A Coordinate Gradient Descent Method for Linearly Constrained Smooth Optimization and Support Vector Machines Training

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

UW optimization seminar
March 06, 2007
(Joint work with Paul Tseng)

Talk Outline

- Support Vector Machines
- General Problem Model
- Coordinate Gradient Descent Method
- Convergence Results
- Index Subset Selection
- Numerical Experiance on SVM QP
- Conclusions & Future Work

Support Vector Machines

Support Vector Classification

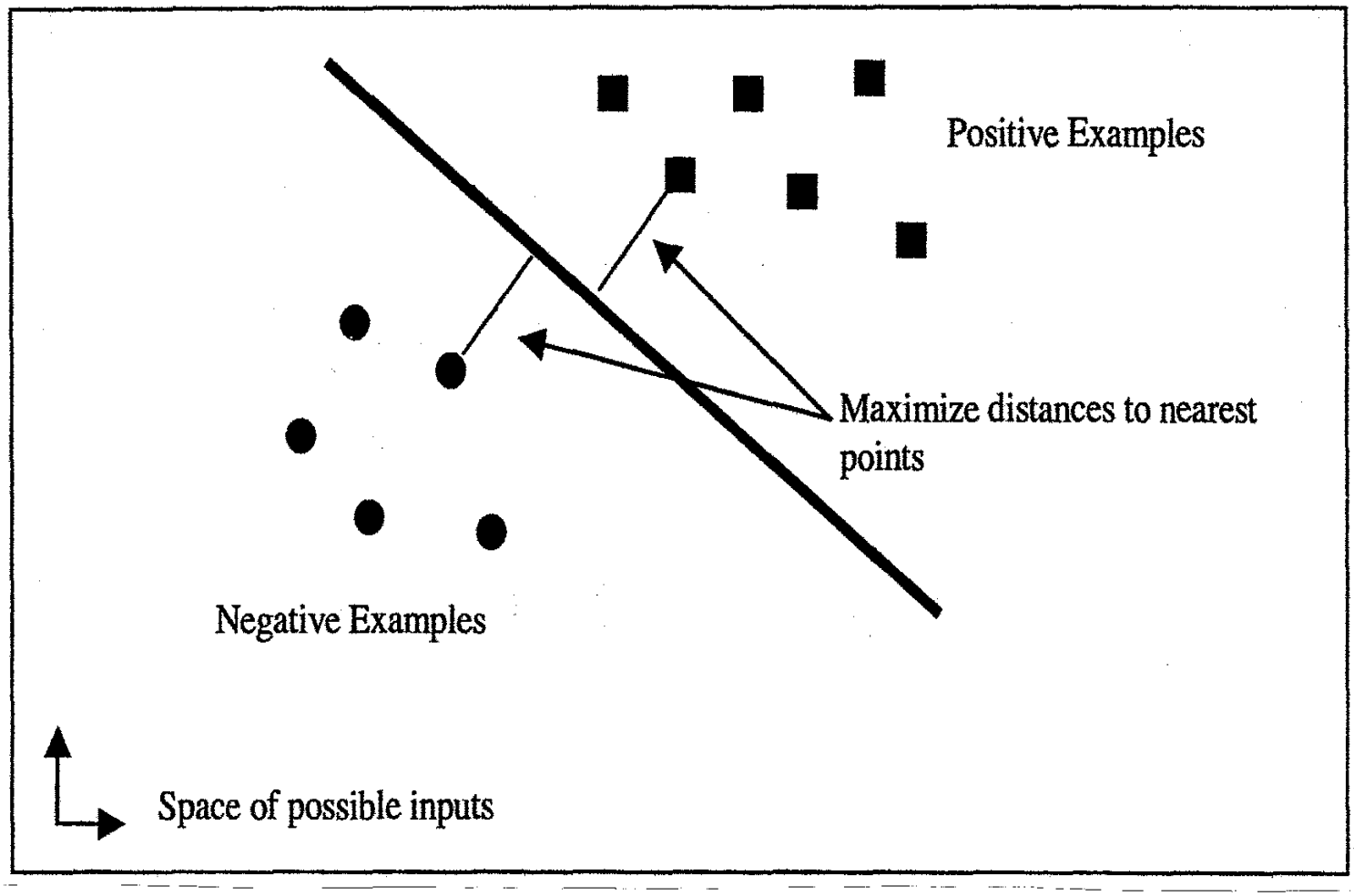
- Training points : $z_i \in \Re^p$, $i = 1, ..., n$.
- Consider a simple case with two classes (linear separable case):

  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

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

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

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

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 $K(z_i, z_j) = \exp(-\gamma \|z_i - z_j\|^2)$
- sigmoid kernel $K(z_i, z_j) = \tanh(\gamma z_i^T z_j + s)$

where γ and s are constants.

- 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.

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

General Problem Model

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

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

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

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

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

Facts:

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

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

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

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

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

- Error Bound

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

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

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 $|\mathrm{supp}(d)| \leq \mathrm{rank}(A) + 1$.

$\mathcal{J}$ when $m = 1$

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

- 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 $\bar{i} \in \arg\min_{i \in \{1,\ldots,r\}} g^T d^i + \frac{1}{2}(d^i)^T D d^i$.

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

Numerical Experiance on SVM QP

- Implement CGD method in Fortran.

- The index subset $\mathcal{J}$ by Gauss-Southwell-q rule with

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

- Our implementation of the CGD method has the form

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

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.

- $x^{\text{init}} = 0$: $O(n)$ opers. to compute gradient $Qx^{\text{init}} - e$.
  (for general x^{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 ($\gamma = 1/p$, $s = 0$: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 solving a cont. quad. knapsack problem and for updating the gradient.
- CGD-3pair is comparable to LIBSVM in speed and solution quality for nonlinear kernel.

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

where $z_i \in \Re^p$, γ and s are constants.

Conclusions & Future Work

1. On SVM QP, CGD method achieves linear convergence without assuming strict complementarity or Q is positive definite as in previous analyses of decomposition methods.

2. It is implementable in $O(n)$ opers. per iteration.

3. 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. For large-scale applications such as ν-SVM, $m = 2$. A conformal realization can be found in $O(n \log n)$ operations when $m = 2$. However, this can still be slow. Can this be improved to $O(n)$ operations?

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

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

where $c > 0$, $P : \Re^n \to (-\infty, \infty]$ is a separable proper convex lsc function.

Thank you!

Tseng, P. and Yun S., A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. (PDF file available at http://www.math.washington.edu/~sangwoon/)

Caching and Other Choices of $\mathcal{J}$

Using `k1vfo` and updating the gradient are the dominant computations.

- Cache the most recently used columns of Q, up to a user-specified limit `maxCN`, when updating the gradient $Qx - e$.

- There exists at least one elementary vector in this realization whose support $\mathcal{J}$ satisfies

$$q_D(x; \mathcal{J}) \leq \frac{1}{n-1} q_D(x; \mathcal{N}).$$

- From among all such $\mathcal{J}$, we find the best one (i.e., has the least $q_Q(x; \mathcal{J})$ value) and make this our choice for index subset.

- In addition, find from among all such $\mathcal{J}$ the second-best and third-best ones, if they exist. (In our tests, they always exist.)

- If the second-best one is disjoint from the best one, we make it the next index subset, and if the third-best one is disjoint from both the best and the second-best, we make it the second-next index subset.

- The procedure of selecting 3 (possible) pairs of the index subset is repeated at least once every 3 consecutive iterations.