A new interior-point optimization approach for support vector machines for binary classification and outlier detection

Jordi Castro

Group of Mathematical Optimization
Department of Statistics and Operations Research
Institute of Mathematics UPC – IMTech
Universitat Politècnica de Catalunya – BarcelonaTech
Barcelona, Catalonia

New Bridges between Mathematics and Data Science
8–11 November 2021, Valladolid, Spain

Supported by MCIN/AEI/FEDER RTI2018-097580-B-I00

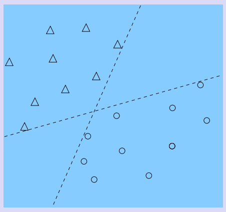
Outline

1. The 2-class and 1-class Support Vector Machine (SVM) problem
2. IPM for block-structured and large-scale problems
3. Results with 2-class SVM problem using real-world instances
4. Results with 1-class SVM problem using same real-world instances

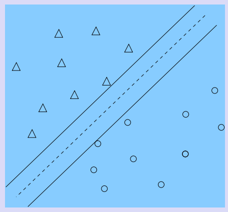
2-class SVM or Support Vector Classifier (SVC)

- Binary supervised classification technique. Useful for text classification.
- Purpose: to find two parallel hyperplanes separating two classes such that we both minimize the classification error and maximize the margin between the two separating hyperplanes:

Bad classification



Good classification

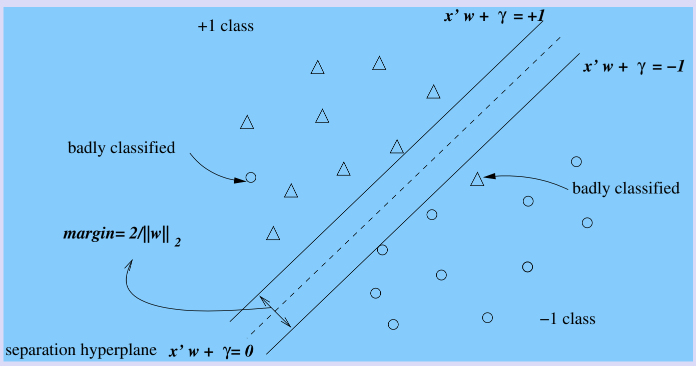


Parameters of the problem

- p points of d features: $x_i \in \mathbb{R}^d \;\; i = 1, \ldots, p$.
- Labels $y_i \in \{+1, -1\} \;\; i = 1, \ldots, p$: class of point i.

Modelling 2-class SVMs

Find hyperplane $(w, \gamma) \in \mathbb{R}^d \times \mathbb{R}$ maximizing separation margin between half-spaces $w^\top x + \gamma \geq +1$ and $w^\top x + \gamma \geq -1$, and minimizing misclassification. These two opposite objectives are weighted by parameter $\nu \in \mathbb{R}^+$.



We consider artificial variables $s_i \geq 0$, $i = 1, \ldots, p$, one for each point, to account for misclassification errors. The resulting constraints are:

$$y_i(w^\top x_i + \gamma) + s_i \geq 1 \quad i = 1, \ldots, p$$

2-class SVM as a quadratic optimization problem

Primal formulation: QO problem in variables w, γ, s

$$\begin{array}{rll} \min\limits_{(w,\gamma,s)\in\mathbb{R}^{d+1+p}} & \frac{1}{2}w^\top w + \nu e^\top s & \\ \text{s. to} & Y(Aw+\gamma e)+s \geq e & [\lambda \in \mathbb{R}^p] \\ & s \geq 0 & [\mu \in \mathbb{R}^p] \end{array}$$

where $Y = diag(y_1, \dots, y_p)$ and $A = [x_1 x_2 \dots x_p]^\top$ stores rowwise vectors $x_i \in \mathbb{R}^d$.

Dual formulation of 2-class SVM: QO problem in variables λ

$$\begin{array}{rl} \max\limits_{\lambda\in\mathbb{R}^p} & \lambda^\top e - \frac{1}{2}\lambda^\top YAA^\top Y\lambda \\ & \lambda^\top Ye = 0 \\ & 0 \leq \lambda \leq \nu e \end{array}$$

Computationally expensive for interior-point solvers

- Systems with $AA^\top$ to be solved in either primal or dual formulation.
- $AA^\top$ might be almost dense, of size p and rank $\min\{p,d\}$.

1-class SVM for Outlier Detection

Purpose of 1-class SVM

- Find hyperplane separating outliers from the rest of points, with maximum margin wrt. the origin.
- Parameter ν is an upper bound on fraction of detected outliers (Schölkopf, Platt, Shawe-Taylor, Smola, Neural Computation 2001) (Chou, Lin, Lin, SIAM Conf. Data Mining, 2020).

Primal formulation of 1-class SVM

$$\begin{array}{rll} \min\limits_{(w,\gamma,s)\in\mathbb{R}^{d+1+p}} & \frac{1}{2}w^\top w - \gamma + \frac{1}{\nu p}e^\top s & \\ \text{s. to} & Aw - \gamma e + s \geq 0 & [\lambda \in \mathbb{R}^p] \\ & s \geq 0 & [\mu \in \mathbb{R}^p] \end{array}$$

Dual formulation of 1-class SVM

$$\begin{array}{rl} \max\limits_{\lambda\in\mathbb{R}^p} & -\frac{1}{2}\lambda^\top AA^\top\lambda \\ & \lambda^\top e = 1 \\ & 0 \leq \lambda \leq \frac{1}{\nu p}e \end{array}$$

Computationally expensive for interior-point solvers

Most efficient IPM approaches for (only) 2-class SVM

Standard dual of 2-class SVM

$$\begin{array}{ll} \max\limits_{\lambda} & \lambda^\top e - \frac{1}{2}\lambda^\top YAA^\top Y\lambda \\ & \lambda^\top Ye = 0 \\ & 0 \le \lambda \le \nu \end{array}$$

Efficient IPMs devised when number of features is small: $d \ll p$.

Ferris, Mundson, SIOPT 2003

- Low-rank updates by Sherman-Morrison-Woodbury for Newton directions.
- Solved random data with millions of points but only 34 features.

Gondzio, Woodsend, COAP 2011: SVM-OOPS

- Separable reformulation defining extra variables u of dimension number of features:
- SVM-OOPS applied to real-world instances.

$$\begin{array}{ll} \max\limits_{\lambda} & \lambda^\top e - \frac{1}{2}u^\top u \\ & \lambda^\top Ye = 0 \\ & A^\top Y\lambda = u \\ & 0 \le \lambda \le \nu, \quad u \text{ free} \end{array}$$

Best SVM packages in machine learning community

LIBSVM for linear/nonlinear kernels (Chang, Lin, ACM TIST, 2011)

- Solves the dual of 2-class and 1-class SVM formulation.
- Uses the SMO algorithm, specific for dual SVM problems.

LIBLINEAR for linear kernels (Fan et al., JMLR, 2008)

- For 2-class SVM it transforms the problem to a “similar” unconstrained one without γ: It either solves the primal

$$\min_{w} \frac{1}{2}w^\top w + \nu \sum_{i=1}^{p} max(0, 1 - y_i w^\top x_i)^2$$

or the dual

$$\begin{array}{ll} \max\limits_{\lambda} & \lambda^\top e - \frac{1}{2}\lambda^\top YAA^\top Y\lambda \\ & 0 \le \lambda \le \nu \end{array}$$

using a trust-region CG Newton method or a coordinate descent algorithm.

- For 1-class SVM it solves the dual including the (removed) linear constraint.

Our new proposal: solve a set of linked smaller SVMs

Use multiple variable splitting:

1. Partition the dataset $A \in \mathbb{R}^{p \times d}$ in k subsets $A^i \in \mathbb{R}^{p_i \times d}, i = 1, \ldots, k$.
2. Consider k smaller SVMs, each with its own $(w^i, \gamma^i, s^i), i = 1, \ldots, k$ variables.
3. Link problems through constraints $(w^i, \gamma^i) = (w^{i+1}, \gamma^{i+1})$.

Complexity of Cholesky factorizations:

- of $AA^\top$ is $O(p^3)$.
- of $A^i A^{i\top}$ for $i = 1, \ldots, k$: $O\left(k \left(\frac{p}{k}\right)^3\right) = O\left(\frac{p^3}{k^2}\right)$.

New primal SVM formulation with multiple variable splitting:

$$
\begin{array}{lll}
\min\limits_{(w^i, \gamma^i, s^i)\ i=1,\ldots,k} & \frac{1}{2}\left(\sum_{i=1}^k w^{i\top} w^i\right)/k + \nu \sum_{i=1}^k \sum_{j=1}^{p_i} s_j^i & \\
\text{s. to} & Y^i(A^i w^i + \gamma^i e) + s^i \geq e & i = 1, \ldots, k \\
 & s^i \geq 0 & i = 1, \ldots, k \\
 & w^i = w^{i+1}, \quad \gamma^i = \gamma^{i+1} & i = 1, \ldots, k-1
\end{array}
$$

Specialized IPM for block-structured problems with linking constraints

- Developed and improved along several papers: EJOR 2021, OM&S 2021, SIOPT 2017, MP 2017, OM&S 2016, EJOR 2013, MP 2011, COAP 2007, AnnOR 2004, SIOPT 2000.
- Implemented in the BlockIP solver (C++, $\approx$ 19000 lines of code)
- Relies on a combination of Cholesky and PCG for computing directions.
- It can be applied to the new SVM formulation.

Formulation of structured problems with linking constraints

For convex separable problems (f_i convex separable)

$$\begin{array}{ll} \min & \sum_{i=0}^{k} f_i(x^i) \\ \text{subject to} & \begin{bmatrix} N_1 & & & \\ & \ddots & & \\ & & N_k & \\ L_1 & \dots & L_k & I \end{bmatrix} \begin{bmatrix} x^1 \\ \vdots \\ x^k \\ x^0 \end{bmatrix} = \begin{bmatrix} b^1 \\ \vdots \\ b^k \\ b^0 \end{bmatrix} \\ & 0 \le x^i \le u^i \quad i = 0, \dots, k. \end{array}$$

In the SVM problem, function is convex quadratic

- $f_i(x^i) = {c^i}^\top x^i + \frac{1}{2} {x^i}^\top Q_i x^i$, $Q_i \succeq 0$ diagonal

Solving normal equations by exploiting structure

Exploiting structure of A and Θ

$$A = \begin{bmatrix} N_1 & & & \\ & \ddots & & \\ & & N_k & \\ L_1 & \dots & L_k & I \end{bmatrix} \qquad \Theta = \begin{bmatrix} \Theta_1 & & & \\ & \ddots & & \\ & & \Theta_k & \\ & & & \Theta_0 \end{bmatrix}$$

$$A\Theta A^\top = \left[\begin{array}{ccc|c} N_1\Theta_1 N_1^\top & & & N_1\Theta_1 L_1^\top \\ & \ddots & & \vdots \\ & & N_k\Theta_k N_k^\top & N_k\Theta_k L_k^\top \\ \hline L_1\Theta_1 N_1^\top & \dots & L_k\Theta_k N_k^\top & \Theta_0 + \sum_{i=1}^{k} L_i\Theta_i L_i^\top \end{array}\right] = \begin{bmatrix} B & C \\ C^\top & D \end{bmatrix}$$

The Schur complement

$$\begin{bmatrix} B & C \\ C^\top & D \end{bmatrix} \begin{bmatrix} \Delta\lambda_1 \\ \Delta\lambda_2 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \Longleftrightarrow \begin{array}{rl} (D - C^\top B^{-1} C)\Delta\lambda_2 = & (g_2 - C^\top B^{-1} g_1) \\ B\Delta\lambda_1 = & (g_1 - C\Delta y_2) \end{array}$$

- System with B solved by k Cholesky factorizations.
- Schur complement $S = D - C^\top B^{-1} C$ with large fill-in: system solved by PCG.

The preconditioner

Based on P-regular splitting $S = D - (C^\top B^{-1} C)$ (SIOPT00,COAP07)

Spectral radius of $D^{-1}(C^\top B^{-1} C))$ satisfies $\rho(D^{-1}(C^\top B^{-1} C))) < 1$ and then

$$(D - C^\top B^{-1} C)^{-1} = \left(\sum_{i=0}^{\infty} (D^{-1}(C^\top B^{-1} C))^i \right) D^{-1}$$

Preconditioner M^{-1} obtained truncating the power series at term h

$$\begin{array}{lcll} M^{-1} & = & D^{-1} & \text{if } h = 0, \\ M^{-1} & = & (I + D^{-1}(C^\top B^{-1} C))D^{-1} & \text{if } h = 1. \end{array}$$

Quality of preconditioner depends on

- $\rho < 1$: the farther from 1, the better the preconditioner.
- Factorization of D: the easier and sparser, the better.

Exploit structure of linking constraints $x^i - x^{i+1} = 0$

For instance for $k = 4$ blocks:

$$[L_1 \ L_2 \ L_3 \ L_4] = \begin{bmatrix} I & 0 & -I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & -I & 0 \end{bmatrix}$$

Then:

$$D = \Theta_0 + \sum_{i=1}^{k} L_i \Theta_i L_i^\top = \Theta_0 + \begin{bmatrix} \Theta_1^x + \Theta_2^x & -\Theta_2^x & 0 \\ -\Theta_2^x & \Theta_2^x + \Theta_3^x & -\Theta_3^x \\ 0 & -\Theta_3^x & \Theta_3^x + \Theta_4^x \end{bmatrix}$$

Properties of D

- D is a shifted tri-diagonal matrix.
- Very sparse, efficient to factorize: good preconditioner.
- Specific routines can be developed for its factorization.

Sizes of real-world 2-class SVM instances

	Instance	#blocks† k	#points p	#features d
d small (few features)	a9a	100	32561	123
	australian	2	690	14
	covtype	10000	581012	54
	ijcnn1	1000	49990	22
	madelon	10	2000	500
	mnist-ge5-lt5	2000	60000	780
	mnist-odd-even	2000	60000	780
	mushrooms	20	8124	112
	sensit-combined	1000	78823	100
	usps	100	7291	256
	w1a	10	2477	300
	w4a	30	7366	300
	w8a	200	49749	300
d large	colon-cancer	10	62	2000
	gisette	100	6000	5000
	leu	2	38	7129
	news20	40	19996	1355191
	rcv1	40	20242	47236
	real-sim	100	72309	20958

† Only used for SVM-BlockIP and CPLEX-20.1 models

CPU time with interior-point approaches

Instance	SVM-BlockIP	CPLEX-20.1	SVM-OOPS†
a9a	0.7	0.5	1.7
australian	0.0	0.1	0.0
covtype	23.9	5.2	12.6
ijcnn1	1.1	0.4	0.6
madelon	0.2	3.8	0.9
mnist-ge5-lt5	15.6	24.7	74.4
mnist-odd-even	12.5	28.8	87.1
mushrooms	1.7	0.1	0.3
sensit-combined	2.7	9.8	7.5
usps	0.3	† 18.7	1.6
w1a	0.1	† 0.6	0.2
w4a	0.5	† 3.6	0.8
w8a	4.3	1.8	25.3
colon-cancer	0.2	0.0	6.2
gisette	3.2	54.0	314.9
leu	0.1	0.1	—
news20	84.8	968.3	—
rcv1	7.9	1236.7	—
real-sim	14.4	40484.8	—

† $k > 1$ blocks used

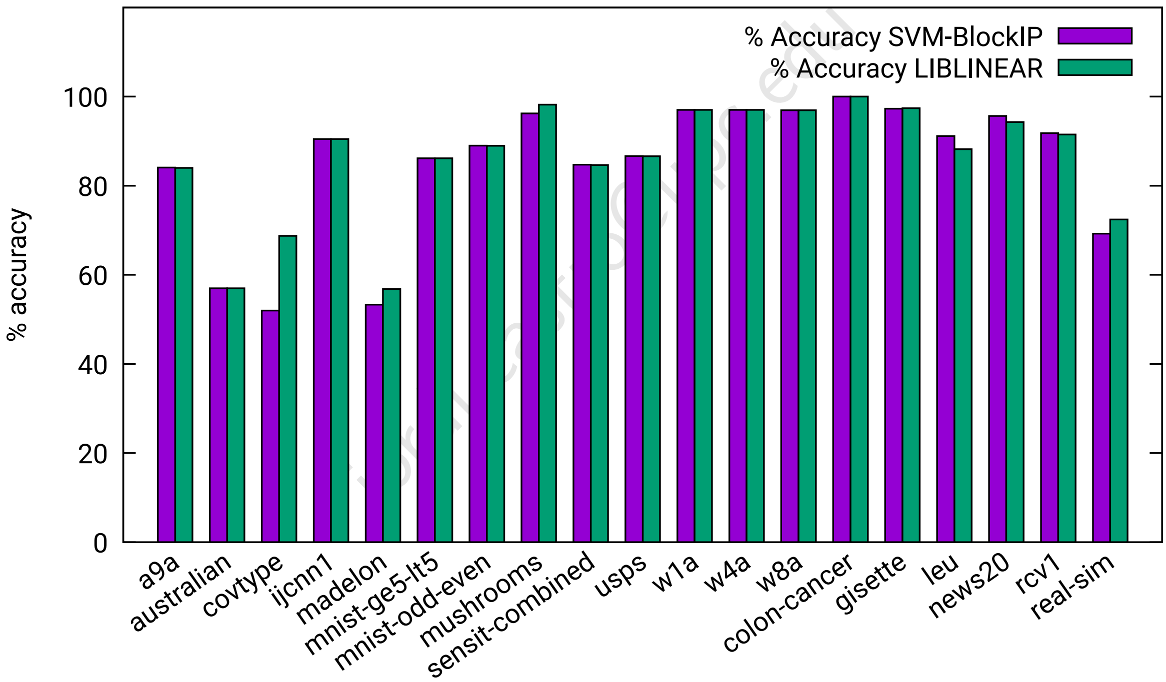
CPU time with LIBSVM and LIBLINEAR(dual)

Instance	SVM-BlockIP	LIBSVM	LIBLINEAR†
a9a	0.7	32.7	4.9
australian	0.0	0.0	0.1
covtype	23.9	9773.6	483.1
ijcnn1	1.1	13.4	25.5
madelon	0.2	2.6	1.8
mnist-ge5-lt5	15.6	1064.2	59.9
mnist-odd-even	12.5	817.4	45.9
mushrooms	1.7	1.2	2.8
sensit-combined	2.7	853.2	92.3
usps	0.3	11.1	9.2
w1a	0.1	0.0	0.2
w4a	0.5	0.2	2.3
w8a	4.3	7.8	15.8
colon-cancer	0.2	0.0	0.0
gisette	3.2	42.3	14.1
leu	0.1	0.1	0.0
news20	84.8	511.4	111.8
rcv1	7.9	151.4	12.2
real-sim	14.4	1777.8	91.2

† Solves different problem, without γ

Classification accuracy of SVM-BlockIP and LIBLINEAR

Similar accuracies for both codes, and, excluding 4 instances, always $\geq 80\%$



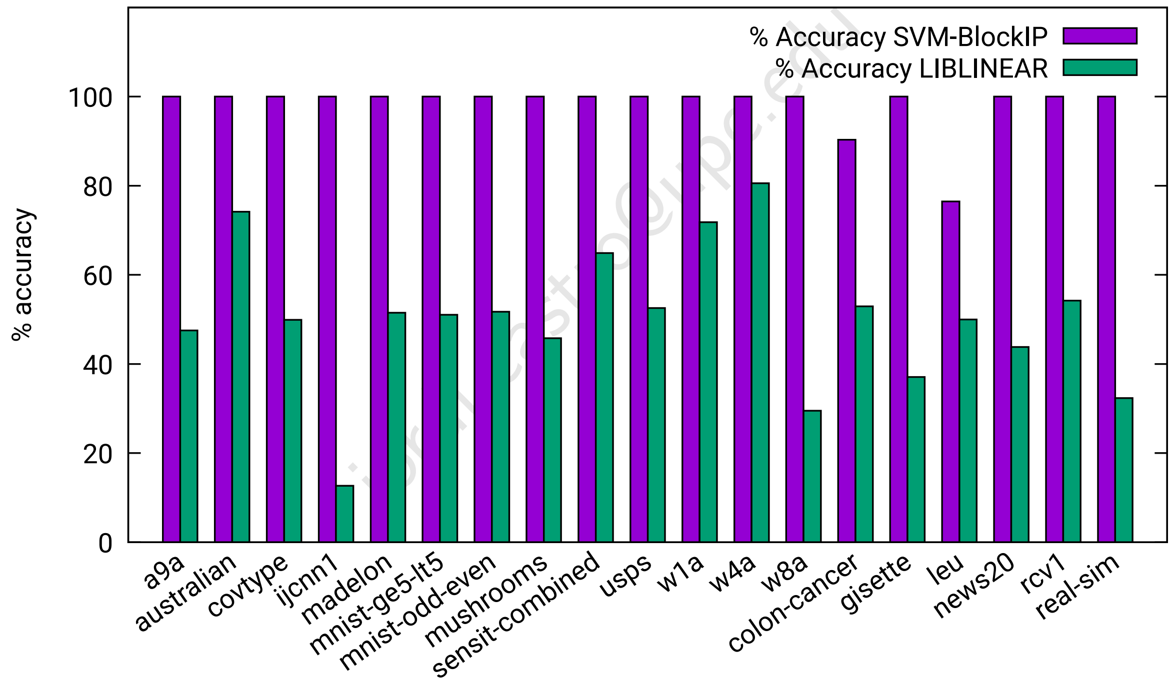
CPU time with all available packages

Instance	SVM-BlockIP	CPLEX-20.1	LIBSVM†	LIBLINEAR†
a9a	0.5	0.4	5.7	0.1
australian	0.1	0.1	0.0	0.0
covtype	10.6	4.2	1030.8	1.0
ijcnn1	1.0	0.3	5.5	0.1
madelon	0.2	8.3	0.4	0.1
mnist-ge5-lt5	4.5	23.5	328.5	1.1
mnist-odd-even	4.6	23.5	331.2	1.1
mushrooms	0.9	0.1	0.4	0.0
sensit-combined	1.6	5.7	79.1	1.1
usps	0.2	100.1	2.0	0.2
w1a	0.1	1.2	0.0	0.0
w4a	0.3	8.8	0.3	0.0
w8a	2.3	1.1	13.3	0.1
colon-cancer	0.1	0.0	0.0	0.0
gisette	1.6	79.5	16.5	0.5
leu	0.1	0.1	0.1	0.1
news20	55.2	2398.8	58.5	1.5
rcv1	4.2	1972.3	17.3	0.2
real-sim	15.1	91542.9	153.2	0.6

† Poor solutions provided (see next slide)

1-class SVM accuracy of SVM-BlockIP and LIBLINEAR

SVM-BlockIP always provides better solutions



Conclusions

BlockIP and SVMs by multiple variable splitting

- BlockIP competitive with state-of-the-art solvers for SVMs.
- It could solve new SVM models whose duals involve linear constraints.

Further applications (other than SVMs) in Data Science

- Any constrained problem which allows multiple variable splitting.

Thanks for your attention