August 2023

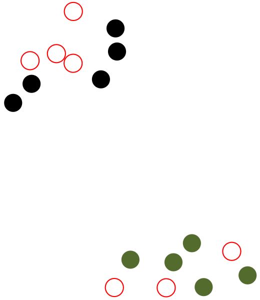
# Semidefinite Programming for Semi-Supervised Support Vector Machines

Joint work with Veronica Piccialli and Antonio M. Sudoso

# Semi-Supervised Support Vector Machines (S3VMs)

## Input

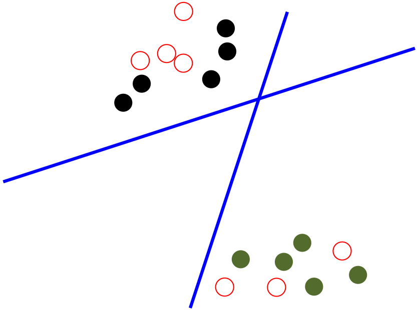
- $n$ data points $x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\},\ i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

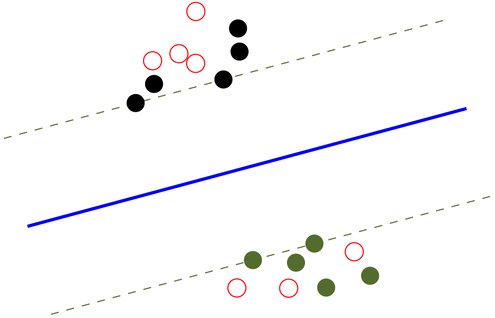
- $n$ data points $x_i \in \mathbb{R}^d, \; i = 1, \ldots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\}, \; i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

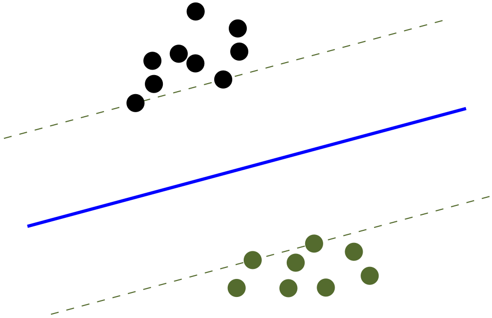
- $n$ data points $x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\},\ i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

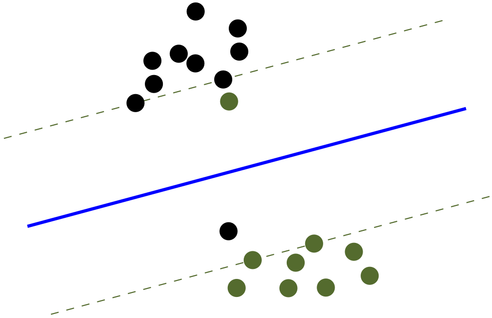
- $n$ data points $x_i \in \mathbb{R}^d, \; i = 1, \ldots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\}, \; i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

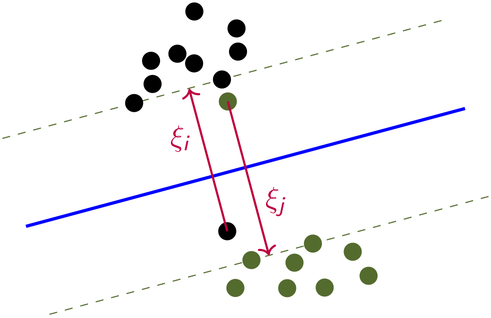
- $n$ data points $x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\},\ i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

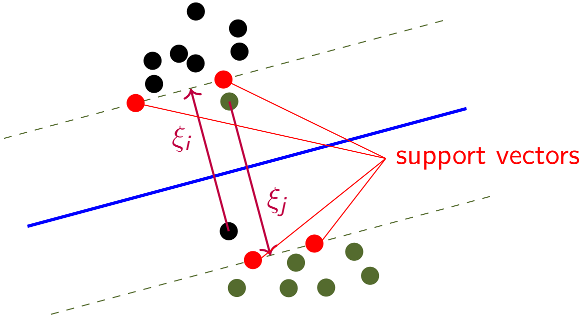
- $n$ data points $x_i \in \mathbb{R}^d,\ i = 1, \dots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\},\ i = 1, \dots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

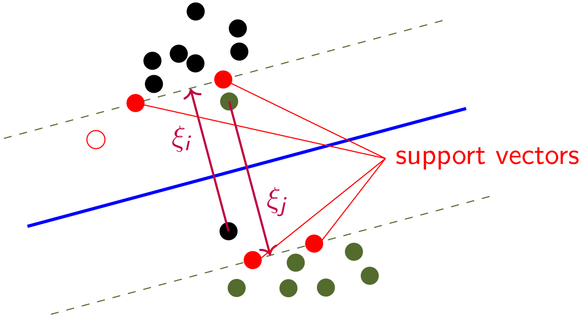
- $n$ data points $x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\},\ i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

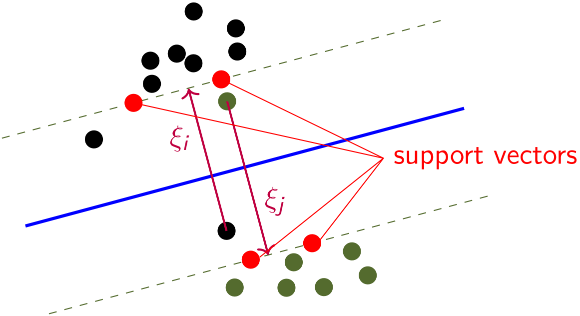
- $n$ data points $x_i \in \mathbb{R}^d,\ i = 1, \ldots, n$, centered around origin
- $\ell$ labeled points $\{(x_i, y_i)\}_{i=1}^{\ell}$ with $y_i \in \{-1, +1\},\ i = 1, \ldots, \ell$
- $n - \ell$ unlabeled points $\{x_i\}_{i=\ell+1}^{n}$



# Semi-Supervised Support Vector Machines (S3VMs)

## Input

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



# Nonconvex Quadratic Formulation of S3VMs

**Reformulation** Bai & Yan (2016)

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

- quadratic programming problem in continuous variables
- $C$ positive definite, i.e., convex objective function
- nonconvex feasible set
- bound constraints: $y_i x_i \geq 1$ means either $x_i \leq -1$ or $x_i \geq 1$

# Nonconvex Quadratic Formulation of S3VMs

**Reformulation** Bai & Yan (2016)

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

- quadratic programming problem in continuous variables
- $C$ positive definite, i.e., convex objective function
- nonconvex feasible set
- bound constraints: $y_i x_i \geq 1$ means either $x_i \leq -1$ or $x_i \geq 1$

**Overall goal:** exact approach for (P) using branch-and-cut

# Convex Relaxations

## Quadratic programming (QP) relaxation

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

# Convex Relaxations

## Quadratic programming (QP) relaxation

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

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

$$
\begin{aligned}
\min \quad & \langle C, X \rangle \\
\text{s. t.} \quad & y_i x_i \geq 1, \;\; i = 1, \dots, \ell \\
& X_{ii} \geq 1, \;\; i = \ell + 1, \dots, n \\
& \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0
\end{aligned}
\tag{SDP}
$$

# Convex Relaxations

## Quadratic programming (QP) relaxation

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

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

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{s. t.} \quad & y_i x_i \geq 1, \; i = 1, \dots, \ell \\ & X_{ii} \geq 1, \; i = \ell + 1, \dots, n \\ & \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0 \end{aligned} \tag{SDP}$$

**Two goals:**

- stronger SDP relaxation
- efficient algorithm to solve SDP relaxation

# Global Optimization Problem

- compute box constraints $L_i \leq x_i \leq U_i,\ i = 1, \ldots, n$

**Textbook-like form with box constraints**

$$\begin{aligned} \min \quad & x^\top C x \\ \text{s.t.} \quad & L_i \leq x_i \leq U_i,\ i = 1, \ldots, n \\ & x_i^2 \geq 1,\ i = \ell + 1, \ldots, n \\ & x \in \mathbb{R}^n \end{aligned}$$

# Global Optimization Problem

- compute box constraints $L_i \leq x_i \leq U_i, \; i = 1, \ldots, n$

**Textbook-like form with box constraints**

$$\begin{aligned} \min \quad & x^\top C x \\ \text{s. t.} \quad & L_i \leq x_i \leq U_i, \; i = 1, \ldots, n \\ & x_i^2 \geq 1, \; i = \ell + 1, \ldots, n \\ & x \in \mathbb{R}^n \end{aligned}$$

- add RLT cuts (Sherali & Adams, 1998) to SDP relaxation:

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

# Global Optimization Problem

- compute box constraints $L_i \leq x_i \leq U_i, \; i = 1, \dots, n$

**Textbook-like form with box constraints**

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

- add RLT cuts (Sherali & Adams, 1998) to SDP relaxation:

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

- marginals-based bound tightening (Ryoo & Sahinidis, 1995)
- projecting box constraints:
  - $L_i > -1 \;\Rightarrow\; L_i := \max\{L_i, 1\}$
  - $U_i < 1 \;\Rightarrow\; U_i := \min\{U_i, -1\}$

# A New Mixing Method for S3VM (inspired by Wang, Chang, Kolter, 2018)

**Change of variables: Burer-Monteiro factorization**

$$\begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} = V^\top V \text{ with } V = (v_0 | v_1 | \dots | v_n) \in \mathbb{R}^{k \times n}$$

# A New Mixing Method for S3VM (inspired by Wang, Chang, Kolter, 2018)

## Change of variables: Burer-Monteiro factorization

$$\begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} = V^\top V \text{ with } V = (v_0|v_1|\dots|v_n) \in \mathbb{R}^{k\times n}$$

## Nonconvex reformulation

For some $k \leq n$, (SDP) is equivalent to

$$\begin{aligned} \min \quad & \langle \bar{C}, V^\top V \rangle \\ \text{s.t.} \quad & y_i v_0^\top v_i \geq 1, \; i = 1, \dots, \ell, \\ & \|v_i\|^2 \geq 1, \; i = \ell+1, \dots, n, \\ & \|v_0\|^2 = 1, \\ & V = (v_0|v_1|\dots|v_n) \in \mathbb{R}^{k\times n}, \end{aligned} \tag{$*$}$$

$$\bar{C} = \begin{pmatrix} 0 & 0 \\ 0 & C \end{pmatrix}.$$

# Coordinate Descent: Column Updates

## Updating column $i \neq 0$

Let $g = 2\sum_{j \neq i}^{n} C_{ij} v_j$. Fixing all other columns, $(*)$ reduces to

$$\begin{cases} \begin{array}{ll} \min & C_{ii}\|v_i\|^2 + g^\top v_i \\ \text{s.t.} & y_i v_0^\top v_i \geq 1, \end{array}, & \text{if } i \in \{1, \ldots, \ell\}, \\ \\ \begin{array}{ll} \min & C_{ii}\|v_i\|^2 + g^\top v_i \\ \text{s.t.} & \|v_i\|^2 \geq 1, \end{array}, & \text{if } i \in \{\ell + 1, \ldots, n\}. \end{cases}$$

# Coordinate Descent: Column Updates

## Updating column $i \neq 0$

Let $g = 2\sum_{j \neq i}^{n} C_{ij} v_j$. Fixing all other columns, $(*)$ reduces to

$$\begin{cases} \begin{array}{ll} \min & C_{ii}\|v_i\|^2 + g^\top v_i \\ \text{s. t.} & y_i v_0^\top v_i \geq 1, \end{array} & \text{if } i \in \{1, \dots, \ell\}, \\ \\ \begin{array}{ll} \min & C_{ii}\|v_i\|^2 + g^\top v_i \\ \text{s. t.} & \|v_i\|^2 \geq 1, \end{array} & \text{if } i \in \{\ell + 1, \dots, n\}. \end{cases}$$

## Primal-dual solution

There is a closed-form primal-dual solution to both subproblems.

# Simple Algorithm

1. Choose $k \leq \lceil \sqrt{2n} \rceil$.
2. Initialize $v_0, v_1, \ldots, v_n$ randomly on the unit sphere.
3. Repeat until done: update $v_1, \ldots, v_n$.

# Simple Algorithm

1. Choose $k \leq \lceil \sqrt{2n} \rceil$.
2. Initialize $v_0, v_1, \ldots, v_n$ randomly on the unit sphere.
3. Repeat until done: update $v_1, \ldots, v_n$.

- robust and very fast
- access to approximate dual solution (even if $k$ too small)

# Simple Algorithm

1. Choose $k \leq \lceil \sqrt{2n} \rceil$.
2. Initialize $v_0, v_1, \ldots, v_n$ randomly on the unit sphere.
3. Repeat until done: update $v_1, \ldots, v_n$.

- robust and very fast
- access to approximate dual solution (even if $k$ too small)

**Future work:**

- further strengthen relaxation (disjunctive cuts?)
- convergence proof?
- partial Lagrangian approach to dualize cuts

# Simple Algorithm

1. Choose $k \leq \lceil \sqrt{2n} \rceil$.
2. Initialize $v_0, v_1, \ldots, v_n$ randomly on the unit sphere.
3. Repeat until done: update $v_1, \ldots, v_n$.

- robust and very fast
- access to approximate dual solution (even if $k$ too small)

**Future work:**

- further strengthen relaxation (disjunctive cuts?)
- convergence proof?
- partial Lagrangian approach to dualize cuts

# Thank you!