Successive Overrelaxation for Support Vector Machines

Olvi L. Mangasarian & David R. Musicant

University of Wisconsin – Madison

www.cs.wisc.edu/˜olvi

ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-18.ps.Z

Outline

- The support vector machine and its variant
- SOR for support vector machines
- Numerical results
- Conclusion

The Discrimination Problem

- **Given m points in the n dimensional real space R^n**
- **Represented by an $m \times n$ matrix A**
- **Membership of each point A_i in the classes 1 or -1 is specified by:**
  - **An $m \times m$ diagonal matrix D with ± 1 along its diagonal**
- **Separate by two bounding planes: $x'w = \gamma \pm 1$ such that:**

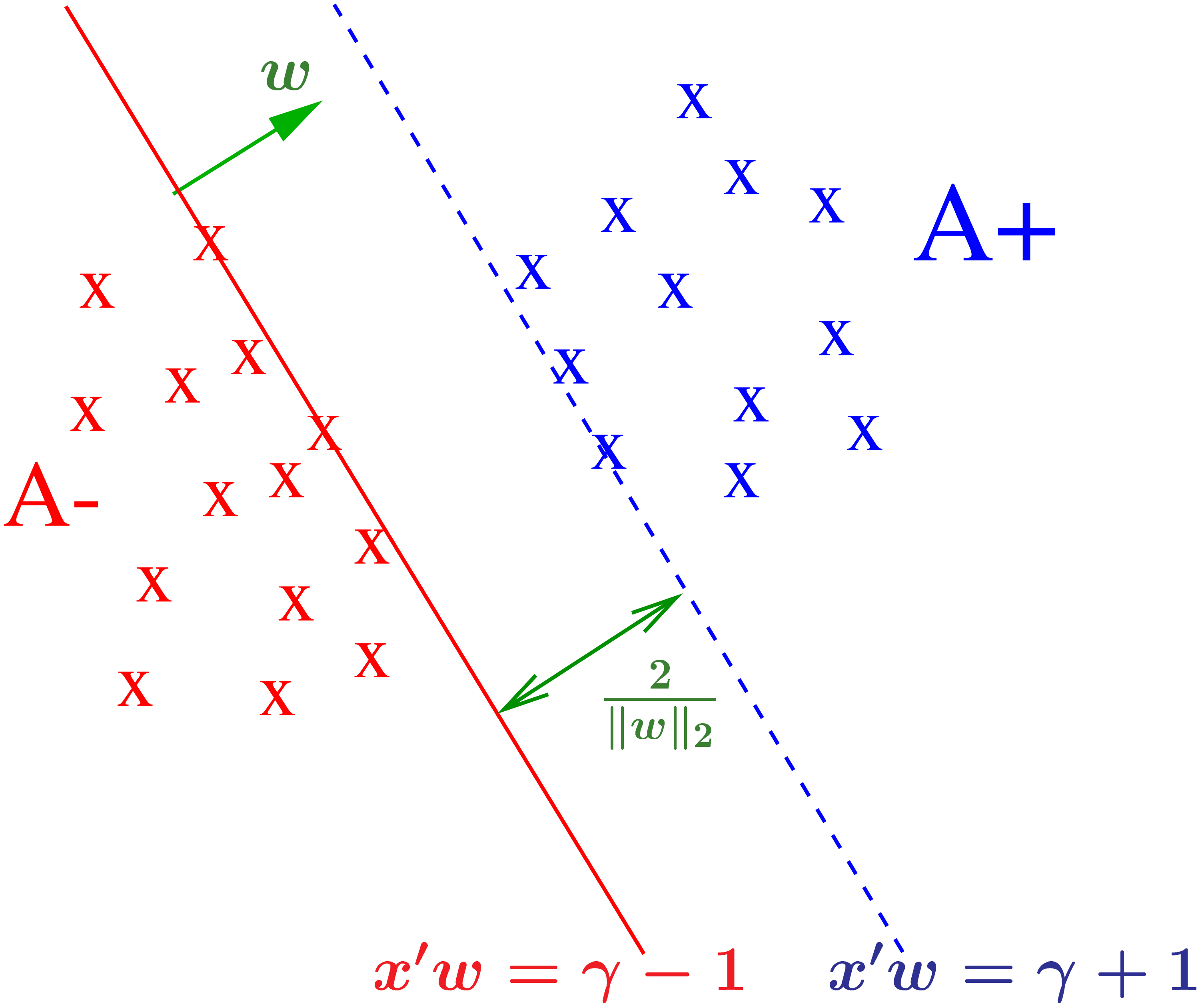
$$\begin{aligned} A_i w &\geq \gamma + 1, \quad \text{for } D_{ii} = +1, \\ A_i w &\leq \gamma - 1, \quad \text{for } D_{ii} = -1. \end{aligned}$$

More succinctly:

$$D(Aw - e\gamma) \geq e,$$

where e is a vector of ones.

Large Margin Classifier
Standard Formulation



(Linear) Support Vector Machine
Standard Formulation in w Space

$$
\begin{array}{rrcl}
\min\limits_{w,\gamma,y} & \nu e'y + \frac{1}{2}w'w & & \\
\text{s.t.} & D(Aw - e\gamma) + y & \geq & e \\
 & y & \geq & 0.
\end{array}
$$

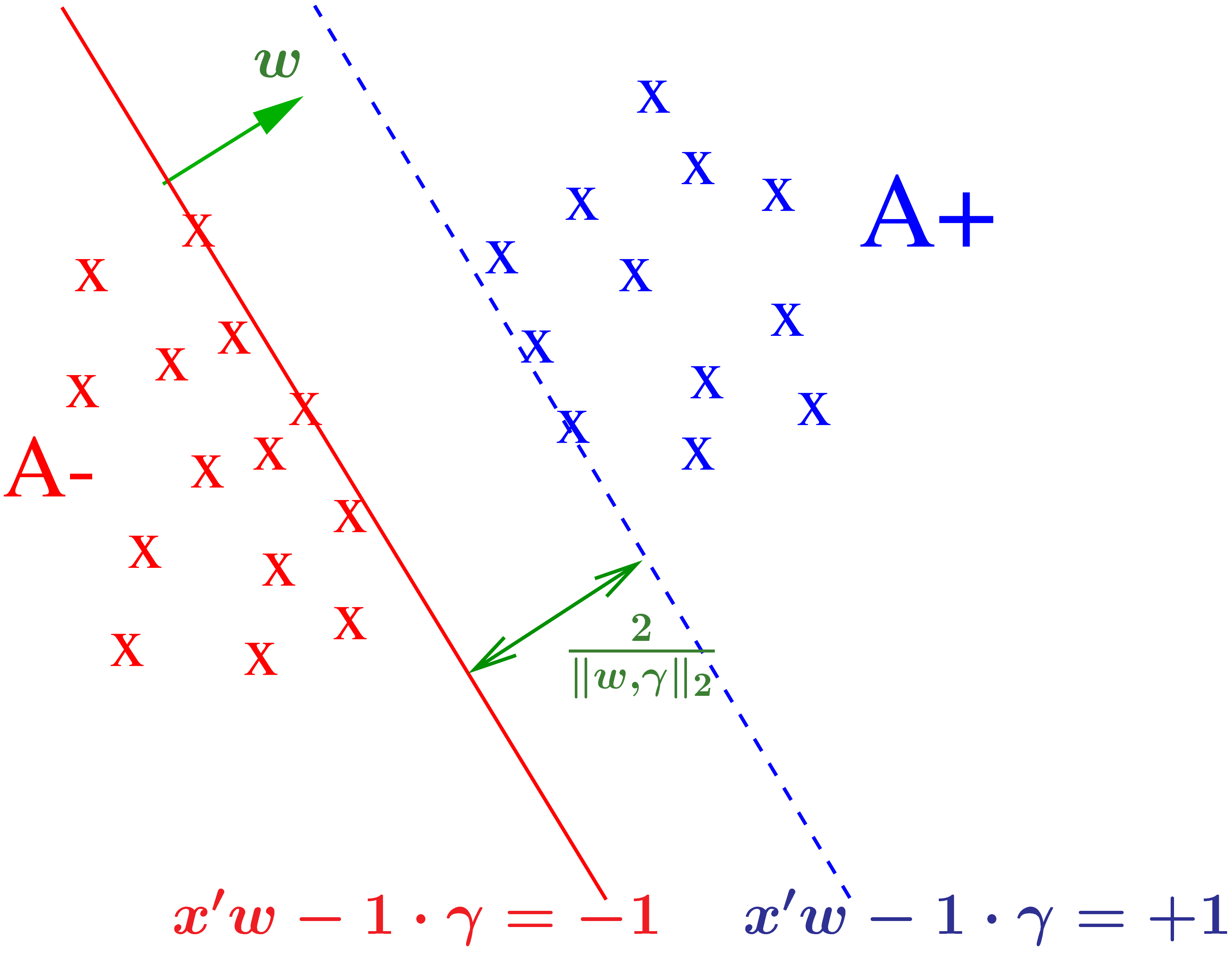
Here, $\nu > 0$ is a parameter, w is the normal to the bounding planes:

$$
\begin{array}{rcl}
x'w & = & \gamma + 1 \\
x'w & = & \gamma - 1.
\end{array}
$$

The quadratic term maximizes the distance between the bounding planes in the space R^n of w.

Large Margin Classifier

Formulation in (w, γ) Space



Variant of Support Vector Machine

$$\begin{array}{rrcl} \min\limits_{w,\gamma,y} & \nu e'y + \frac{1}{2}(w'w + \gamma^2) & & \\ \text{s.t.} & D(Aw - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array}$$

The quadratic term here maximizes the distance between the bounding planes in the space R^{n+1} of (w, γ):

$$\begin{array}{rcccl} x'w & - & \gamma & = & +1 \\ x'w & - & \gamma & = & -1. \end{array}$$

Wolfe Dual

For Support Vector Machine

For Variant of Support Vector Machine

$$\max_{u} \; -\frac{1}{2}u'DAA'Du + e'u, \;\; \text{s.t.} \;\; e'Du = 0, \;\; 0 \le u \le \nu e$$

$$\max_{u} \; -\frac{1}{2}u'DAA'Du - \frac{1}{2}u'Dee'Du + e'u, \;\; \text{s.t.} \;\; 0 \le u \le \nu e$$

Key difference: No equality constraint in dual of SVM variant!

Both formulations give same answers under appropriate conditions.

Wolfe Dual
For Support Vector Machine
For Variant of Support Vector Machine

$$\max_{u} \; -\frac{1}{2}u'DAA'Du + e'u, \text{ s.t. } e'Du = 0, \; 0 \leq u \leq \nu e$$

$$(w = A'Du, \;\; x'w = x'ADu = \gamma)$$

$$\max_{u} \; -\frac{1}{2}u'DAA'Du - \frac{1}{2}u'Dee'Du + e'u, \text{ s.t. } 0 \leq u \leq \nu e$$

$$(w = A'Du, \; \gamma = -e'Du, \;\; x'w = x'ADu = \gamma = -e'Du)$$

When do Both Formulations Give the Same Answer?

Proposition Each solution $(\tilde{w}, \tilde{\gamma}, \tilde{y})$ of of the variant SVM is a solution of the standard SVM, for a possibly larger value of ν, whenever the following linear system has a solution $\tilde{v}$:

$$A'Dv = 0,\ e'Dv = \tilde{\gamma},\ v \geq 0,$$

such that

$$e'\tilde{v}(e'\tilde{y} - 1) \leq \tilde{\gamma}^2.$$

For $1,000$ randomly generated problems with $A \in R^{40\times 5}$, the SVM and its variant with the same ν, only 34 cases had solution vectors w that differed by more than 0.001 in their 2-norm.

Nonlinear Support Vector Machine

Nonlinear Variant of Support Vector Machine

$$\max_{u} \; -\frac{1}{2}u'DK(A,A')Du + e'u, \text{ s.t. } e'Du = 0,\; 0 \le u \le \nu e$$

$$K(x',A')Du = \gamma \;\; \text{(Separating Surface)}$$

$$\max_{u} \; -\frac{1}{2}u'DK(A,A')Du - \frac{1}{2}u'Dee'Du + e'u, \text{ s.t. } 0 \le u \le \nu e$$

$$K(x',A')Du = -e'Du \;\; \text{(Separating Surface)}$$

Examples of Kernels

- Examples ($a \in R^m$, $\mu \in R$ and d integer):
  - Polynomial Kernel $(AA' + \mu aa')^d_{\bullet}$,
    where $[\cdot]^d_{\bullet}$ denotes component-wise exponentiation as in MATLAB.
  - Neural Network Kernel $(AA' + \mu aa')_{\bullet *}$,
    where $[\cdot]_{\bullet *}$ denotes the step function: $R \longrightarrow \{0, 1\}$ component-wise.
  - Radial Basis Kernel $\varepsilon^{-\mu\|A_i - A_j\|^2}$, $i, j = 1, \ldots, m$.

SOR for Support Vector Machines
Problem Setup

Define:

$$H = D[A \quad -e], \;\; L + E + L' = HH'$$

Then, dual of SVM variant becomes:

$$\min_{u} \frac{1}{2}\|H'u\|^2 - e'u, \text{ s.t. } u \in S = \{u \mid 0 \leq u \leq \nu e\}$$

Gradient projection necessary & sufficient optimality condition:

$$u = (u - \omega E^{-1}(HH'u - e))_{\#}, \; \omega > 0,$$

where $(\cdot)_{\#}$ denotes the 2-norm projection on the feasible region S.

Matrix Splitting Methods for Quadratic Programs
Subsume: SOR, Jacobi, SSOR

$$\min_u \frac{1}{2}\|H'u\|^2 - e'u, \text{ s.t. } u \in S = \{u \mid 0 \le u \le \nu e\}.$$

$$HH' = \omega^{-1}E(B+C), \text{ s.t. } B - C \text{ is positive definite.}$$

This leads to the following matrix splitting algorithm:

$$u^{i+1} = (u^{i+1} - Bu^{i+1} - Cu^i + \omega E^{-1}e)_\#.$$

SOR Splitting:

$$B = (I + \omega E^{-1}L), \; C = ((\omega - 1)I + \omega E^{-1}L'), \; 0 < \omega < 2,$$

for which:

$$B + C = \omega E^{-1}HH', \; B - C = (2-\omega)I + \omega E^{-1}(L - L').$$

SOR for Support Vector Machines
Algorithm & Convergence

SOR ALGORITHM: Choose $\omega \in (0,2)$. Start with any $u^0 \in R^m$. Having u^i compute u^{i+1} as follows:

$$\begin{aligned} u^{i+1} &= (u^i - \omega E^{-1}(HH'u^i - e + L(u^{i+1} - u^i)))_\# \\ &= (u^i - \omega E^{-1}(Lu^{i+1} + (E + L')u^i - e))_\#, \end{aligned}$$

until $\|u^{i+1} - u^i\|$ is less than some prescribed tolerance.

SOR LINEAR CONVERGENCE: The iterates $\{u^i\}$ of the SOR Algorithm converge R-linearly to a solution of $\bar{u}$ of the dual problem, and the objective function values $\{f(u^i)\}$ converge Q-linearly to $f(\bar{u})$. That is for $i \geq \bar{i}$ for some $\bar{i}$:

$$\begin{aligned} \|u^i - \bar{u}\| &\leq \mu\delta^i, \text{ for some } \mu > 0,\ \delta \in (0,1), \\ f(u^{i+1}) - f(\bar{u}) &\leq \tau(f(u^i) - f(\bar{u})), \text{ for some } \tau \in (0,1). \end{aligned}$$

Numerical Testing

SOR PERFORMANCE versus TRAINING SET SEPARABILITY

DATA SETS:

- **UCI Adult Dataset: 1605 to 32,561 points in R^{121}**
- **Simon Fraser University Synthetic Classification Data Set Generator (SCDS): 1 million points in R^{32}**
- **Prescribed Separability Random Set: 10 million points in R^{32}**

METHODS COMPARED TO:

- **Platt's Sequential Minimal Optimization (SMO) on Adult Dataset**
- **Bradley & OLM Linear Programming Chunking (LPC) on SCDS Dataset**

MACHINES:

- **Adult Data Set:**
  - **SOR: 200 MHz Pentium Pro, 64 Meg RAM, Solaris 5.6, GNU C++**
  - **SMO: 266 MHz Pentium II, Windows NT 4, MS Visual C++ 5.0**
- **Other Data Sets: Cluster of 4 Workstations with 16 UltraSPARC II 250 MHz Processors, 2 Gig RAM Each. Total: 64 Processors, 8 Gig RAM**

SOR Performance Versus True Separability on 50,000 Points in R^{32}

True Separability	CPU Minutes To 95% of True Separability To Final Convergence	Iterations To 95% of True Separability To Final Convergence
80.0%	303.0 498.4	301,500 559,898
85.0%	83.3 334.6	76,000 445,213
90.0%	51.9 260.9	63,000 443,868
95.0%	39.2 230.6	62,500 351,266
98.5%	15.2 178.3	24,500 228,788
99.5%	4.5 127.8	9,000 129,535

SOR Performance Versus Training Set Separability
50,000 Point Dataset in R^{32}

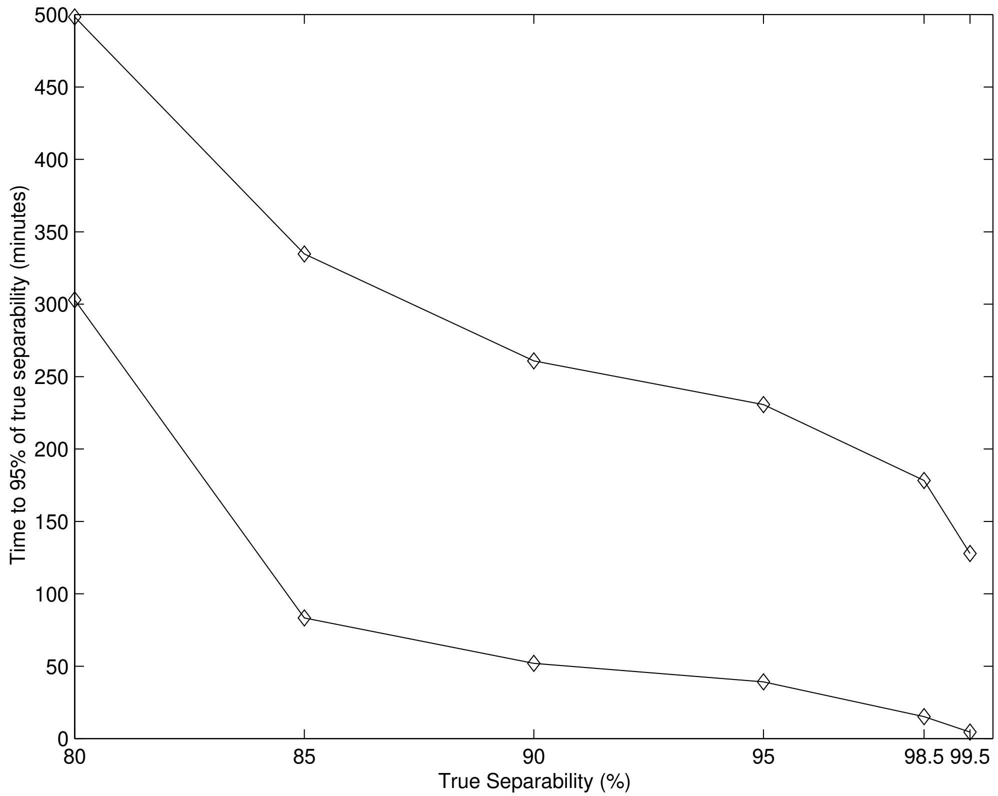


Figure 1: **Time to convergence (top curve) and time to 95% of true separability (bottom curve) versus true training set separability**

SMO: Sequential Minimal Optimization
Platt 1998

- **Iterative method for solving large box-constrained convex QPs with one linear equality constraint**
- **Updates two variables at a time by minimizing the objective function subject to the equality constraint and the box constraints**
- **"Finite termination" of SMO is based on a quadratic programming chunking method due to Osuna-Freund-Girosi 1997**

SOR & SMO Comparison on Adult Data Set in R^{121}

Training Set Size	Time (CPU sec) SOR SMO	Non-Bound SVs SOR SMO	Bound SVs SOR SMO	Iterations SOR SMO
4781	4.0 3.6	47 63	1559 1791	1166 9847
6414	6.0 5.5	65 61	2379 2370	1523 10669
11221	22.0 17.0	76 79	4147 4079	3308 17128
16101	46.0 35.3	82 67	5861 5854	4113 22770
22697	64.0 85.7	103 88	8168 8209	4180 35822
32561	64.0 163.6	177 149	11574 11558	9927 44774

SOR & SMO Comparison on Adult Data Set in R^{121}

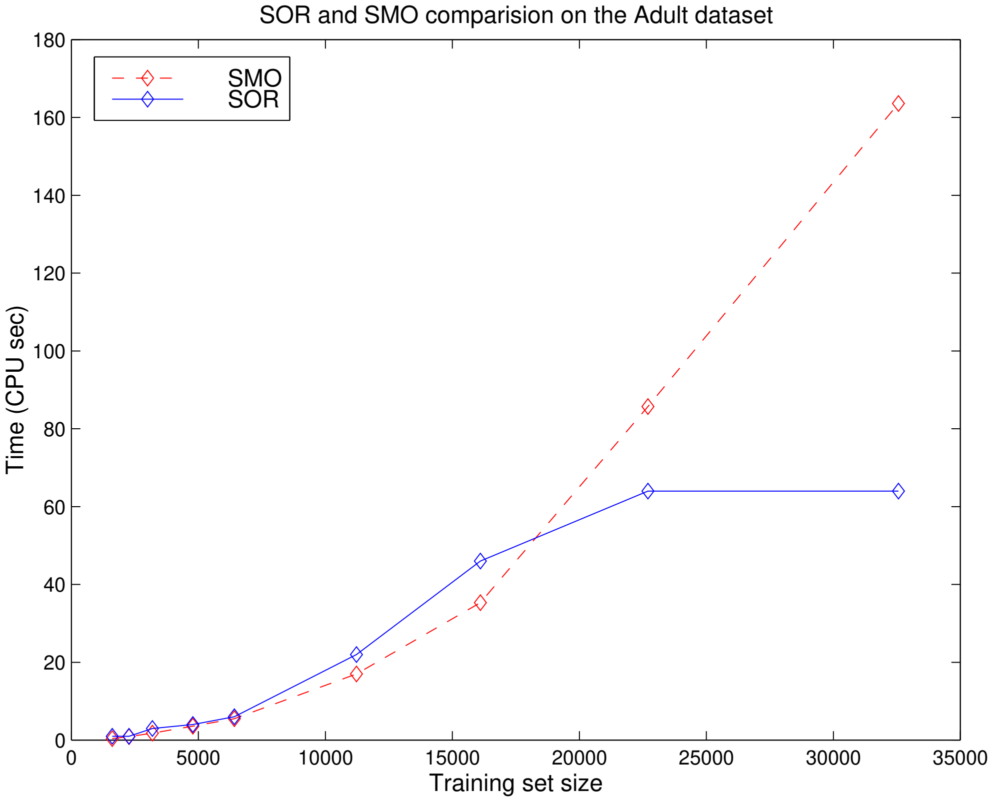


Figure 2: **Time to convergence versus training set size for SMO and SOR**

LPC: Linear Programming Chunking
Bradley & OLM 1998

- **LP-based finite method for solving linear programs with very large number of constraints**
- **Sweep through chunks of the constraints while maintaining constraints with positive multipliers**
- **Monotonically nondecreasing objective function appraoching minimum from below in a finite number of steps**

SOR & LPC on 1 Million Point Dataset in R^{32}

Training Set Size	Time (CPU hrs)	Time to 99.7% Accuracy (CPU hrs)
	SOR	SOR
	LPC	LPC
1,000,000	9.7	0.3
	231.1	—

SOR 10 Million Point Dataset in R^{32}

Training Set Size	Time to 95% of true separability	Iterations
10,000,000	14.3 hours	10,000

Conclusion

- SOR is a versatile method for large dataset discrimination
- SOR can handle problems not (easily) solvable by other methods
- Very well suited for support vector machine training
- SOR converges faster than other methods that require presence of substantial data in memory
- SOR scales up with virtually no changes
- Parallel versions already in place and tested on 9.4 million variable problems
- Future directions:
  - massive multicategory discrimination via SOR
  - nonlinear SVM training via SOR