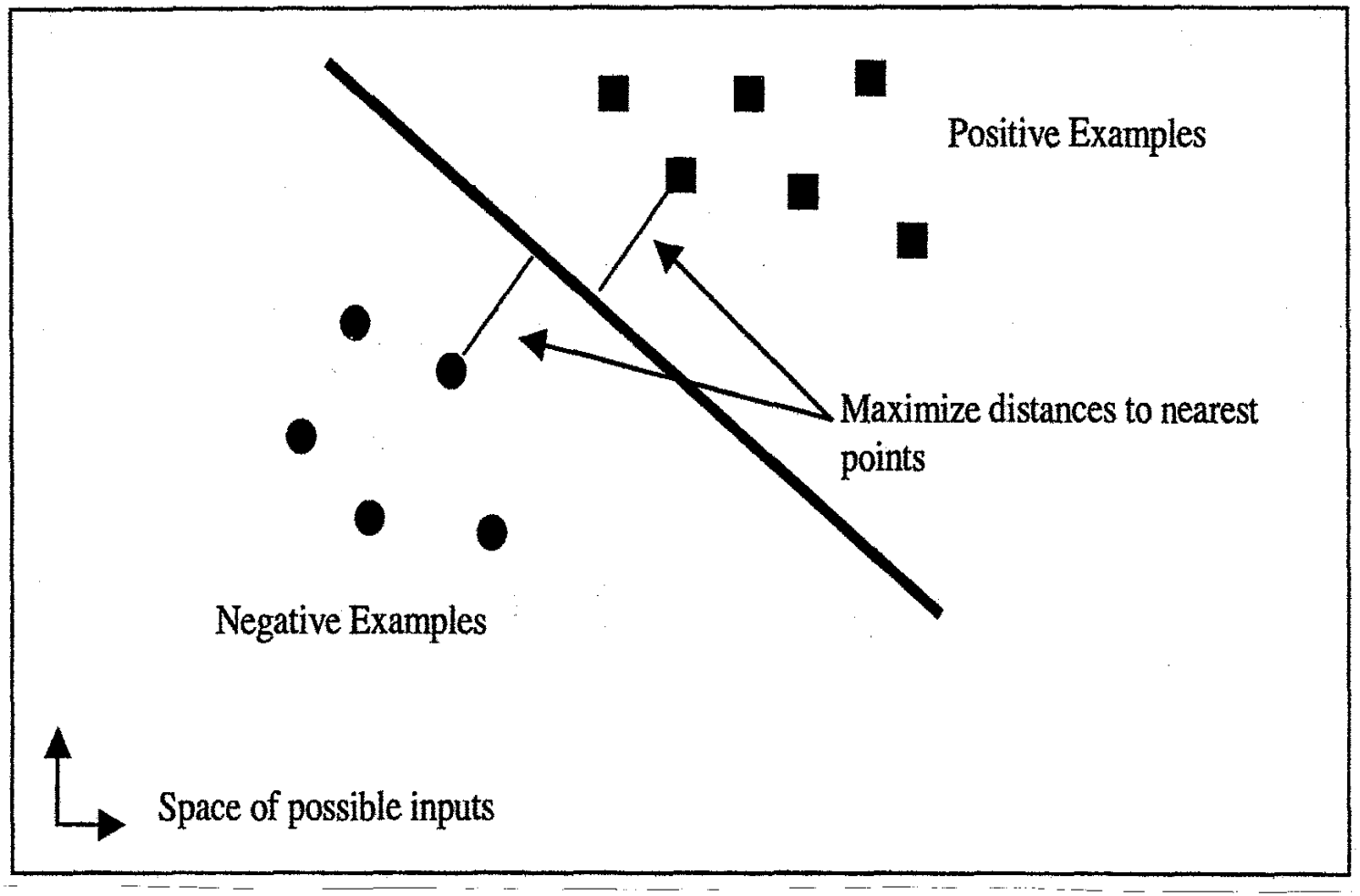
- Training points : $z_i \in \Re^p$, $i = 1, ..., n$.
- Consider a simple case with two classes (linear separable case):

  Define a vector a :

$$a_i = \begin{cases} 1 & \text{if } z_i \text{ in class } 1 \\ -1 & \text{if } z_i \text{ in class } 2 \end{cases}$$

- A hyperplane ($0 = w^T z - b$) separates data with the maximal margin.

Margin is the distance of the hyperplane to the nearest of the positive and negative points.
Nearest points lie on the planes $\pm 1 = w^T z - b$


Positive Examples
Maximize distances to nearest points
Negative Examples
Space of possible inputs

SVM Optimization Problem

- The (original) Optimization Problem

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{s.t.} \quad & a_i \left(w^T z_i - b\right) \geq 1, \;\; i = 1, ..., n. \end{aligned}$$

- The Modified Optimization Problem (allows, but penalizes, the failure of a point to reach the correct margin, by Cortes and Vapnik, 1995)

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & a_i \left(w^T z_i - b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \; i = 1, ..., n. \end{aligned}$$

SVM (Dual) Optimization Problem (Convex Quadratic Program)

$$\begin{array}{rl} \min_x & \frac{1}{2}x^T Q x - e^T x \\ \text{subject to} & 0 \le x_i \le C, \quad i = 1, ..., n, \\ & a^T x = 0, \end{array}$$

where $a \in \{-1, 1\}^n$, $0 < C \le \infty$, $e = [1, ..., 1]^T$, $Q \in \Re^{n \times n}$ is a sym. pos. semidef. with $Q_{ij} = a_i a_j K(z_i, z_j)$, $K : \Re^p \times \Re^p \to \Re$ ("kernel function"), and $z_i \in \Re^p$ ("ith data point"), $i = 1, ..., n$.

Popular Choices of K:

- linear kernel $K(z_i, z_j) = z_i^T z_j$
- radial basis function kernel $K(z_i, z_j) = \exp(-\gamma \|z_i - z_j\|_2^2)$
- sigmoid kernel $K(z_i, z_j) = \tanh(\gamma z_i^T z_j)$

where γ is a constant.

Q is an $n \times n$ fully dense matrix and even indefinite. ($n \geq 5000$)

Interior-point methods cannot be directly applied, except in the case of linear kernel.

Previous methods

Decomposition methods based on iterative block-coordinate descent have become popular for solving SVM QP.

- Joachims (98)
- Platt (99)
- Chang et al. (00)
- Keerthi et al. (00)
- Hush and Scovel (03)
- Palagi and Sciandrone (05)
- Fan et al. (05)

Decomposition methods use search directions of small support (i.e., few nonzeros) and achieve linear convergence under additional assumptions such as Q being positive definite.

General Problem Model: Linearly Constrained Smooth Minimization

$$\begin{array}{ll} \min\limits_{x \in \Re^n} & f(x) \\ \text{s.t.} & x \in X := \{x \mid l \leq x \leq u,\ Ax = b\}, \end{array}$$

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

$A \in \Re^{m \times n}$, $b \in \Re^m$, and $l \leq u$ (possibly with $-\infty$ or ∞ components).

- For SVM QP, f is quadratic (possibly nonconvex) and $m = 1$.

Coord. Gradient Descent Method

Descent Direction.

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

$$\min_{x+d \in X,\ d_j = 0\ \forall j \notin \mathcal{J}} \{\nabla f(x)^T d + \frac{1}{2} d^T H d\}.$$

direc. subprob

Let $d_H(x; \mathcal{J})$ and $q_H(x; \mathcal{J})$ be the opt. soln and obj. value of the direc. subprob.

Facts:

- $d_H(x; \mathcal{N}) = 0 \Leftrightarrow x \in X$ is a stationary point of f over X. stationarity
- $d_H(x; \mathcal{J}) = 0 \Leftrightarrow q_H(x; \mathcal{J}) = 0$.

Stepsize: **Armijo rule**

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

$$f(x+\alpha d) - f(x) \leq \sigma \alpha q_H(x;\mathcal{J}) \quad (0<\beta<1,\, 0<\sigma<1).$$

For a QP, the minimization rule or the limited minimization rule can also be used.

Choose $\mathcal{J}$ (Gauss-Southwell-q rule): $\mathcal{J}$ satisfies

$$q_D(x;\mathcal{J}) \leq \upsilon\; q_D(x;\mathcal{N}),$$

Where $D \succ 0_n$ is diagonal, $0 < \upsilon \leq 1$.

Convergence Results

Global Convergence If

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

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

Local Convergence Rate If

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

in addition, if f satisfies 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 X

C2 f is (nonconvex) quadratic. (e.g., SVM QP)

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 $\Re^m$.

Notes:

Proof of convergence rate uses a local Lipschitzian error bound

- Error Bound

$$\text{dist}(x, X^*) \leq \kappa \|d_I(x; \mathcal{N})\|_2 \quad \text{whenever } \|d_I(x; \mathcal{N})\|_2 \leq \epsilon,$$

for some $\kappa > 0$, $\epsilon > 0$, where X^* denotes the set of stationary points of f over X and $\text{dist}(x, X^*) = \min_{x^* \in X^*} \|x - x^*\|_2$.

Complexity Bound

If $0 \prec \underline{\lambda} I \preceq D,\ H \preceq \bar{\lambda} I$ and f is convex with Lipschitz cont. grad., then the number of iterations for achieving ϵ-optimality is

$$O\left(\frac{Lr^0}{\upsilon\epsilon} + \max\left\{0, \frac{L}{\upsilon}\ln\left(\frac{e^0}{r^0}\right)\right\}\right),$$

where L is a Lipschitz constant, $e^0 = f(x^0) - \min_{x\in X} f(x)$, and $r^0 = \max_{x\in X}\left\{\mathrm{dist}(x, X^*)^2 \mid f(x) \le f(x^0)\right\}$.

The constant in $O(\cdot)$ depends on $\underline{\lambda}$, $\bar{\lambda}$, σ, β.

When specialized to SVM QP, our complexity bound for achieving ϵ-optimality compares favorably with existing bounds (Hush and Scovel '03, List and Simon '05).

Index Subset Selection

Elementary Vector (Rockafellar, 1969)

- For any $d \in \Re^n$, the support of d is $\mathrm{supp}(d) := \{j \in \mathcal{N} \mid d_j \neq 0\}$.
- A d' is *conformal* to d if $\mathrm{supp}(d') \subseteq \mathrm{supp}(d)$ and $d'_j d_j \geq 0 \; \forall j \in \mathcal{N}$.
- A nonzero d is an *elementary vector* of $\mathrm{Null}(A)$ if $d \in \mathrm{Null}(A)$ and there is no nonzero $d' \in \mathrm{Null}(A)$ that is conformal to d and $\mathrm{supp}(d') \neq \mathrm{supp}(d)$.
- Each elementary vector d satisfies $|\mathrm{supp}(d)| \leq \mathrm{rank}(A) + 1$.

Find $\mathcal{J}$ with $|\mathcal{J}| = 2$ in $O(n)$ opers. (SVM QP, $m = 1$)

- Step 1: Find $d_D(x; \mathcal{N})$ in $O(n)$ opers. by solving a cont. quad. knapsack problem:

$$\begin{array}{ll} \min_d & \frac{1}{2} d^T D d + g^T d \\ \text{s.t.} & l \leq x + d \leq u, \\ & Ad = 0, \end{array}$$

Where $D \succ 0_n$ is diagonal, and $g = \nabla f(x)$.

- Step 2: Find a *conformal realization* of $d_D(x; \mathcal{N})$:

$d_D(x; \mathcal{N}) = \sum_{i=1}^{r} d^i$ where d^i is an elementary vector of $\mathrm{Null}(A)$

and $r \leq n - 1$.

Choose $\mathcal{J} = \mathrm{supp}(d^{\bar{i}})$ where $\bar{i} \in \underset{i \in \{1,\ldots,r\}}{\arg\min}\ g^T d^i + \frac{1}{2}(d^i)^T D d^i$.

This finds a $\mathcal{J}$ satisfying $|\mathcal{J}| = 2$ and $q_D(x; \mathcal{J}) \leq \frac{1}{n-1} q_D(x; \mathcal{N})$ in $O(n)$ opers.

Numerical Experience on SVM QP

- Implement CGD method in Fortran.
- Choose $\mathcal{J}$ by Gauss-Southwell-q rule with

$$D = \operatorname{diag}\left[\max\{Q_{jj}, 10^{-5}\}\right]_{j=1,\ldots,n},$$

as described in previous slide.

- Our implementation of the CGD method has the form

$$x^{\text{new}} = x + d_Q(x; \mathcal{J}),$$

with $|\mathcal{J}| = 2$. This corresponds to the CGD method with α chosen by the minimization rule. (The choice of H is actually immaterial here.)

- Compute $d_D(x, \mathcal{N})$ and $q_D(x; \mathcal{N})$ by using a linear-time Fortran code `k1vfo` provided by Krzysztof Kiwiel.

- $x^{\rm init} = 0$: $O(n)$ opers. to compute gradient $Qx^{\rm init} - e$.
  (for general $x^{\rm init}$, $O(n^2)$ opers.)

- $O(n)$ opers. per iteration to update gradient $Qx - e$.

- The CGD method is terminated when $-q_D(x; \mathcal{N}) \leq 10^{-5}$.

- Additional refinements such as caching most recently used columns of Q and using supports of 3 elementary vectors for a conformal realization of $d_D(x; \mathcal{N})$ are used to speed up the method.

- Numerical tests on some large two-class data classification problems.

- Comparison with LIBSVM (version 2.83), which chooses $\mathcal{J}$ differently, but with the same cardinality of 2.

Test Results ($\gamma = 1/p$:default values of LIBSVM)

Data	n/p	C/kernel	**LIBSVM**	**CGD-3pair**
			iter/obj/cpu	iter/obj/cpu
a7a	16100/122	1/lin	64108/-5699.253/1.3	56869/-5699.246/6.3
		10/lin	713288/-56875.57/4.6	322000/-56873.58/32.8
		1/rbf	4109/-5899.071/1.3	4481/-5899.070/1.0
		10/rbf	10385/-55195.29/1.4	16068/-55195.30/2.0
		1/sig	3941/-6095.529/1.7	4201/-6095.529/1.2
		10/sig	9942/-57878.56/1.7	10890/-57878.57/1.8
ijcnn1	49990/22	1/lin	16404/-8590.158/3.0	20297/-8590.155/6.5
		10/lin	155333/-85441.01/4.2	155274/-85441.00/46.9
		1/rbf	5713/-8148.187/4.6	6688/-8148.187/3.8
		10/rbf	6415/-61036.54/3.5	12180/-61036.54/4.8
		1/sig	6796/-9156.916/7.0	6856/-9156.916/5.0
		10/sig	10090/-88898.40/6.4	12420/-88898.39/6.5
w7a	24692/300	1/lin	66382/-765.4115/0.4	72444/-765.4116/8.2
		10/lin	662877/-7008.306/1.1	493842/-7008.307/60.6
		1/rbf	1550/-1372.011/0.4	1783/-1372.010/0.5
		10/rbf	4139/-10422.69/0.4	4491/-10422.70/0.8
		1/sig	1477/-1427.453/0.4	2020/-1427.455/0.4
		10/sig	2853/-11668.85/0.3	5520/-11668.86/0.9

- CGD-3pair is slower than LIBSVM when the linear kernel is used, due to the greater times spent in finding $d_D(x; \mathcal{N})$ and for updating the gradient.
- CGD-3pair is comparable to LIBSVM in speed and solution quality for nonlinear kernel.

III. Extensions

Outline

- Linearly Constrained Nonsmooth Separable Minimization

- Sparse Covariance Selection

Linearly Constrained Nonsmooth Separable Minimization

- The CGD method ($\mathcal{J}$ is chosen by Gauss-Southwell-q rule) can be extended to solve a linearly constrained nonsmooth separable minimization:

$$\begin{array}{ll} \min\limits_{x\in\Re^n} & f(x)+cP(x) \\ \text{s.t.} & Ax=b. \end{array}$$

P is not necessarily separable for the convergence (and the complexity bound).

Sparse Covariance Selection

d'Aspremont, Banerjee, El Ghaoui '07
Zhaosong '07
Friedman, Hastie, Tibshirani '07

$$\min_{X \in \mathcal{S}_+^n} f(X) + c\|X\|_1$$

$f(X) = -\log\det(X) + \mathrm{tr}(XS)$ ($S \in \mathcal{S}_+^n$ is empirical covariance matrix), $\|X\|_1 = \sum_{ij} |X_{ij}|$, $c > 0$.

- f is strictly convex, cont. diff. on its domain, $O(n^3)$ opers. to evaluate. $\|\cdot\|_1$ is convex, nonsmooth. In applications, n can exceed 6000.

The dual problem is a bound-constrained convex program:

$$\min_{W \in \mathcal{S}_+^n, \|W - S\|_\infty \le c} -\log\det(W) - n$$

$\|Y\|_\infty = \max_{ij} |Y_{ij}|$.

- IP method requires $O(n^7 \log(1/\epsilon))$ opers. to find ϵ-optimal soln. Impractical!

Nesterov's first-order smoothing method requires $O(n^4/\epsilon)$ opers. Zhaosong '07.

- Use CD (GSeidel) to solve the dual problem, cycling thru columns $(i = 1, ..., n)$ of W. Each iteration reduces (via determinant property) to

$$\begin{array}{ll} \min\limits_{y \in \Re^{n-1}} & y^T (W_{i^-i^-})^{-1} y \\ \text{s.t.} & |y - S_{ij}| \leq c \quad i = 1, ..., n-1. \end{array}$$

or (via duality)

$$\min_{\zeta \in \Re^{n-1}} \frac{1}{2} \zeta^T W_{i^-i^-} \zeta - S_{i^-i}^T \zeta + c \|\zeta\|_1.$$

Solve this using IP method Banerjee et al '07 or CD (GSeidel) Friedman et al '07.

- Can apply CGD (GSeidel) to either primal or dual problem. When applied to the primal, each iteration entails

$$\min_{u \in \Re_n} \left\{ \mathrm{tr}((-X^{-1} + S)D) + \frac{1}{2} u^T H u + c\|X + D\|_1 \right\}_{D = u e_i^T + e_i u^T}.$$

For diagonal $H \succ 0_n$, the minimizing D has closed form. For each trial α in the Armijo LS, $\det(X + \alpha D)$ can be evaluated from $\det(X)$ and X^{-1} in $O(n^2)$ opers. Update X^{-1} in $O(n^2)$ opers.

Similar application to the dual.

Global convergence, local convergence rate, complexity analysis.

Numerical tests (ongoing).

Conclusions

1. Numerical results shows the practical efficiency of the method for a large-scale ℓ_1-regularized convex minimization.

2. The CGD method is the first globally convergent block-coordinate update method for general linearly constrained smooth minimization.

3. For SVM QP, numerical results show that CGD method can be competitive with state-of-the-art SVM code on large data classification problems when a nonlinear kernel is used.

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

5. Can CGD method be extended to handle the following generalized problem:

$$\begin{array}{ll} \min\limits_{x \in \Re^n} & f(x) + cP(x) \\ \text{s.t.} & f_1(x) = 0, \ldots, f_m(x) = 0, \end{array}$$

where $f_1(x), \ldots, f_m(x)$ are twice continuously differentiable functions.

6. Can CGD method be extended to a nonconvex nonsmooth regularization problem (e.g. ℓ_p-regularization, $0 < p < 1$)?

Thank you!

Tseng, P. and Yun S., A coordinate gradient descent method for nonsmooth separable minimization.
Tseng, P. and Yun S., A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training.
Tseng, P. and Yun S., A block-coordinate gradient descent method for linearly constrained 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)

Bi-level Optimization

Problem Model

$$\min_{x \in S_f} P(x)$$

$P : \Re^n \rightarrow (-\infty, \infty]$ is a proper, convex, lsc function.

S_f denotes the set of stationary points of a smooth convex function f over $\mathrm{dom} P = \{x \mid P(x) < \infty\}$.

Sparse solution of an underdetermined system of linear equations

$$P(x) = \|x\|_1.$$

$$f(x) = \|Ax - b\|_2^2.$$

Algorithm (Regularization Strategy)

Choose $x^0 \in \mathrm{dom} P$, $c^0 > 0$, $\epsilon^0 > 0$. For $k = 1, 2, \ldots,$ generate x^k from x^{k-1} according to the iteration:

1. Choose $c^k > 0$ and $\epsilon^k > 0$.

2. Compute x^k as a point satisfying

- $\|d_{D^k}(x^k; \mathcal{N})\|_2 \leq \epsilon^k$,
- $\|D^k d_{D^k}(x^k; \mathcal{N})\|_2 \leq \epsilon^k$
- $-(D^k x^k + \nabla f(x^k))^T d_{D^k}(x^k; \mathcal{N}) \leq \epsilon^k$

by applying the CGD method to the problem:

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

with $c = c^k$ and an initial point $x = x^{k-1}$.

Convergence Results

Assume P is level-bounded and $\mathrm{dom} P \cap S_f \neq \emptyset$.

If we choose c^k and ϵ^k to tend to zero so that

$$\lim_{k \to \infty} \frac{\epsilon^k}{c^k} = 0,$$

then every cluster point of $\{x^k\}$ is an optimal solution of the bi-level problem.