An Exact SDP based Approach for Semi-Supervised Support Vector Machines

May 22, 2024

Joint work with Veronica Piccialli* and Antonio M. Sudoso

*Veronica Piccialli's work has been supported by PNRR MUR project PE0000013-FAIR

FAIR Future Artificial Intelligence Research

FWF Der Wissenschaftsfonds.

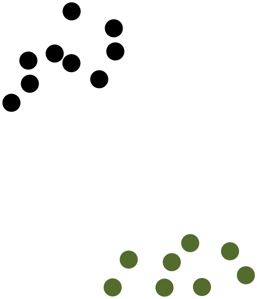
UNIVERSITÄT KLAGENFURT

Modeling - Analysis - Optimization DOC.FUNDS DOCTORAL SCHOOL

Support Vector Machines (SVMs) Vapnik & Chervonenkis (1963)

Input

- training set $\mathcal{T} = \{(x_i, y_i),\ i = 1, \dots, n,\ x_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\}\}$



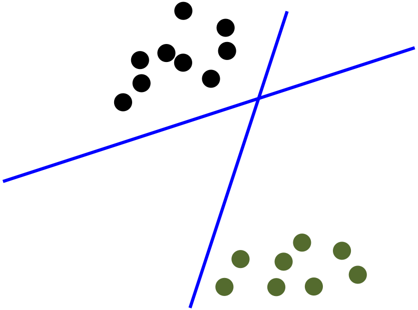
Support Vector Machines (SVMs) Vapnik & Chervonenkis (1963)

Input

- training set $\mathcal{T} = \{(x_i, y_i),\ i = 1, \ldots, n,\ x_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\}\}$

Goal/Output

- separating hyperplane $w^\top x + b = 0$



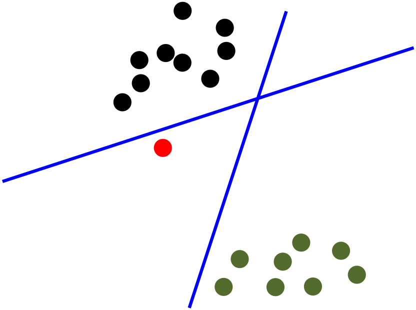
Support Vector Machines (SVMs) Vapnik & Chervonenkis (1963)

Input

- training set $\mathcal{T} = \{(x_i, y_i),\ i = 1, \dots, n,\ x_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\}\}$

Goal/Output

- separating hyperplane $w^\top x + b = 0$
- decision function $y(x) = \text{sign}(w^\top x + b)$ for new data



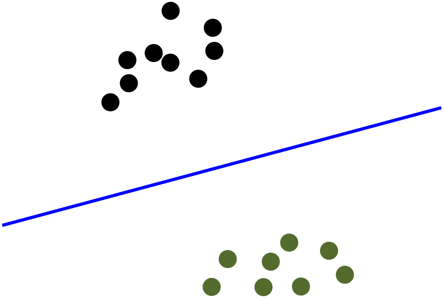
Support Vector Machines (SVMs) Vapnik & Chervonenkis (1963)

Input

- training set $\mathcal{T} = \{(x_i, y_i),\ i = 1, \dots, n,\ x_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\}\}$

Goal/Output

- separating hyperplane $w^\top x + b = 0$
- decision function $y(x) = \text{sign}(w^\top x + b)$ for new data



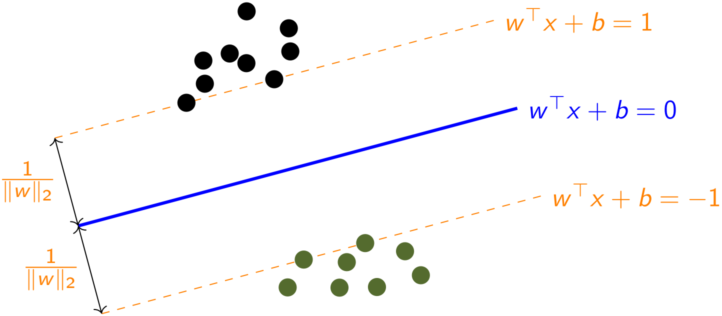
Support Vector Machines (SVMs) Vapnik & Chervonenkis (1963)

Input

- training set $\mathcal{T} = \{(x_i, y_i),\ i = 1, \ldots, n,\ x_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\}\}$

Goal/Output

- separating hyperplane $w^\top x + b = 0$ (maximum margin)
- decision function $y(x) = \operatorname{sign}(w^\top x + b)$ for new data



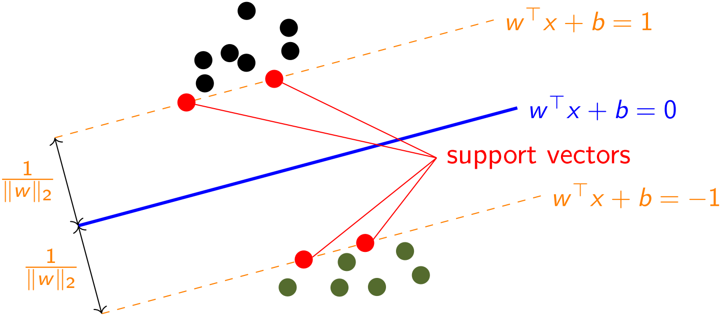
Support Vector Machines (SVMs) Vapnik & Chervonenkis (1963)

Input

- training set $\mathcal{T} = \{(x_i, y_i),\ i = 1, \ldots, n,\ x_i \in \mathbb{R}^d,\ y_i \in \{-1, 1\}\}$

Goal/Output

- separating hyperplane $w^\top x + b = 0$ (maximum margin)
- decision function $y(x) = \text{sign}(w^\top x + b)$ for new data



Hard margin approach

Maximum hard margin hyperplane

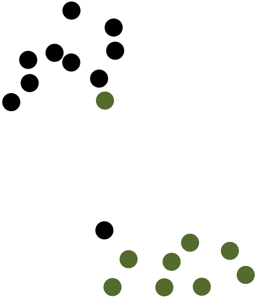
$$\begin{aligned} \min_{w,b} \quad & \tfrac{1}{2}\|w\|_2^2 \\ \text{s. t.} \quad & y_i[w^\top x_i + b] \geq 1,\ \ i = 1, \dots, n \end{aligned}$$

Hard margin approach

Maximum hard margin hyperplane

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{s. t.} \quad & y_i[w^\top x_i + b] \geq 1, \ i = 1, \ldots, n \end{aligned}$$

Question: What if the data is **not** linearly separable?



Soft margin approach Cortes & Vapnik (1995)

Maximum soft margin hyperplane w.r.t. $C > 0$

- data ‘almost’ linearly separable ⇒ allow misclassifications
- introduce slack variables ξ_i and add penalty term to objective:

Soft margin approach Cortes & Vapnik (1995)

Maximum soft margin hyperplane w.r.t. $C > 0$

- data 'almost' linearly separable ⇒ allow misclassifications
- introduce slack variables ξ_i and add penalty term to objective:

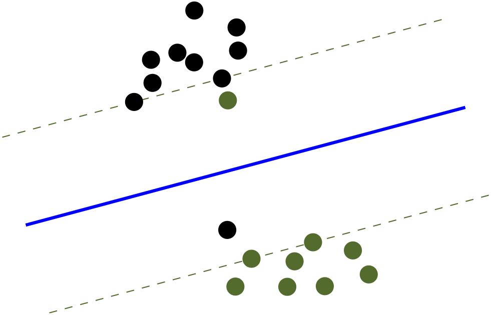
$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\|w\|_2^2 + C \textstyle\sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i[w^\top x_i + b] \geq 1 - \xi_i, \;\; i = 1, \ldots, n \\ & \xi_i \geq 0, \;\; i = 1, \ldots, n \end{aligned}$$

Soft margin approach Cortes & Vapnik (1995)

Maximum soft margin hyperplane w.r.t. $C > 0$

- data ‘almost’ linearly separable ⇒ allow misclassifications
- introduce slack variables ξ_i and add penalty term to objective:

$$\begin{array}{ll} \min\limits_{w,b,\xi} & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{s. t.} & y_i[w^\top x_i + b] \geq 1 - \xi_i,\ i = 1, \dots, n \\ & \xi_i \geq 0,\ i = 1, \dots, n \end{array}$$

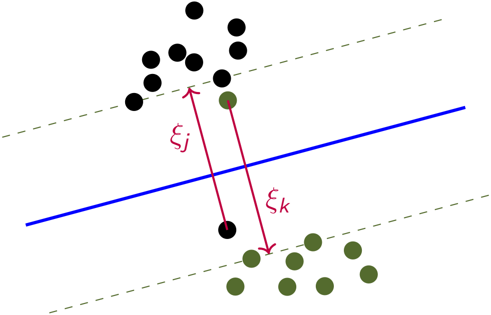


Soft margin approach Cortes & Vapnik (1995)

Maximum soft margin hyperplane w.r.t. $C > 0$

- data 'almost' linearly separable ⇒ allow misclassifications
- introduce slack variables ξ_i and add penalty term to objective:

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\|w\|_2^2 + C\textstyle\sum_{i=1}^n \xi_i \\ \text{s. t.} \quad & y_i[w^\top x_i + b] \geq 1 - \xi_i,\ \ i = 1, \dots, n \\ & \xi_i \geq 0,\ \ i = 1, \dots, n \end{aligned}$$

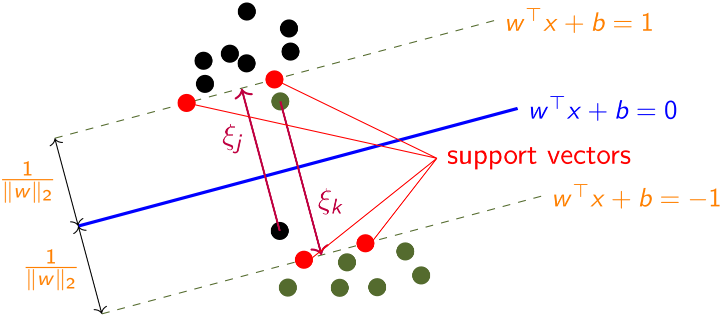


Soft margin approach Cortes & Vapnik (1995)

Maximum soft margin hyperplane w.r.t. $C > 0$

- data ‘almost’ linearly separable ⇒ allow misclassifications
- introduce slack variables ξ_i and add penalty term to objective:

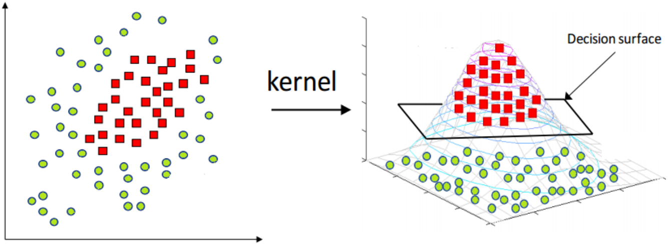
$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n \xi_i \\ \text{s. t.} \quad & y_i[w^\top x_i + b] \geq 1 - \xi_i,\ i = 1,\dots,n \\ & \xi_i \geq 0,\ i = 1,\dots,n \end{aligned}$$



Nonlinear SVMs: the kernel trick Boser, Guyon, Vapnik (1992)

Kernel trick

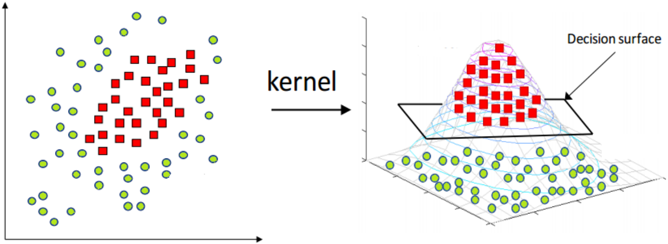
Map data into a higher-dimensional space via $\phi\colon \mathbb{R}^d \to \mathbb{R}^m$, $m \geq d$. Then find a separating hyperplane in the new space.



Nonlinear SVMs: the kernel trick Boser, Guyon, Vapnik (1992)

Kernel trick

Map data into a higher-dimensional space via $\phi \colon \mathbb{R}^d \to \mathbb{R}^m$, $m \geq d$. Then find a separating hyperplane in the new space.



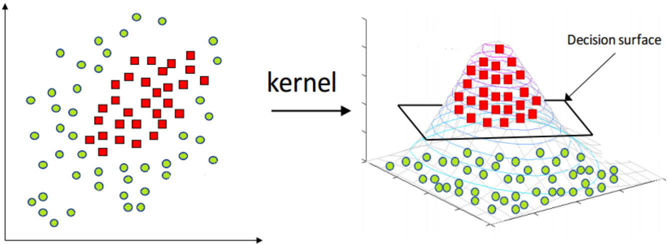
- linear or polynomial kernel, radial basis function kernel, . . .
- no explicit mapping into higher dimension via kernel function

$$k(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle$$

Nonlinear SVMs: the kernel trick Boser, Guyon, Vapnik (1992)

Kernel trick

Map data into a higher-dimensional space via $\phi\colon \mathbb{R}^d \to \mathbb{R}^m$, $m \geq d$. Then find a separating hyperplane in the new space.

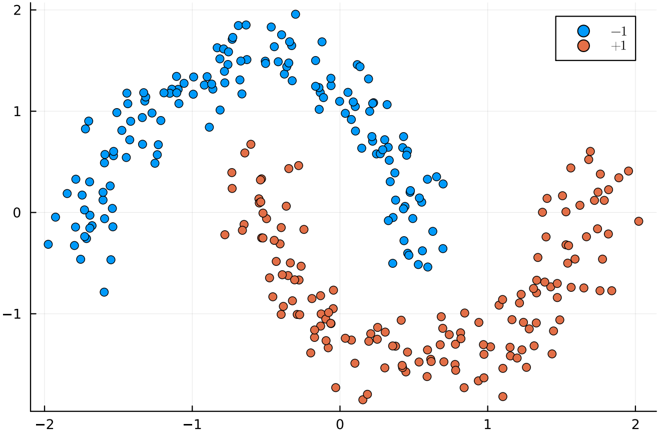


- linear or polynomial kernel, radial basis function kernel, ...
- no explicit mapping into higher dimension via kernel function

$$k(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle$$

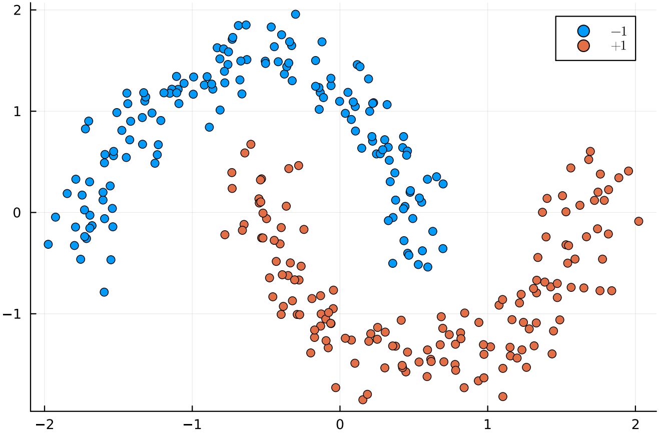
- separator is nonlinear in the original space

Example: two moons dataset



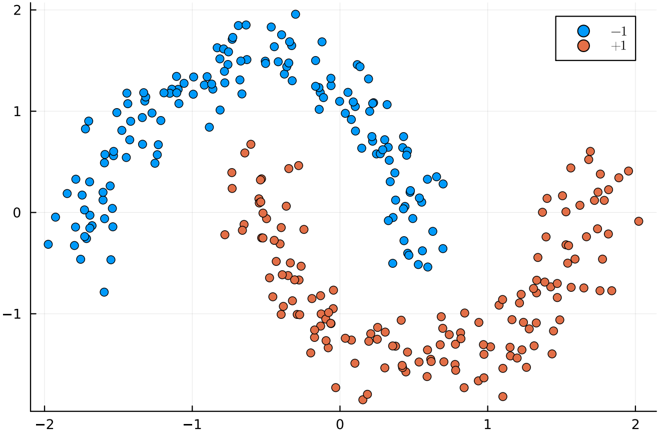
- linear kernel inappropriate

Example: two moons dataset



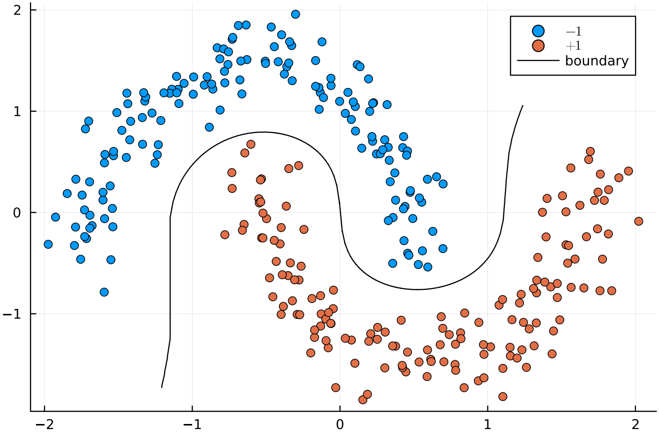
- linear kernel inappropriate
- RBF kernel: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2),\ \gamma > 0$

Example: two moons dataset



- linear kernel inappropriate
- RBF kernel: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2),\ \gamma > 0$
- we choose $C = 1$ and $\gamma = 0.5$ here

Example: two moons dataset



- linear kernel inappropriate
- RBF kernel: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2),\ \gamma > 0$
- we choose $C = 1$ and $\gamma = 0.5$ here

Semi-supervised Support Vector Machines (S3VMs)

Bennett & Demiriz (1998)

Input

- n data points $x_i \in \mathbb{R}^d, \; i = 1, \dots, n$
- ℓ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\}, \; i = 1, \dots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$

Semi-supervised Support Vector Machines (S3VMs)

Bennett & Demiriz (1998)

Input

- n data points $x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$
- ℓ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\},\ i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$

Assumption

All data points are centered around the origin ($\Rightarrow b = 0$).

Semi-supervised Support Vector Machines (S3VMs)

Bennett & Demiriz (1998)

Input

- n data points $x_i \in \mathbb{R}^d, \; i = 1, \ldots, n$
- ℓ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\}, \; i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$

Assumption

All data points are centered around the origin ($\Rightarrow b = 0$).

Kernel-based S3VM model

$$\begin{aligned} \min_{w,\xi,y^u} \quad & \tfrac{1}{2}\|w\|_2^2 + C_l \textstyle\sum_{i=1}^{\ell} \xi_i^2 + C_u \sum_{i=l+1}^{n} \xi_i^2 \\ \text{s. t.} \quad & y_i \, w^\top \phi(x_i) \geq 1 - \xi_i, \; i = 1, \ldots, n \\ & y^u := (y_{\ell+1}, \ldots, y_n) \in \{-1, +1\}^{n-\ell} \end{aligned}$$

Dual reformulation of S3VM model

Reformulation as non-convex QCQP Bai & Yan (2016)

$$\begin{aligned} \min \quad & v^\top C v \\ \text{s. t.} \quad & y_i v_i \geq 1,\ i = 1, \ldots, \ell \\ & v_i^2 \geq 1,\ i = \ell + 1, \ldots, n \\ & v \in \mathbb{R}^n \end{aligned}$$

- quadratic programming problem in continuous variables
- symmetric positive definite $C \Rightarrow$ convex objective function
- nonconvex feasible set
- bound constraints: $y_i v_i \geq 1$ means either $v_i \leq -1$ or $v_i \geq 1$

Balancing constraint

Chapelle & Zien (2005): balancing constraint for linear kernel

$$\frac{1}{n-\ell}\sum_{i=\ell+1}^{n}\operatorname{sign}(w^\top x_i)=\frac{1}{\ell}\sum_{i=1}^{\ell}y_i$$

- no degenerate solutions (all unlab. data points in one class)
- enhances performance and robustness
- difficult to handle explicitly

Balancing constraint

Chapelle & Zien (2005): balancing constraint for linear kernel

$$\frac{1}{n-\ell}\sum_{i=\ell+1}^{n} \operatorname{sign}(w^\top x_i) = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i$$

- no degenerate solutions (all unlab. data points in one class)
- enhances performance and robustness
- difficult to handle explicitly

We use the following “relaxation” instead:

Soft-balancing constraint

$$\frac{1}{n-\ell}\sum_{i=\ell+1}^{n} v_i = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i$$

Global optimization problem

Textbook-like form

$$
\begin{aligned}
\min \quad & x^\top C x \\
\text{s. t.} \quad & L_i \le x_i \le U_i, \; i = 1, \dots, n \\
& x_i^2 \ge 1, \; i = 1, \dots, n \\
& x \in \mathbb{R}^n
\end{aligned}
$$

- rename variables
- C symmetric and positive definite
- $L_i \in \mathbb{R} \cup \{-\infty\}$ and $U_i \in \mathbb{R} \cup \{+\infty\}$
- some constraints redundant

Semidefinite programming (SDP) relaxation

Matrix-based reformulation

$$\begin{array}{ll} \min & \langle C, X\rangle \\ \text{s. t.} & L_i \le x_i \le U_i,\ i = 1, \ldots, n \\ & X_{ii} \ge 1,\ i = 1, \ldots, n \\ & X = xx^\top,\ x \in \mathbb{R}^n,\ X \in \mathcal{S}^n \end{array}$$

Semidefinite programming (SDP) relaxation

Matrix-based reformulation

$$\begin{array}{ll} \min & \langle C, X\rangle \\ \text{s.t.} & L_i \leq x_i \leq U_i,\ i = 1, \ldots, n \\ & X_{ii} \geq 1,\ i = 1, \ldots, n \\ & X = xx^\top,\ x \in \mathbb{R}^n,\ X \in \mathcal{S}^n \end{array}$$

We relax $X - xx^\top = 0$ to $X - xx^\top \succeq 0 \Leftrightarrow \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0$:

Semidefinite programming (SDP) relaxation

Matrix-based reformulation

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{s. t.} \quad & L_i \leq x_i \leq U_i,\ i = 1, \ldots, n \\ & X_{ii} \geq 1,\ i = 1, \ldots, n \\ & X = xx^\top,\ x \in \mathbb{R}^n,\ X \in \mathcal{S}^n \end{aligned}$$

We relax $X - xx^\top = 0$ to $X - xx^\top \succeq 0 \Leftrightarrow \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0$:

Semidefinite programming (SDP) relaxation Bai & Yan (2016)

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{s. t.} \quad & L_i \leq x_i \leq U_i,\ i = 1, \ldots, n \\ & X_{ii} \geq 1,\ i = 1, \ldots, n \\ & \bar{X} := \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0,\ x \in \mathbb{R}^n,\ X \in \mathcal{S}^n \end{aligned} \tag{SDP}$$

Optimality-based box constraints

Convex QCQP

$$\begin{aligned} L_i / U_i := \min / \max \quad & x_i \\ \text{s.t.} \quad & L_i \le x_i \le U_i, \; i = 1, \dots, n \\ & x^\top C x \le \text{UB} \\ & x \in \mathbb{R}^n \end{aligned} \tag{$*$}$$

- UB: upper bound on optimal S3VM objective
- $(*)$ is equivalent to convex problem with just bound constraints

Optimality-based box constraints

Convex QCQP

$$\begin{aligned} L_i / U_i := \min / \max \quad & x_i \\ \text{s. t.} \quad & L_i \leq x_i \leq U_i, \; i = 1, \ldots, n \\ & x^\top C x \leq \mathrm{UB} \\ & x \in \mathbb{R}^n \end{aligned} \qquad (*)$$

- UB: upper bound on optimal S3VM objective
- $(*)$ is equivalent to convex problem with just bound constraints

Dual problem for maximizing x_i

$$\begin{aligned} \min \quad & \frac{1}{4\mu} \left(e_i + \lambda^L - \lambda^U\right)^\top C^{-1} \left(e_i + \lambda^L - \lambda^U\right) - L^\top \lambda^L + U^\top \lambda^U + \mu \mathrm{UB} \\ \text{s. t.} \quad & \lambda^L, \lambda^U \geq 0, \; \mu \geq \varepsilon \end{aligned}$$

SDP relaxation with bounded main diagonal

More stable SDP relaxation

$$
\begin{aligned}
\min \quad & \langle C, X \rangle \\
\text{s. t.} \quad & L_i \leq x_i \leq U_i,\ i = 1, \dots, n \\
& 1 \leq X_{ii} \leq \max\{L_i^2, U_i^2\},\ i = 1, \dots, n \\
& \bar{X} = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0,\ x \in \mathbb{R}^n,\ X \in \mathcal{S}^n
\end{aligned}
\qquad (*)
$$

SDP relaxation with bounded main diagonal

More stable SDP relaxation

$$\begin{aligned} \min \quad & \langle C, X\rangle \\ \text{s. t.} \quad & L_i \leq x_i \leq U_i,\ i = 1, \ldots, n \\ & 1 \leq X_{ii} \leq \max\{L_i^2, U_i^2\},\ i = 1, \ldots, n \\ & \bar{X} = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0,\ x \in \mathbb{R}^n,\ X \in \mathcal{S}^n \end{aligned} \qquad (*)$$

For any feasible solution $\bar{X} \succeq 0$, we have:

$$\lambda_{\max}(\bar{X}) \leq \operatorname{trace}(\bar{X}) \leq 1 + \sum_{i=1}^{n} \max\{L_i^2, U_i^2\}$$

SDP relaxation with bounded main diagonal

More stable SDP relaxation

$$
\begin{aligned}
\min \quad & \langle C, X \rangle \\
\text{s.t.} \quad & L_i \leq x_i \leq U_i, \; i = 1, \dots, n \\
& 1 \leq X_{ii} \leq \max\{L_i^2, U_i^2\}, \; i = 1, \dots, n \\
& \bar{X} = \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0, \; x \in \mathbb{R}^n, \; X \in \mathcal{S}^n
\end{aligned}
\qquad (*)
$$

For any feasible solution $\bar{X} \succeq 0$, we have:

$$\lambda_{\max}(\bar{X}) \leq \operatorname{trace}(\bar{X}) \leq 1 + \sum_{i=1}^{n} \max\{L_i^2, U_i^2\}$$

- solvers can exploit this information
- helps to find dual bounds on $(*)$

Reformulation Linearization Technique cuts Sherali & Adams (1998)

For any $x_i, x_j,\ i, j = 1, \ldots, n$, we have:

- $U_i - x_i \geq 0$
- $x_i - L_i \geq 0$

- $U_j - x_j \geq 0$
- $x_j - L_j \geq 0$

Reformulation Linearization Technique cuts Sherali & Adams (1998)

For any $x_i, x_j,\ i, j = 1, \dots, n$, we have:

- $U_i - x_i \geq 0$
- $x_i - L_i \geq 0$

- $U_j - x_j \geq 0$
- $x_j - L_j \geq 0$

- $(U_i - x_i)(x_j - L_j) \geq 0 \quad \Leftrightarrow \quad X_{ij} \leq U_i x_j + L_j x_i - U_i L_j$

Reformulation Linearization Technique cuts Sherali & Adams (1998)

For any $x_i, x_j,\ i, j = 1, \ldots, n$, we have:

- $U_i - x_i \geq 0$
- $x_i - L_i \geq 0$
- $U_j - x_j \geq 0$
- $x_j - L_j \geq 0$

- $(U_i - x_i)(x_j - L_j) \geq 0 \quad \Leftrightarrow \quad X_{ij} \leq U_i x_j + L_j x_i - U_i L_j$

RLT cuts

$$X_{ij} \geq \max\{U_i x_j + U_j x_i - U_i U_j,\ L_i x_j + L_j x_i - L_i L_j\}$$
$$X_{ij} \leq \min\{L_i x_j + U_j x_i - L_i U_j,\ U_i x_j + L_j x_i - U_i L_j\}$$

Reformulation Linearization Technique cuts Sherali & Adams (1998)

For any $x_i, x_j,\ i,j = 1,\dots,n$, we have:

- $U_i - x_i \geq 0$
- $x_i - L_i \geq 0$
- $U_j - x_j \geq 0$
- $x_j - L_j \geq 0$

- $(U_i - x_i)(x_j - L_j) \geq 0 \quad \Leftrightarrow \quad X_{ij} \leq U_i x_j + L_j x_i - U_i L_j$

RLT cuts

$$X_{ij} \geq \max\{U_i x_j + U_j x_i - U_i U_j,\ L_i x_j + L_j x_i - L_i L_j\}$$
$$X_{ij} \leq \min\{L_i x_j + U_j x_i - L_i U_j,\ U_i x_j + L_j x_i - U_i L_j\}$$

- cutting-plane approach
- significant stronger lower bounds

Optimality-based tightening Ryoo & Sahinidis (1995)

- UB: best known upper bound for nonconvex problem (P)
- LB: optimal value of SDP relaxation

Optimality-based tightening Ryoo & Sahinidis (1995)

- UB: best known upper bound for nonconvex problem (P)
- LB: optimal value of SDP relaxation

Optimality-based tightening (in our setting)

Let $g(x, X) \leq 0$ be an active constraint in the SDP relaxation with corresponding optimal dual multiplier $\lambda > 0$. Then the constraint

$$g(x, X) \geq -\frac{\mathrm{UB} - \mathrm{LB}}{\lambda}$$

is valid for all solutions of (P) with objective value better than UB.

Optimality-based tightening Ryoo & Sahinidis (1995)

- UB: best known upper bound for nonconvex problem (P)
- LB: optimal value of SDP relaxation

Optimality-based tightening (in our setting)

Let $g(x, X) \leq 0$ be an active constraint in the SDP relaxation with corresponding optimal dual multiplier $\lambda > 0$. Then the constraint

$$g(x, X) \geq -\frac{\mathrm{UB} - \mathrm{LB}}{\lambda}$$

is valid for all solutions of (P) with objective value better than UB.

- $-\frac{\mathrm{UB}-\mathrm{LB}}{\lambda} \leq g(x, X) \leq 0$ for all optimal solutions (x, X) of (P)
- new constraint is convex

Marginals-based bound tightening Ryoo & Sahinidis (1995)

Bound tightening

If the constraint $L_i - x_i \leq 0$ is active at the optimal SDP solution with dual multiplier $\lambda_i^L > 0$, then the inequality

$$L_i - x_i \geq -\frac{\mathrm{UB} - \mathrm{LB}}{\lambda_i^L}$$

can be added to (P) and to the SDP relaxation.

Marginals-based bound tightening Ryoo & Sahinidis (1995)

Bound tightening

If the constraint $L_i - x_i \leq 0$ is active at the optimal SDP solution with dual multiplier $\lambda_i^L > 0$, then the inequality

$$L_i - x_i \geq -\frac{\mathrm{UB} - \mathrm{LB}}{\lambda_i^L}$$

can be added to (P) and to the SDP relaxation.

- if $\lambda_i^L > 0$, update U_i via $U_i \coloneqq \min\left\{U_i, L_i + \frac{\mathrm{UB}-\mathrm{LB}}{\lambda_i^L}\right\}$

Marginals-based bound tightening Ryoo & Sahinidis (1995)

Bound tightening

If the constraint $L_i - x_i \leq 0$ is active at the optimal SDP solution with dual multiplier $\lambda_i^L > 0$, then the inequality

$$L_i - x_i \geq -\frac{\mathrm{UB} - \mathrm{LB}}{\lambda_i^L}$$

can be added to (P) and to the SDP relaxation.

- if $\lambda_i^L > 0$, update U_i via $U_i \coloneqq \min\left\{U_i, L_i + \frac{\mathrm{UB}-\mathrm{LB}}{\lambda_i^L}\right\}$
- if $\lambda_i^U > 0$, update L_i via $L_i \coloneqq \max\left\{L_i, U_i - \frac{\mathrm{UB}-\mathrm{LB}}{\lambda_i^U}\right\}$

Applying optimality-based tightening to main diagonal

(x, X) feasible for (P) $\;\Rightarrow\; 1 \leq x_i^2 = X_{ii} \leq \max\{L_i^2, U_i^2\}$

Applying optimality-based tightening to main diagonal

(x, X) feasible for (P) $\Rightarrow\ 1 \leq x_i^2 = X_{ii} \leq \max\{L_i^2, U_i^2\}$

Lemma

Let $i \in \{1, \ldots, n\}$. If the constraint $X_{ii} \geq 1$ is active at the optimal SDP solution with dual multiplier $\lambda > 0$, then we can update

$$L_i := \max\left\{L_i, -\sqrt{1 + \frac{UB - LB}{\lambda}}\right\}, \quad U_i := \min\left\{U_i, \sqrt{1 + \frac{UB - LB}{\lambda}}\right\}.$$

Applying optimality-based tightening to main diagonal

(x, X) feasible for (P) $\Rightarrow\ 1 \leq x_i^2 = X_{ii} \leq \max\{L_i^2, U_i^2\}$

Lemma

Let $i \in \{1, \dots, n\}$. If the constraint $X_{ii} \geq 1$ is active at the optimal SDP solution with dual multiplier $\lambda > 0$, then we can update

$$L_i := \max\left\{L_i, -\sqrt{1 + \frac{UB - LB}{\lambda}}\right\}, \quad U_i := \min\left\{U_i, \sqrt{1 + \frac{UB - LB}{\lambda}}\right\}.$$

Lemma

Let $i \in \{1, \dots, n\}$. Assume that a constraint of type $X_{ii} \leq \gamma$ is active at the optimal SDP solution with dual multiplier $\lambda > 0$ such that $p := \gamma - \frac{UB - LB}{\lambda} \geq 1$. Then the following holds:

1. *If $L_i > -\sqrt{p}$, then we can update L_i via $L_i := \max\{L_i, \sqrt{p}\}$.*
2. *If $U_i < \sqrt{p}$, then we can update U_i via $U_i := \min\{U_i, -\sqrt{p}\}$.*

Lower bound computation

1. Find an initial good upper bound UB.
2. Compute optimality-based box constraints.
3. Solve SDP + RLT relaxation using a cutting-plane approach.

Lower bound computation

1. Find an initial good upper bound UB.
2. Compute optimality-based box constraints.
3. Solve SDP + RLT relaxation using a cutting-plane approach.

- Mosek as SDP solver
- bound tightening and primal heuristic in every iteration
- box constraints are recomputed whenever UB is updated

Lower bound computation

1. Find an initial good upper bound UB.
2. Compute optimality-based box constraints.
3. Solve SDP + RLT relaxation using a cutting-plane approach.

- Mosek as SDP solver
- bound tightening and primal heuristic in every iteration
- box constraints are recomputed whenever UB is updated

Projecting box constraints

$L_i > -1 \;\Rightarrow\; L_i := \max\{L_i, 1\}$ and $U_i < 1 \;\Rightarrow\; U_i := \min\{U_i, -1\}$

Lower bound computation

1. Find an initial good upper bound UB.
2. Compute optimality-based box constraints.
3. Solve SDP + RLT relaxation using a cutting-plane approach.

- Mosek as SDP solver
- bound tightening and primal heuristic in every iteration
- box constraints are recomputed whenever UB is updated

Projecting box constraints

$L_i > -1 \;\Rightarrow\; L_i := \max\{L_i, 1\}$ and $U_i < 1 \;\Rightarrow\; U_i := \min\{U_i, -1\}$

Binary branching

- choose a variable x_i with $L_i \leq -1$ and $U_i \geq 1$
- set $U_i := -1$ in one subproblem and set $L_i := 1$ in the other

Primal heuristic

SVM with respect to $\bar{y} \in \{-1, 1\}^n$

$$\begin{aligned} \min \quad & x^\top C x \\ \text{s. t.} \quad & \bar{y}_i \, x_i \geq 1, \;\; i = 1, \dots, n, \\ & x \in \mathbb{R}^n \end{aligned} \qquad \text{(QP)}$$

Primal heuristic

SVM with respect to $\bar{y} \in \{-1, 1\}^n$

$$\begin{aligned} \min \quad & x^\top C x \\ \text{s.t.} \quad & \bar{y}_i \, x_i \geq 1, \; i = 1, \ldots, n, \\ & x \in \mathbb{R}^n \end{aligned} \tag{QP}$$

Let $(\hat{x}, \hat{X})$ be the SDP solution.

1. Construct $\bar{y}$ with entries $\bar{y}_i = \operatorname{sign}(\hat{x}_i)$ and solve (QP).
2. Improve the solution found by applying 2-opt local search.

Computational results

- implementation in `Julia` using `JuMP`
- `Mosek` for SDPs
- optimality gap computed as $\varepsilon = \frac{\text{UB}-\text{LB}}{\text{UB}}$
- branch-and-bound is stopped when ε smaller than 0.1%
- results are averaged over three different seeds
- kernel and hyperparameters are chosen by 10-fold cross-validation

Root node relaxation for 10%, 20%, 30% labeled points

Instance	ℓ	$n - \ell$	Time Box [s]	Gap [%]	Time [s]	Iter
2moons	30	270	11.86	0.00	7.57	3.00
2moons	60	240	12.45	0.00	7.35	3.00
2moons	90	210	11.12	0.00	7.31	3.00
art150	14	136	1.30	0.04	1.44	3.00
art150	29	121	1.59	0.00	1.69	3.00
art150	44	106	1.32	0.01	1.38	3.00
connectionist	20	188	3.21	0.19	6.13	4.00
connectionist	41	167	3.09	0.16	9.84	4.67
connectionist	62	146	3.05	0.45	9.03	4.67
GunPoint	44	407	47.15	0.00	57.56	4.00
GunPoint	89	362	46.59	0.04	55.44	4.00
GunPoint	134	317	43.90	0.01	50.60	4.00
heart	27	243	6.92	0.22	10.36	4.00
heart	54	216	6.96	0.08	13.93	4.33
heart	81	189	6.37	0.15	12.05	4.33
ionosphere	34	317	19.84	0.66	19.53	3.67
ionosphere	70	281	19.67	0.01	20.73	3.33
ionosphere	104	247	17.98	0.00	27.77	4.00
PowerCons	36	324	21.80	0.04	22.79	3.67
PowerCons	72	288	19.12	0.01	26.26	4.00
PowerCons	108	252	18.87	0.01	28.53	4.00

Gurobi vs. SDP-S3VM

			Gurobi		SDP-S3VM		
Instance	ℓ	$n - \ell$	Gap [%]	Time [s]	Gap [%]	Time [s]	Solved
art100	10	90	7.37	3600	0.10	**26.11**	3
art100	20	80	3.09	2467.43	0.10	**13.28**	3
art100	30	70	3.27	2401.26	0.10	**37.48**	3
art150	14	136	8.44	3600	0.10	**61.05**	3
art150	29	121	2.72	1450.20	0.10	**1.89**	3
art150	44	106	2.52	2629.13	0.10	**2.44**	3
connectionist	20	188	16.83	3600	0.88	2587.20	1
connectionist	62	146	12.87	3600	0.10	**248.07**	3
connectionist	41	167	10.71	3600	0.10	**104.95**	3
heart	27	243	14.00	3600	0.10	**38.89**	3
heart	54	216	10.21	3600	0.10	**64.45**	3
heart	81	189	10.58	3600	0.10	**16.22**	3
2moons	30	270	6.52	3600	0.10	**16.22**	3
2moons	60	140	0.03	1023.52	0.10	**22.07**	3
2moons	90	210	0.05	**1.95**	0.10	21.50	3

- time limit of 3600 seconds

SVM vs. S3VM

Instance	ℓ	$n-\ell$	Kernel	Nodes	Time [s]	Acc. [%]	SVM [%]
ionosphere	34	317	RBF	59	529.48	**91.80**	81.70
ionosphere	34	317	linear	73	492.74	88.33	**88.96**
ionosphere	34	317	linear	3	50.05	**87.38**	84.23
ionosphere	70	281	RBF	3	107.89	**90.75**	90.04
ionosphere	70	281	RBF	7	181.61	**91.46**	85.05
ionosphere	70	281	linear	1	43.55	**88.61**	87.54
ionosphere	104	247	RBF	5	128.45	90.28	**90.69**
ionosphere	104	247	linear	37	221.45	**88.26**	86.64
ionosphere	104	247	linear	1	56.87	89.47	**90.69**
PowerCons	36	324	RBF	11	139.97	**95.06**	93.83
PowerCons	36	324	RBF	1	45.2	95.37	**96.3**
PowerCons	36	324	linear	53	534.19	**97.84**	94.44
PowerCons	72	288	RBF	11	101.41	**95.83**	94.79
PowerCons	72	288	RBF	1	30.79	96.53	**97.57**
PowerCons	72	288	linear	55	375.76	**98.61**	97.57
PowerCons	108	252	linear	11	129.53	**98.81**	**98.81**
PowerCons	108	252	linear	15	109.83	98.81	**99.21**
PowerCons	108	252	linear	17	169.85	98.41	**99.21**

Conclusion and future work

Conclusion

- S3VM models **can** be solved to **optimality**
- tools: SDP and global optimization
- S3VMs **can** be much **better** than SVMs

Future work

- first-order solver for SDPs
- parallel branch-and-bound

Conclusion and future work

Conclusion

- S3VM models **can** be solved to **optimality**
- tools: SDP and global optimization
- S3VMs **can** be much **better** than SVMs

Future work

- first-order solver for SDPs
- parallel branch-and-bound

Thank you!