Kernel Machines

Alain Celisse

SAMM

Paris 1-Panthéon Sorbonne University

alain.celisse@univ-paris1.fr

Lecture 1: Support vector classifiers and Kernel machines

—

Master 2 Data Science – Centrale Lille, Lille University
Fall 2022

Outline of the lectures

Successive topics of the coming lectures:

1. Introduction to Kernel methods
2. Support vector classifiers and Kernel methods (Today!)
3. Extending classical strategies to high dimension
   - KRR/LS-SVMs
   - KPCA
4. Duality gap and KKT conditions
5. Designing reproducing kernels
6. Maximum Mean Discrepancy (MMD)
7. Change-point detection, KCP

Outline of the lecture

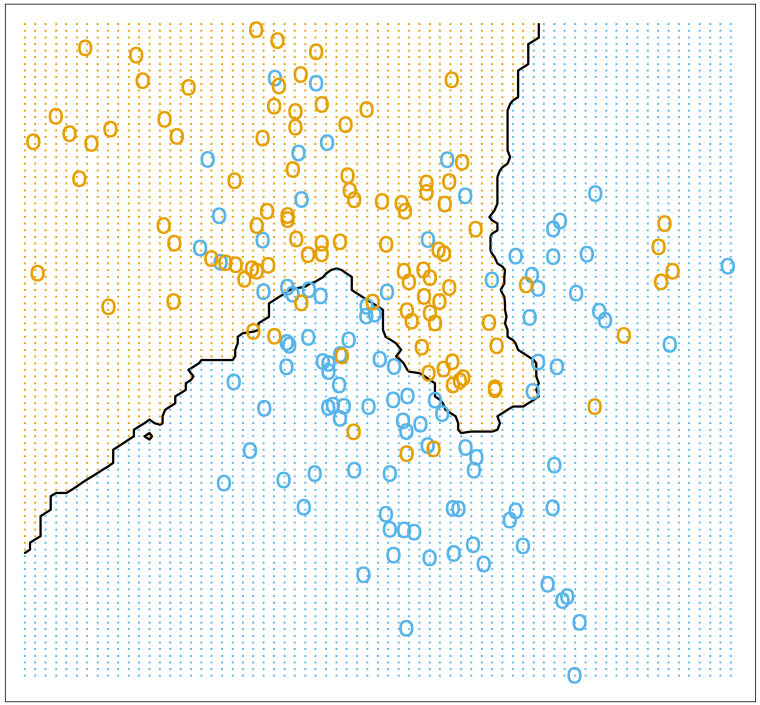
- Optimal Separating Hyperplanes
- Support Vectors
- Kernel machines and SVM
- Towards regression

Optimal Separating Hyperplanes

Binary classification

Framework

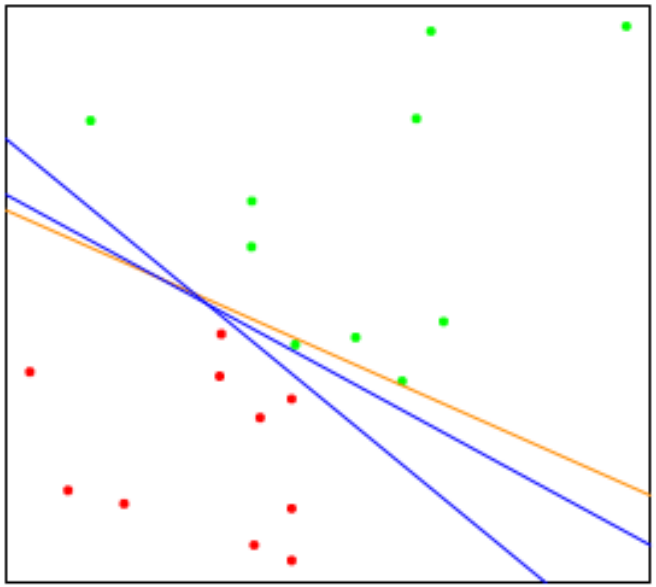
- $(X_1, Y_1), \ldots, (X_n, Y_n) \overset{iid}{\sim} P$: unknown
- $X_i \in \mathcal{X} \subset \mathbb{R}^d$
- $Y_i \in \{-1, 1\} = \mathcal{Y}$



Goal and context

- Design a *data-driven* classification rule
- The classification rule achieves the smallest possible "misclassification rate"
- No parametric model for P (nonparametric)

Linearly separable classes



Definition (Linearly separable classes)

Two sets of points are *linearly separable* if there exists a

hyperplane $B = \left\{ x \in \mathbb{R}^d \mid \underbrace{\beta_0 + \beta^\top x}_{=f(x)} = 0 \right\}$ such that, for all i,

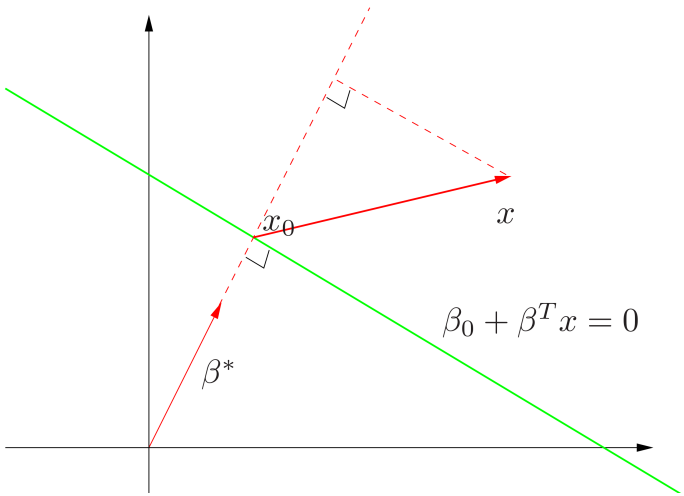
$$Y_i = \text{sign}(\beta_0 + \beta^\top X_i) \quad \Leftrightarrow \quad \begin{array}{l} Y_i = 1, \quad \beta_0 + \beta^\top X_i > 0 \\ Y_i = -1, \quad \underbrace{\beta_0 + \beta^\top X_i}_{=f(X_i)} < 0 \end{array}$$

Hyperplane and orthogonal projection

- With linearly separable classes, several separating hyperplanes do exist
  $\rightarrow$ Need for a criterion to choose one of them

Orthogonal projection

- $x_0 \in B \quad \Leftrightarrow \quad \beta_0 + \langle \beta, x_0 \rangle = 0$
- $\beta^\star = \beta / \|\beta\|$: normal vector to B
- x: any vector



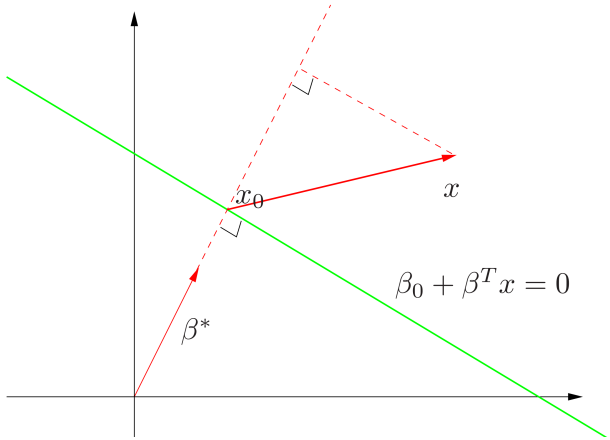
1. $p^\perp(x)$: orthogonal projection of x onto B
   $\Rightarrow \beta_0 + \langle \beta, p^\perp(x) \rangle = f(p^\perp(x)) = 0 \qquad (p^\perp(x) \in B)$
2. $x - p^\perp(x) = \alpha \beta^\star$ for some $\alpha \in \mathbb{R}$

$\Rightarrow \alpha = \langle x - x_0, \beta^\star \rangle$ and $x - p^\perp(x) = \langle x - x_0, \beta^\star \rangle \, \beta^\star$

Distance to a hyperplane

Signed distance from x to B

- $x - p^{\perp}(x) = \langle x - x_0, \beta^\star \rangle \, \beta^\star$



- **Signed distance of x to B:**

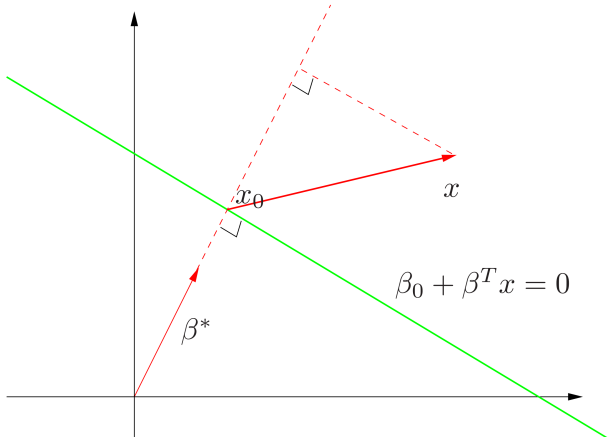
$$\left\langle x - p^{\perp}(x), \beta^\star \right\rangle = \langle x - x_0, \beta^\star \rangle$$

$$= \frac{\langle \beta, x \rangle + \beta_0}{\|\beta\|} = \frac{f(x)}{\|\beta\|} = \frac{f(x)}{\|f'(x)\|}$$

Distance to a hyperplane

Signed distance from x to B

- $x - p^{\perp}(x) = \langle x - x_0, \beta^\star \rangle \, \beta^\star$



- **Signed distance of x to B:**

$$\left\langle x - p^{\perp}(x), \beta^\star \right\rangle = \langle x - x_0, \beta^\star \rangle$$

$$= \frac{\langle \beta, x \rangle + \beta_0}{\|\beta\|} = \frac{f(x)}{\|\beta\|} = \frac{f(x)}{\|f'(x)\|}$$

Remarks:

- Positive distance iff B predicts label "1" at x ($f(x) > 0$)
- Normalizing by $\|\beta\|$ mandatory because $\beta^\star$ is unit norm

Cumulated distance to the seprating hyperplane

Definition (Misclassified points)

Any observation x_i is said *misclassified* if

$$y_i f(x_i) < 0$$

$$(f(x) = \langle \beta, x \rangle + \beta_0)$$

Cumulated distance of well-classified points to B

$$dist\left(\widetilde{\beta}\right) = \sum_{i=1}^{n} \left(y_i \frac{f(x_i)}{\|f'(x_i)\|} \right)_+ \quad \propto \quad \sum_{i=1}^{n} (y_i f(x_i))_+$$

Perceptron learning algorithm

Perceptron algorithm

$(\widetilde{\beta} = (\beta, \beta_0))$

- Perceptron aims at finding $\widetilde{\beta} \in \mathbb{R}^d$ that minimizes $\sum_{i=1}^n (y_i f(x_i))_+$
- Solves the optimization problem:

$$\min \sum_{i=1}^{n} \left(y_i f_{\widetilde{\beta}}(x_i) \right)_+ \qquad s.t. \quad \widetilde{\beta} \in \mathbb{R}^d$$

Perceptron learning algorithm

Perceptron algorithm $(\widetilde{\beta} = (\beta, \beta_0))$

- Perceptron aims at finding $\widetilde{\beta} \in \mathbb{R}^d$ that minimizes $\sum_{i=1}^{n} (y_i f(x_i))_+$
- Solves the optimization problem:

$$\min \sum_{i=1}^{n} \left(y_i f_{\widetilde{\beta}}(x_i) \right)_+ \qquad s.t. \quad \widetilde{\beta} \in \mathbb{R}^d$$

SGD iterations:

- Perceptron solution computed by SGD:

$$\left(\widetilde{\beta}^{t+1}\right)^\top = \left(\widetilde{\beta}^{t}\right)^\top + \rho \left(x_i y_i, y_i\right)^\top$$

for all missclassified x_is and $\rho > 0$

Perceptron learning algorithm

Perceptron algorithm $(\widetilde{\beta} = (\beta, \beta_0))$

- Perceptron aims at finding $\widetilde{\beta} \in \mathbb{R}^d$ that minimizes $\sum_{i=1}^n (y_i f(x_i))_+$
- Solves the optimization problem:

$$\min \sum_{i=1}^{n} \left(y_i f_{\widetilde{\beta}}(x_i) \right)_+ \qquad s.t. \quad \widetilde{\beta} \in \mathbb{R}^d$$

SGD iterations:

- Perceptron solution computed by SGD:

$$\left(\widetilde{\beta}^{t+1}\right)^\top = \left(\widetilde{\beta}^{t}\right)^\top + \rho \left(x_i y_i, y_i\right)^\top$$

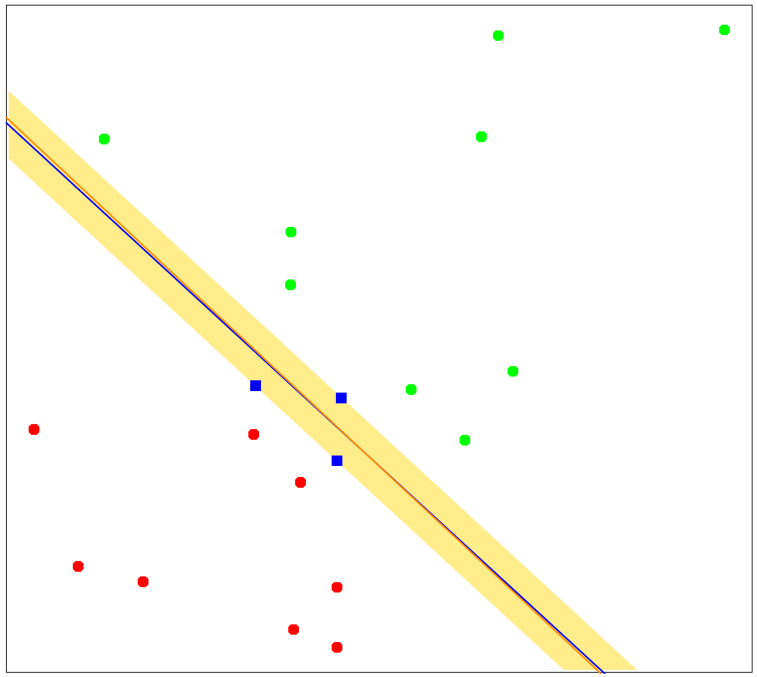
  for all missclassified x_is and $\rho > 0$
- **Limitations:**
  - Not scale-invariant (solution depends on $\|\beta\|$)
  - Only converges with linearly separable classes

Optimal Separating Hyperplane and Margin

Optimal hyperplane

- With linearly separable classes, no unique hyperplane
- **Optimal separating hyperplane:**

  "The one which departs the most from each class"



Margin

Half the width of the yellow band

Margin: Formal definition

Well-classified points and Margin $(f_{\widetilde{\beta}}(x) = \langle \beta, x \rangle + \beta_0 = 0)$

- Sets of well-classified points:

$$C_1 = \{x_i \mid y_i = 1, y_i f(x_i) > 0\} \quad C_{-1} = \{x_i \mid y_i = -1, y_i f(x_i) > 0\}$$

Margin: Formal definition

Well-classified points and Margin $(f_{\widetilde{\beta}}(x) = \langle \beta, x \rangle + \beta_0 = 0)$

- Sets of well-classified points:

$$C_1 = \{x_i \mid y_i = 1, y_i f(x_i) > 0\} \quad C_{-1} = \{x_i \mid y_i = -1, y_i f(x_i) > 0\}$$

- **Distance from x_i to B:**

$$dist(x_i, B) = y_i f_{\widetilde{\beta}}(x_i) / \left\| f'_{\widetilde{\beta}}(x_i) \right\|$$

Margin: Formal definition

Well-classified points and Margin $(f_{\widetilde{\beta}}(x) = \langle \beta, x \rangle + \beta_0 = 0)$

- Sets of well-classified points:

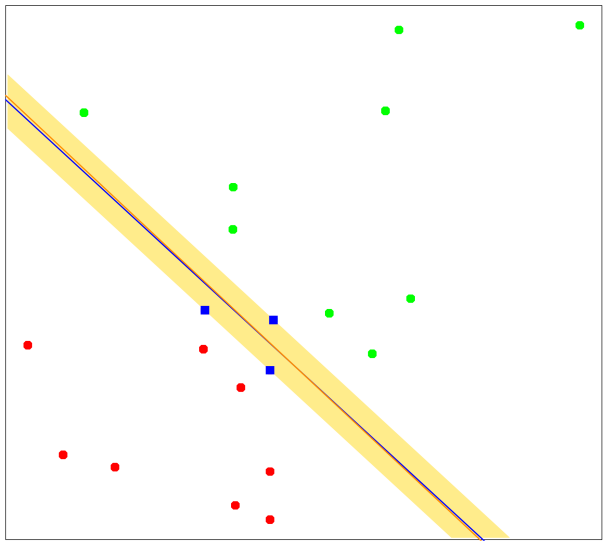
$$C_1 = \{x_i \mid y_i = 1, y_i f(x_i) > 0\} \quad C_{-1} = \{x_i \mid y_i = -1, y_i f(x_i) > 0\}$$

- **Distance from x_i to B:**

$$dist(x_i, B) = y_i f_{\widetilde{\beta}}(x_i) / \left\| f'_{\widetilde{\beta}}(x_i) \right\|$$

- **Distance from a (finite) set C_ℓ to B:** $(\ell \in \{-1, 1\})$

$$dist(C_\ell, B) = \min \{dist(x_i, B) \mid x_i \in C_\ell\}$$



Margin: Formal definition

Well-classified points and Margin $(f_{\widetilde{\beta}}(x) = \langle \beta, x \rangle + \beta_0 = 0)$

- Sets of well-classified points:

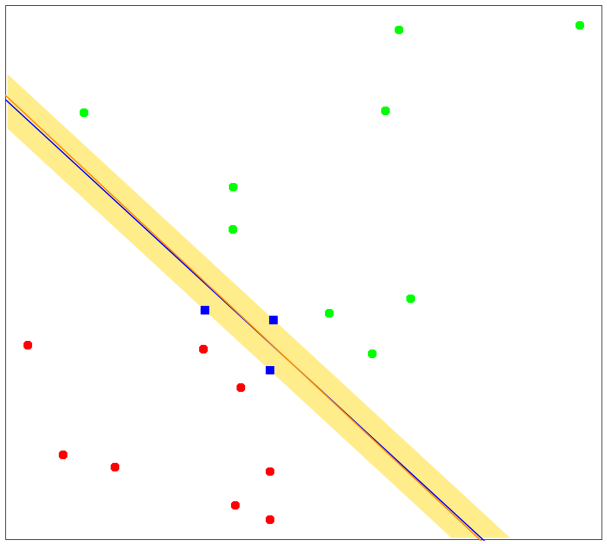
$$C_1 = \{x_i \mid y_i = 1, y_i f(x_i) > 0\} \quad C_{-1} = \{x_i \mid y_i = -1, y_i f(x_i) > 0\}$$

- **Distance from x_i to B:**

$$dist(x_i, B) = y_i f_{\widetilde{\beta}}(x_i) / \left\| f'_{\widetilde{\beta}}(x_i) \right\|$$

- **Distance from a (finite) set C_ℓ to B:** $(\ell \in \{-1, 1\})$

$$dist(C_\ell, B) = \min \{dist(x_i, B) \mid x_i \in C_\ell\}$$



Margin: Formal definition

Well-classified points and Margin $(f_{\widetilde{\beta}}(x) = \langle \beta, x \rangle + \beta_0 = 0)$

- Sets of well-classified points:

$$C_1 = \{x_i \mid y_i = 1, y_i f(x_i) > 0\} \quad C_{-1} = \{x_i \mid y_i = -1, y_i f(x_i) > 0\}$$

- **Distance from x_i to B:**

$$dist(x_i, B) = y_i f_{\widetilde{\beta}}(x_i) / \left\| f'_{\widetilde{\beta}}(x_i) \right\|$$

- **Distance from a (finite) set C_ℓ to B:** $(\ell \in \{-1, 1\})$

$$dist(C_\ell, B) = \min \{dist(x_i, B) \mid x_i \in C_\ell\}$$

Definition (Margin)

The margin $M > 0$ is given by

$$M = M(B) = \min [\, dist(C_1, B), dist(C_{-1}, B) \,]$$

Remark:

Margin: function of the hyperplane

Optimization problem and Margin

$$M = M(B) = \min\left(dist(C_1, B), dist(C_{-1}, B)\right)$$

Maximizing the margin

- For all well-classified points x_i, require:

$$dist(x_i, B) = y_i\left(\langle \beta, x_i \rangle + \beta_0\right) / \|\beta\| \geq M > 0$$

- M is a "confidence level" on well-classified points

Optimization problem and Margin

$$M = M(B) = \min\left(dist(C_1, B), dist(C_{-1}, B)\right)$$

Maximizing the margin

- For all well-classified points x_i, require:

$$dist(x_i, B) = y_i\left(\langle \beta, x_i \rangle + \beta_0\right) / \|\beta\| \geq M > 0$$

- M is a "confidence level" on well-classified points

Optimization problem

Optimal hyperplane given by the unique (β, β_0) such that M is maximum

Optimization problem and Margin

$$M = M(B) = \min\left(dist(C_1, B), dist(C_{-1}, B)\right)$$

Maximizing the margin

- For all well-classified points x_i, require:

$$dist(x_i, B) = y_i\left(\langle \beta, x_i \rangle + \beta_0\right) / \|\beta\| \geq M > 0$$

- M is a "confidence level" on well-classified points

Optimization problem

Optimal hyperplane given by the unique (β, β_0) such that M is maximum

Solve ($M = M(\beta, \beta_0)$):

$$\max_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} M, \quad \text{s.t.}$$

$$y_i\left(\langle \beta, x_i \rangle + \beta_0\right) \geq M \|\beta\|, \quad \text{for } i = 1, \ldots, n$$

Optimization problem and Margin

$$M = M(B) = \min\left(dist(C_1, B), dist(C_{-1}, B)\right)$$

Maximizing the margin

- For all well-classified points x_i, require:

$$dist(x_i, B) = y_i\left(\langle \beta, x_i \rangle + \beta_0\right) / \|\beta\| \geq M > 0$$

- M is a "confidence level" on well-classified points

Optimization problem

Optimal hyperplane given by the unique (β, β_0) such that M is maximum

Solve ($M = M(\beta, \beta_0)$):

$$\max_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} M, \quad \text{s.t.}$$

$$y_i\left(\langle \beta, x_i \rangle + \beta_0\right) \geq M \|\beta\|, \quad \text{for } i = 1, \ldots, n$$

- Not scale invariant: β and $2 \cdot \beta$ lead to the same solution for M (Hint: $\left(\langle \beta, x_i \rangle + \beta_0\right) / \|\beta\| = \langle x - x_0, \beta^\star \rangle$)

Optimization problem reformulation

- Add a constraint on M:

$$M \cdot \|\beta\| = 1$$

Recasting the optimization problem

$$\max_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} M, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq M \|\beta\|, \quad \text{for } i = 1, \ldots, n$$

$$\Leftrightarrow \quad \min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Optimization problem reformulation

- Add a constraint on M:

$$M \cdot \|\beta\| = 1$$

Recasting the optimization problem

$$\max_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} M, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq M \|\beta\|, \quad \text{for } i = 1, \ldots, n$$

$$\Leftrightarrow \quad \min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Remark:

Square the norm without changing the minimum location (differentiability)

Support Vectors

Lagrange multipliers formulation

Constrained optimization problem

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Lagrange multipliers formulation

Constrained optimization problem

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Lagrange multipliers $(\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}_+^n)$

First step: Primal formulation

1. Minimize w.r.t β, β_0:

$$\mathcal{L}_{P,\alpha}(\beta, \beta_0) = \frac{\|\beta\|^2}{2} - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) - 1 \right)$$

Lagrange multipliers formulation

Constrained optimization problem

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Lagrange multipliers $(\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n_+)$

First step: Primal formulation

1. Minimize w.r.t β, β_0:

$$\mathcal{L}_{P,\alpha}(\beta, \beta_0) = \frac{\|\beta\|^2}{2} - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) - 1 \right)$$

2. Solution (stationary points): For $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n_+$

$$\mathcal{L}_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

Lagrange multipliers formulation (Cont'd)

Lagrange multipliers

Second step: Dual formulation

1. Maximize:

$$\mathcal{L}_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

such that: $\alpha_i \geq 0, \quad \sum_{j=1}^{n} \alpha_j y_j = 0$

Lagrange multipliers formulation (Cont'd)

Lagrange multipliers

Second step: Dual formulation

1. Maximize:

$$\mathcal{L}_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

such that: $\alpha_i \geq 0, \quad \sum_{j=1}^{n} \alpha_j y_j = 0$

2. Solution (KKT conditions):

Solution to this problem: $(\widehat{\alpha}_1, \ldots, \widehat{\alpha}_n)$

- If $\widehat{\alpha}_i > 0$: $y_i(\left\langle x_i, \widehat{\beta} \right\rangle + \widehat{\beta}_0) = 1$

  $\Leftrightarrow$ x_i on the boundary of the slab

Lagrange multipliers formulation (Cont'd)

Lagrange multipliers

Second step: Dual formulation

1. Maximize:

$$\mathcal{L}_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{1 \leq i,j \leq n} \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

such that: $\alpha_i \geq 0, \quad \sum_{j=1}^{n} \alpha_j y_j = 0$

2. Solution (KKT conditions):

Solution to this problem: $(\widehat{\alpha}_1, \ldots, \widehat{\alpha}_n)$

- If $\widehat{\alpha}_i > 0$: $y_i(\left\langle x_i, \widehat{\beta} \right\rangle + \widehat{\beta}_0) = 1$
  $\Leftrightarrow$ x_i on the boundary of the slab
- If $\widehat{\alpha}_i = 0$: $y_i(\left\langle x_i, \widehat{\beta} \right\rangle + \widehat{\beta}_0) > 1$
  $\Leftrightarrow$ x_i out of the slab

Support Vector classifier

- If $\widehat{\alpha}_i > 0$: $y_i(\left\langle x_i, \widehat{\beta} \right\rangle + \widehat{\beta}_0) = 1$

Definition (Support vectors)

The points x_is such that $\widehat{\alpha}_i > 0$ are:

- called *support vectors*. $(S = \{1 \leq i \leq n \mid \widehat{\alpha}_i > 0\})$

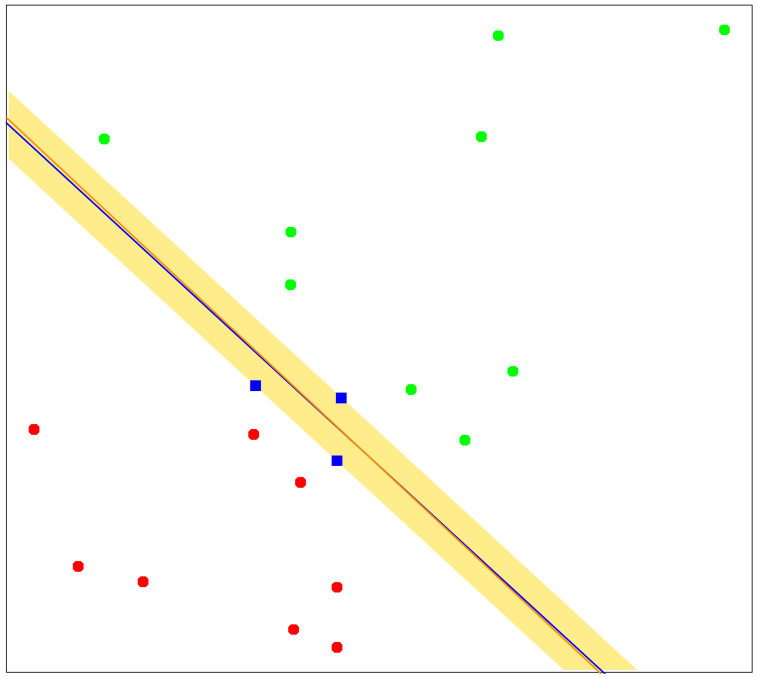
Support Vector classifier

- If $\widehat{\alpha}_i > 0$: $\quad y_i(\left\langle x_i, \widehat{\beta} \right\rangle + \widehat{\beta}_0) = 1$

Definition (Support vectors)

The points x_is such that $\widehat{\alpha}_i > 0$ are:

- called *support vectors*. $\quad (S = \{1 \leq i \leq n \mid \widehat{\alpha}_i > 0\})$
- located on the boundary of the slab that is, their distance to the hyperplane is equal to the margin.



Support vector classifier

Support Vector Classifier

From previous equality conditions:

- $\widehat{\beta} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i x_i = \sum_{i \in S} \widehat{\alpha}_i y_i x_i$

Support vector classifier

Support Vector Classifier

From previous equality conditions:

- $\widehat{\beta} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i x_i = \sum_{i \in S} \widehat{\alpha}_i y_i x_i$
- $\widehat{\beta}_0 = y_i - \left\langle x_i, \sum_{j=1}^{n} \widehat{\alpha}_j y_j x_j \right\rangle$, for all $i \in S$

Support vector classifier

Support Vector Classifier

From previous equality conditions:

- $\widehat{\beta} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i x_i = \sum_{i \in S} \widehat{\alpha}_i y_i x_i$
- $\widehat{\beta}_0 = y_i - \left\langle x_i, \sum_{j=1}^{n} \widehat{\alpha}_j y_j x_j \right\rangle$, for all $i \in S$
- Classifier: $\widehat{f}(x) = \text{sign}\left(\left\langle \widehat{\beta}, x \right\rangle + \widehat{\beta}_0\right)$

Support vector classifier

Support Vector Classifier

From previous equality conditions:

- $\widehat{\beta} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i x_i = \sum_{i \in S} \widehat{\alpha}_i y_i x_i$
- $\widehat{\beta}_0 = y_i - \left\langle x_i, \sum_{j=1}^{n} \widehat{\alpha}_j y_j x_j \right\rangle$, for all $i \in S$
- Classifier: $\widehat{f}(x) = \text{sign}\left(\left\langle \widehat{\beta}, x \right\rangle + \widehat{\beta}_0 \right)$

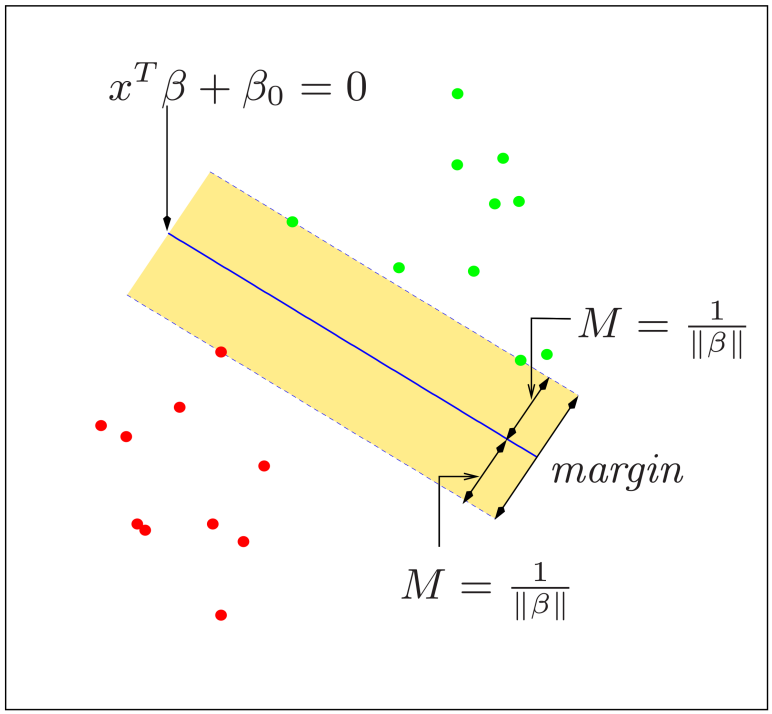
Proof.

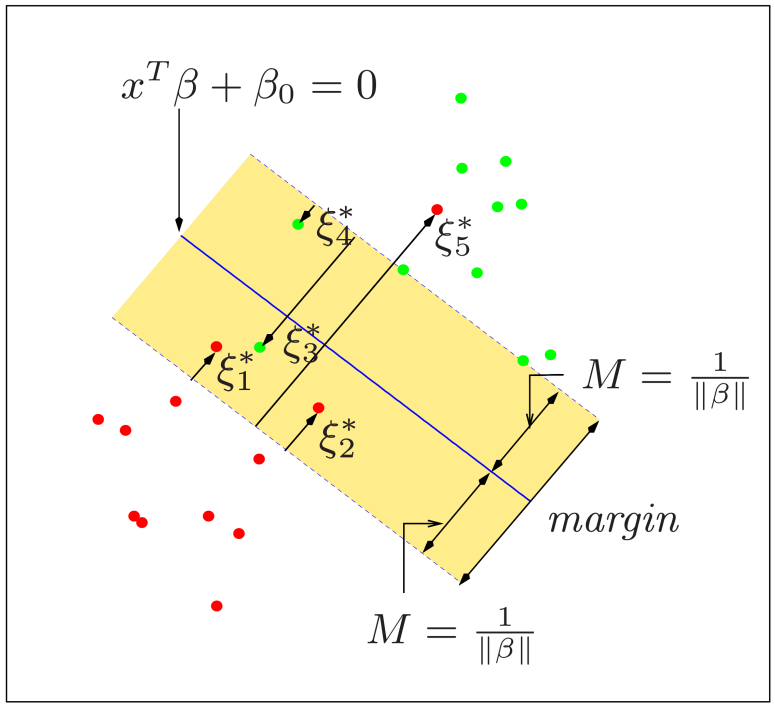
$$y_i(\left\langle x_i, \widehat{\beta} \right\rangle + \widehat{\beta}_0) = 1 \quad \Leftrightarrow \quad y_i \widehat{\beta}_0 = 1 - y_i \left\langle x_i, \widehat{\beta} \right\rangle$$

$$\Leftrightarrow \quad \widehat{\beta}_0 = y_i(1 - y_i \left\langle x_i, \sum_{j=1}^{n} \widehat{\alpha}_j y_j x_j \right\rangle)$$

□

From non-overlap. to overlap. classes





Limitation: The overlapping case

- The previous procedure addresses the non-overlapping case
- With overlapping classes, such a linear classification rule commits mistakes
- Requires to modify the optimality criterion

Misclassifications and Hinge loss

Former formulation

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Accounting for misclassifications

- x_i misclassified means:

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) < 1 \quad \Leftrightarrow \quad 0 < 1 - y_i \left(\langle \beta, x_i \rangle + \beta_0 \right)$$

Misclassifications and Hinge loss

Former formulation

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{such that}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Accounting for misclassifications

- x_i misclassified means:

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) < 1 \quad \Leftrightarrow \quad 0 < 1 - y_i \left(\langle \beta, x_i \rangle + \beta_0 \right)$$

- **Minimizing misclassifications** amounts to minimize

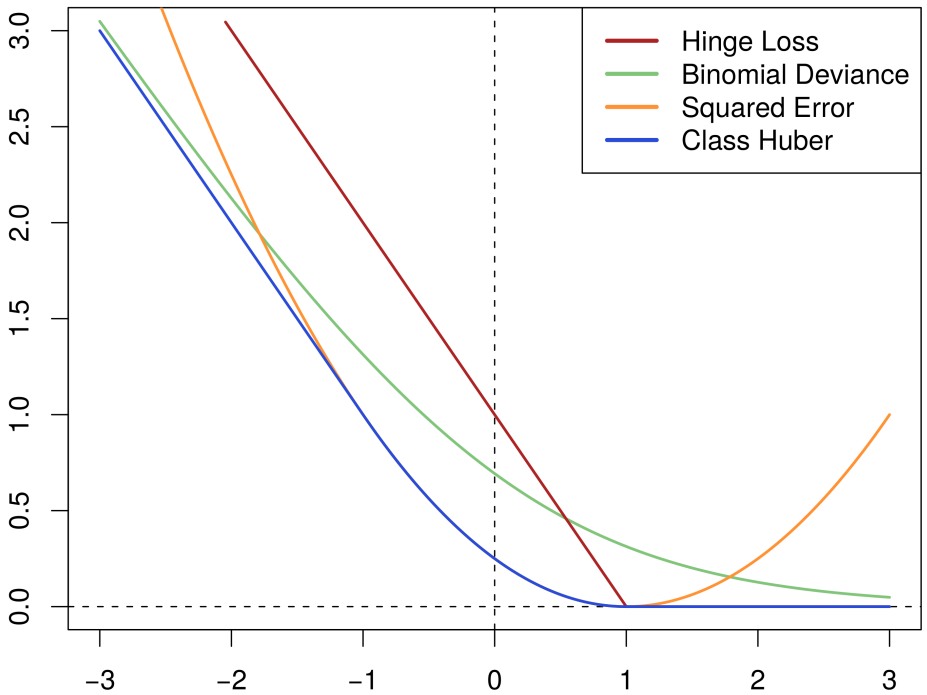
$$\sum_{i=1}^{n} \left(1 - y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \right)_+ = \sum_{i=1}^{n} h(y_i, f(x_i))$$

$\longrightarrow$ Only misclassifications come into play!

Hinge loss

Definition (Hinge loss)

$$h(y, f(x)) = (1 - y \cdot f(x))_+ = \max(0, 1 - y \cdot f(x))$$



$$x \mapsto (1 - x)_+$$

Hinge-loss and optimal classifier

Hinge-loss

$$h(y, f(x)) = (1 - y \cdot f(x))_+$$

Hinge-loss and optimal classifier

Hinge-loss

$$h(y, f(x)) = (1 - y \cdot f(x))_+$$

Prediction error (Misclassification rate) for the Hinge-loss

$$PE(f) = \mathbb{E}' \left[h(Y', f(X')) \right]$$

Hinge-loss and optimal classifier

Hinge-loss

$$h(y, f(x)) = (1 - y \cdot f(x))_+$$

Prediction error (Misclassification rate) for the Hinge-loss

$$PE(f) = \mathbb{E}' \left[h(Y', f(X')) \right]$$

Optimal function minimizer

$$f^\star = Arg \min_{f \in \mathcal{M}} \mathbb{E}' \left[h(Y', f(X')) \right]$$

yields

Hinge-loss and optimal classifier

Hinge-loss

$$h(y, f(x)) = (1 - y \cdot f(x))_+$$

Prediction error (Misclassification rate) for the Hinge-loss

$$PE(f) = \mathbb{E}' \left[h(Y', f(X')) \right]$$

Optimal function minimizer

$$f^\star = Arg \min_{f \in \mathcal{M}} \mathbb{E}' \left[h(Y', f(X')) \right]$$

yields

$$\begin{aligned} f^\star(x) &= \text{sign} \left[\frac{2\mathbb{P}(Y = +1 \mid X = x) - 1}{2} \right] \\ &= \text{sign} \left[2\mathbb{P}(Y = +1 \mid X = x) - 1 \right] \\ &= \text{sign} \left(\mathbb{E} \left[Y \mid X = x \right] \right) \end{aligned}$$

Remark:

$\longrightarrow$ Justifies the classifier expression: $\text{sign}(\langle \beta, x \rangle + \beta_0)$

Optimization problem and regularization

Past formulation

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Optimization problem and regularization

Past formulation

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$

$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Relaxation

Solve for $r \geq 1$

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$

$$\sum_{i=1}^{n} h(y_i, f_{\beta, \beta_0}(x_i)) \leq r$$

Optimization problem and regularization

Past formulation

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$
$$y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) \geq 1, \quad \text{for } i = 1, \ldots, n$$

Relaxation

Solve for $r \geq 1$

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$
$$\sum_{i=1}^{n} h(y_i, f_{\beta, \beta_0}(x_i)) \leq r$$

Remarks:

- Tackles misclassifications
- Addresses overlapping classes

Optimization problem and regularization

Relaxation

Solve for $r \geq 1$

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$

$$\sum_{i=1}^{n} h(y_i, f_{\beta,\beta_0}(x_i)) \leq r$$

Optimization problem and regularization

Relaxation

Solve for $r \geq 1$

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$

$$\sum_{i=1}^{n} h(y_i, f_{\beta,\beta_0}(x_i)) \leq r$$

Regularized formulation

Solve, for all $\lambda > 0$:

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, f_{\beta,\beta_0}(x_i)) + \lambda \|\beta\|^2 \right\}$$

where $h(y, f(x)) = (1 - yf(x))_+$, $f_{\beta,\beta_0}(x) = \langle \beta, x \rangle + \beta_0$

Optimization problem and regularization

Relaxation

Solve for $r \geq 1$

$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \|\beta\|^2 / 2, \quad \text{s.t.}$$

$$\sum_{i=1}^{n} h(y_i, f_{\beta,\beta_0}(x_i)) \leq r$$

Regularized formulation

Solve, for all $\lambda > 0$:

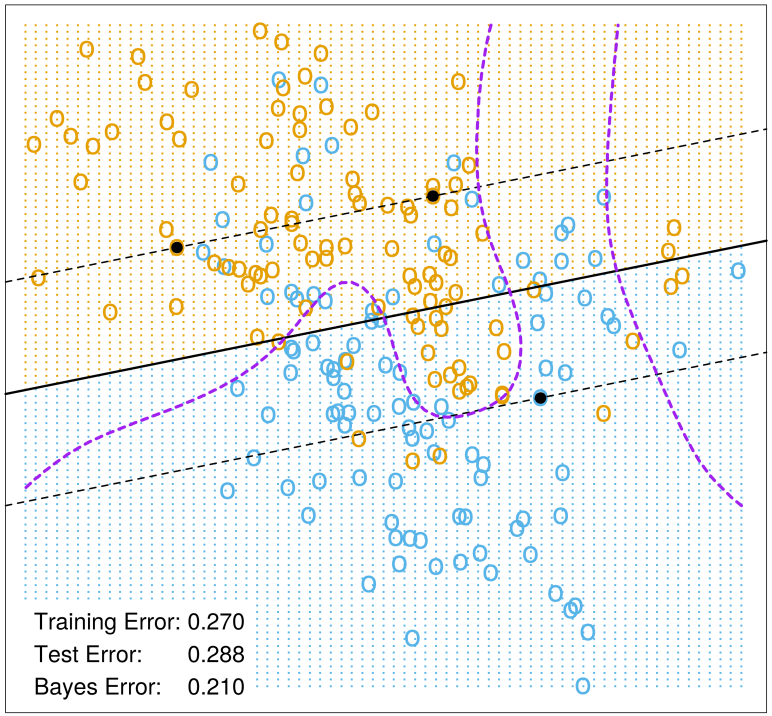
$$\min_{\beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, f_{\beta,\beta_0}(x_i)) + \lambda \|\beta\|^2 \right\}$$

where $h(y, f(x)) = (1 - yf(x))_+$, $f_{\beta,\beta_0}(x) = \langle \beta, x \rangle + \beta_0$

Remarks:

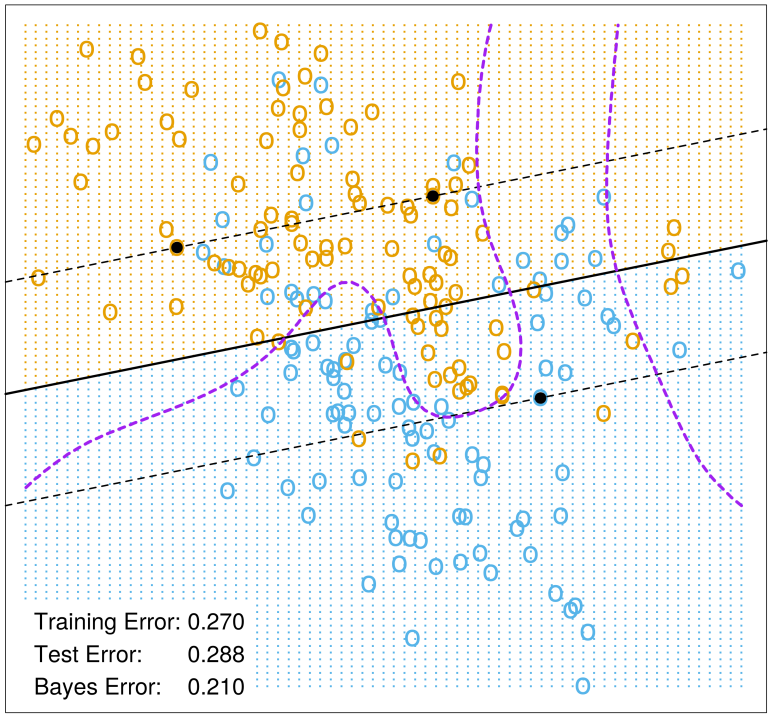
- $\sum_{i=1}^{n} h(y_i, f(x_i))$ quantifies the fit to the data
- $\|\beta\|^2$: regularization term (avoids too complex βs)
- Similar idea to the one discussed for the LASSO pb!

Limitations of the SV classifier



Sub-optimal linear classifier: $f_{\beta,\beta_0}(x) = \langle \beta, x \rangle + \beta_0$

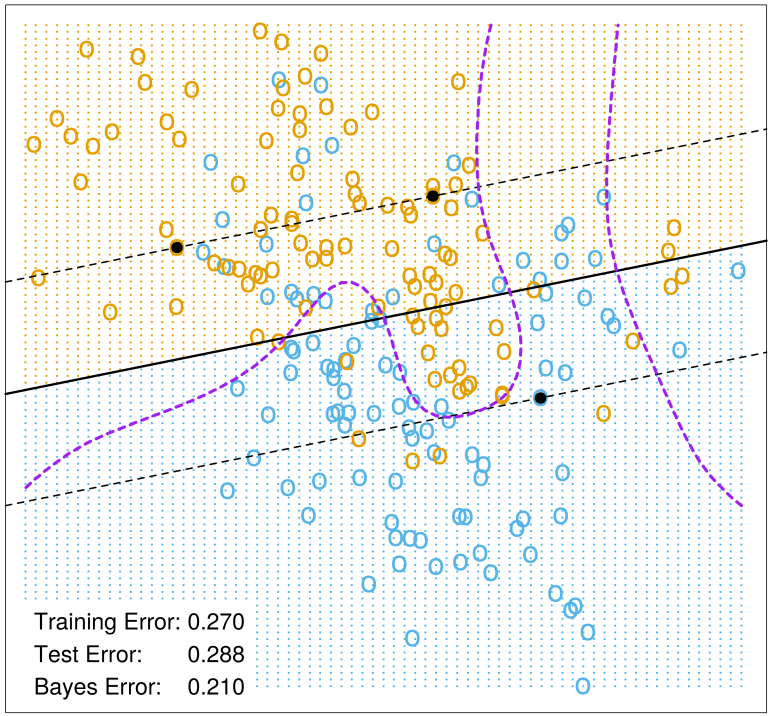
Limitations of the SV classifier



Sub-optimal linear classifier: $f_{\beta,\beta_0}(x) = \langle \beta, x \rangle + \beta_0$

- With strongly overlapping classes, linear classifiers are less effective (straight line): lots of misclassifications

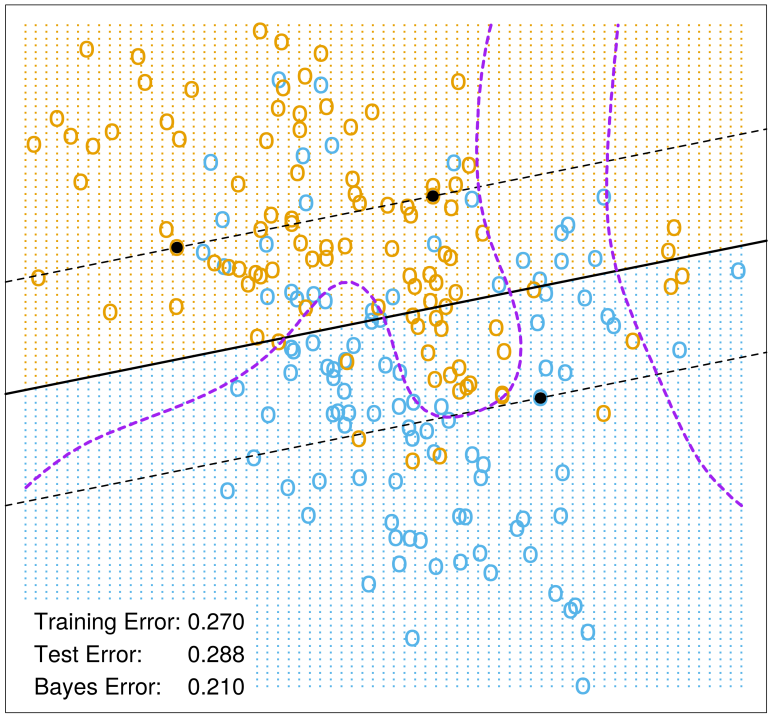
Limitations of the SV classifier



Sub-optimal linear classifier: $f_{\beta,\beta_0}(x) = \langle \beta, x \rangle + \beta_0$

- With strongly overlapping classes, linear classifiers are less effective (straight line): lots of misclassifications
- Comparison with the Bayes classifier (purple dashed line)

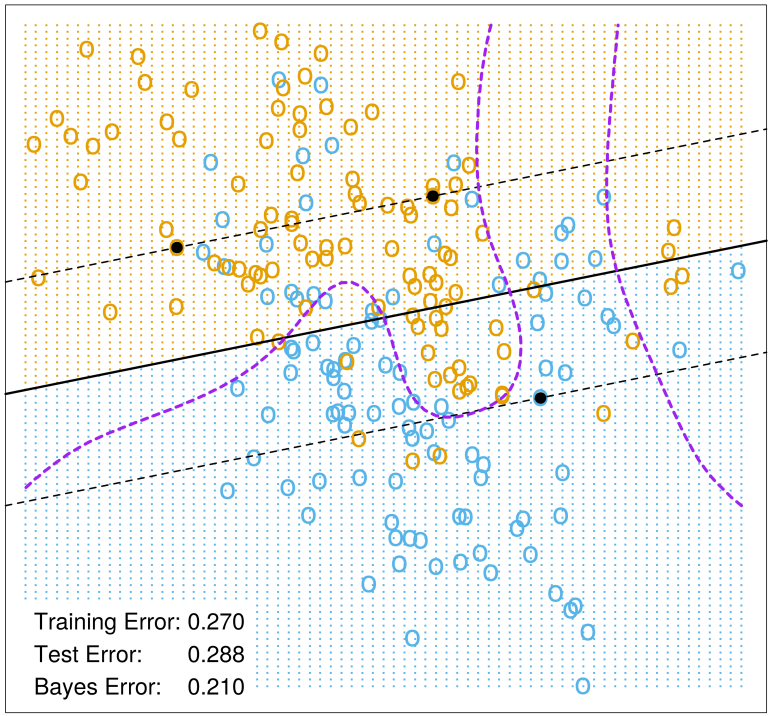
Limitations of the SV classifier



Sub-optimal linear classifier: $f_{\beta,\beta_0}(x) = \langle \beta, x \rangle + \beta_0$

- With strongly overlapping classes, linear classifiers are less effective (straight line): lots of misclassifications
- Comparison with the Bayes classifier (purple dashed line)
- Need for alternative non-linear classifiers

Limitations of the SV classifier

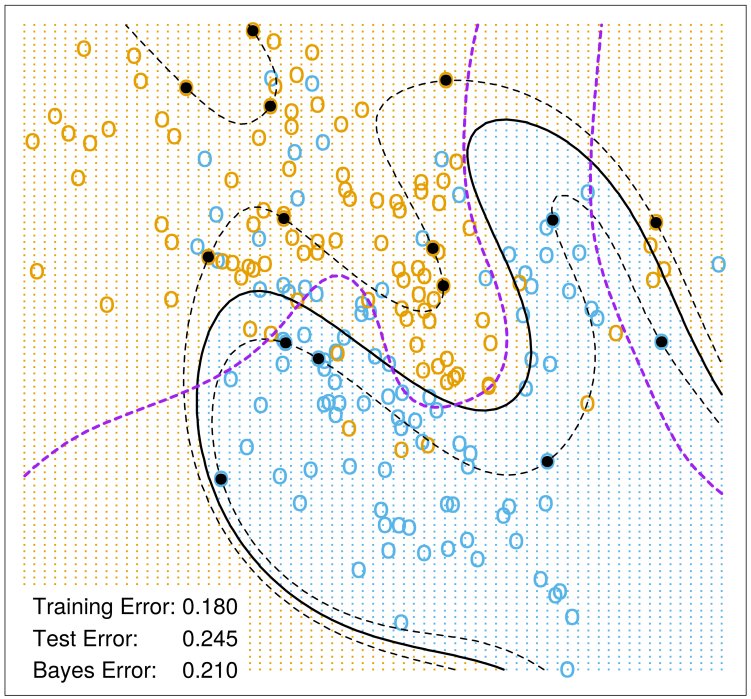


Sub-optimal linear classifier: $f_{\beta,\beta_0}(x) = \langle \beta, x \rangle + \beta_0$

- With strongly overlapping classes, linear classifiers are less effective (straight line): lots of misclassifications
- Comparison with the Bayes classifier (purple dashed line)
- Need for alternative non-linear classifiers
  $\longrightarrow$ **Where Reproducing kernels come into play...**

Reproducing Kernels

Improvement with Reproducing Kernels



Reproducing kernels provide non-linear boundaries

- The Bayes classifier: Non-linear boundary (purple dashed line)
- The Reproducing Kernel-based classifier (black line) is close to optimal

Illustration: "Polynomial extension"

Example: Degree 2-polynomial extension

- $x = (x_1, x_2) \in \mathbb{R}^2$ $(d = 2)$ and $y \in \{0, 1\}$
- Create new features:

$$\phi : \quad x \in \mathbb{R}^2 \mapsto \phi(x) = \left(x_1^2, x_1 \cdot x_2, x_2 \cdot x_1, x_2^2\right)^\top \in \mathbb{R}^4$$

- $\phi(x) \in \mathbb{R}^4$: higher dimensional vector than $x \in \mathbb{R}^2$

Illustration: "Polynomial extension"

Example: Degree 2-polynomial extension

- $x = (x_1, x_2) \in \mathbb{R}^2$ $(d = 2)$ and $y \in \{0, 1\}$
- Create new features:

$$\phi : \quad x \in \mathbb{R}^2 \mapsto \phi(x) = \left(x_1^2, x_1 \cdot x_2, x_2 \cdot x_1, x_2^2\right)^\top \in \mathbb{R}^4$$

- $\phi(x) \in \mathbb{R}^4$: higher dimensional vector than $x \in \mathbb{R}^2$

Support Vectors and extended features

Solve, for all $\lambda > 0$:

$$\min_{\beta \in \mathbb{R}^4, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, f_{\beta,\beta_0}(\phi(x_i))) + \lambda \|\beta\|^2 \right\}$$

where $f_{\beta,\beta_0}(\phi(x_i)) = \langle \beta, \phi(x_i) \rangle_4 + \beta_0$

(see previous picture)

From extended features to PSD kernels

From extended features to kernel

- $x, x' \in \mathbb{R}^d$
- Extended features: $\phi(x) \in \mathbb{R}^p$
- Kernel: $k(\cdot, \cdot) : (x, x') \mapsto k(x, x') \in \mathbb{R}$

$$k(x, x') = \left\langle \phi(x), \phi(x') \right\rangle_p$$

From extended features to PSD kernels

From extended features to kernel

- $x, x' \in \mathbb{R}^d$
- Extended features: $\phi(x) \in \mathbb{R}^p$
- Kernel: $k(\cdot,\cdot) : (x, x') \mapsto k(x, x') \in \mathbb{R}$

$$k(x, x') = \left\langle \phi(x), \phi(x') \right\rangle_p$$

Definition (Psd kernel)

A kernel $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is positive semi-definite if

- $k(x, y) = k(y, x)$, for all $x, y \in \mathbb{R}^d$ (symmetric)
- $\forall n \in \mathbb{N}^*, \forall x_1, \ldots, x_n \in \mathbb{R}^d, \forall a_1, \ldots, a_n \in \mathbb{R}$,

$$\sum_{1 \leq i,j \leq n} a_i a_j k(x_i, x_j) \geq 0$$

which is equivalent to
$\forall n \in \mathbb{N}^*, \forall x_1, \ldots, x_n \in \mathbb{R}^d$,

$$\text{the matrix } \mathrm{K} = \{k(x_i, x_j)\}_{1 \leq i,j \leq n} \in \mathcal{S}_n^+(\mathbb{R})$$

PSD kernel: Examples

How to build a PSD kernel? First example

- $\{\psi_a(\cdot)\}_{1 \leq a \leq A}$: a set (basis) of real-valued functions
- $\phi(x) = (\psi_1(x), \psi_2(x), \ldots, \psi_A(x))^\top \in \mathbb{R}^A$
- PSD kernel:

$$k(x, x') = \left\langle \phi(x), \phi(x') \right\rangle_{\mathbb{R}^A}$$

Other classical examples

- Linear kernel:

$$k(x, y) = \langle x, y \rangle_{\mathbb{R}^d}$$

- Polynomial kernel: $(c \geq 0,\ d > 0)$

$$k(x, y) = (\langle x, y \rangle_{\mathbb{R}^d} + c)^d$$

- Gaussian (Radial Basis Function) kernel:

$$k(x, y) = e^{-\frac{(x-y)^2}{2}}$$

Reproducing kernel and RKHS (Cont'd)

Definition (Reproducing kernel)

$k(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel on a Hilbert space $(\mathcal{H}, \langle \cdot,\cdot \rangle_{\mathcal{H}})$ if

- $\mathcal{H}$ contains all functions $k_x : x \mapsto k(x,\cdot)$, for all $x \in \mathcal{X}$
- $\forall x \in \mathcal{X}$, $\forall f \in \mathcal{H}$,

$$f(x) = \langle f, k_x \rangle_{\mathcal{H}} \qquad \text{(Reproducing property)}$$

Then, $\mathcal{H} = \mathcal{H}_k$: Reproducing Kernel Hilbert Space (RKHS) associated with k.

Definition (Gram matrix)

The so-called Gram matrix is the $n \times n$ matrix denoted by K and defined by

$$\mathrm{K} = \{k(x_i, x_j)\}_{1 \leq i,j \leq n}$$

RKHS: A few examples

Finite-dimensional examples

- If $\text{Card}(\mathcal{X}) < +\infty$, then $\mathcal{H} = \mathbb{R}^{\text{Card}(\mathcal{X})}$

- If $k(x, y) = \langle x, y \rangle_d$, then $\mathcal{H} = \mathbb{R}^d$

Infinite-dimensional examples

- If $k(x, y) = \min(x, y)$ $(x, y \in [0, 1])$, then

$$\mathcal{H} = \left\{ g : \mathcal{X} \to \mathbb{R} \mid g \in L^2(\mathcal{X}), g(0) = 0, \int_X \left(g'(t)\right)^2 \, dx < +\infty \right\}$$

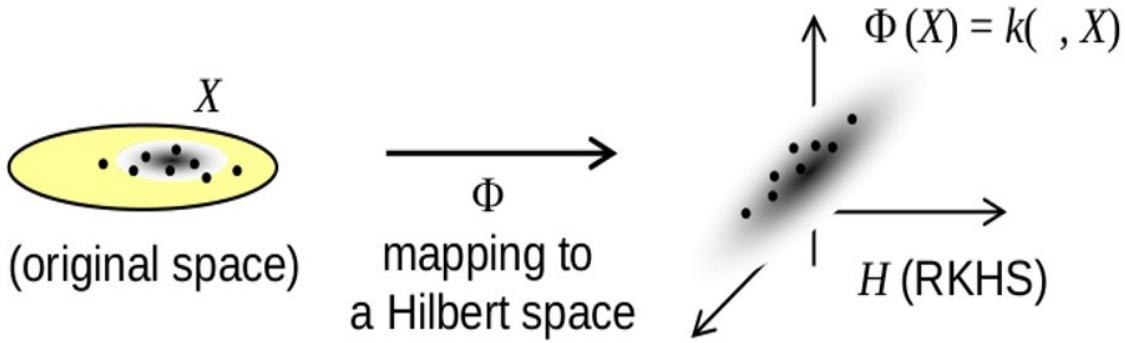
$\to$ function space...

Remark:
RKHS is an example where high-dimensional tools are involved (High-dimensional statistics) although data can belong to $\mathbb{R}^d$ with $d = 1$!

Canonical feature map and kernel trick

Mapping data from $\mathcal{X}$ to $\mathcal{H}$

- $k(\cdot, \cdot)$: reproducing kernel
- $x \in \mathbb{R}^d \mapsto k_x \in \mathcal{H}$: canonical feature map from $\mathbb{R}^d$ to $\mathcal{H}$
- $k_x = \phi(x) \in \mathcal{H}$: new "observation" in the RKHS



Problem:

$$\mathcal{H} \text{ is high-dimensional (function space)}$$

Kernel trick

$$\langle k_x, k_y \rangle_{\mathcal{H}} = k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}\,, \qquad \forall x, y \in \mathbb{R}^d$$

- Avoids manipulating data that are high-dimensional in $\mathcal{H}$

Support Vector Machines

From SV to SVM classifiers

Support Vectors with extended feature vectors

- Extended feature vector: $\phi(x) \in \mathbb{R}^p$
- Solve, for all $\lambda > 0$:

$$\widehat{f} = Arg \min_{\beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle \beta, \phi(x_i) \rangle_p + \beta_0) + \lambda \|\beta\|^2 \right\}$$

Support vectors with general feature map

- Kernel-based feature vector: $\phi(x) = k(x, \cdot) \in \mathcal{H}$
- **Key ingredients:** $(g \in \mathcal{H})$

Replace: $\langle \beta, \phi(x_i) \rangle_p$ by $\langle g, \phi(x_i) \rangle_{\mathcal{H}}$

$\|\beta\|$ by $\|g\|_{\mathcal{H}}$

Support Vector Machine (SVM)

Solve, for all $\lambda > 0$:

$$\widehat{f} = Arg \min_{g \in \mathcal{H}, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \|g\|_{\mathcal{H}}^2 \right\}$$

where $\widehat{f}(x) = \langle \widehat{g}, \phi(x) \rangle_{\mathcal{H}} + \widehat{\beta}_0$, for all $x \in \mathcal{X}$

SVM: Comments

Support Vector Machine (SVM)

Solve, for all $\lambda > 0$:

$$\widehat{f} = Arg \min_{g \in \mathcal{H}, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \|g\|_{\mathcal{H}}^2 \right\}$$

where $\widehat{f}(x) = \langle \widehat{g}, \phi(x) \rangle_{\mathcal{H}} + \widehat{\beta}_0$, for all $x \in \mathcal{X}$

Remark:

- $h(y, f(x)) = (1 - y \cdot f(x))_+$
- $\sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0)$ measures how well g fits the data
- $\lambda \|g\|_{\mathcal{H}}$: regularization term
  $\longrightarrow$ penalizes for too large (complex) $g \in \mathcal{H}$
- $\lambda > 0$: Regularization parameter
  - The larger λ, the smaller $\|\widehat{g}\|_{\mathcal{H}}$, the less complex $\widehat{g}$
  - Tradeoff between: fitting the data *vs* model complexity

Representer theorem

$$\widehat{f} = Arg \min_{g \in \mathcal{H}, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \|g\|_{\mathcal{H}}^2 \right\}$$

Theorem (Representer theorem)

$\Psi : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$, *nondecreasing w.r.t. its* $n+1$*th argument*

$$Arg \min_{g \in \mathcal{H}} \{ \Psi [g(x_1), \ldots, g(x_n), \|g\|_{\mathcal{H}}] \}$$

Any solution $\widehat{g}$ *to the above optimization problem can be written as*

$$\widehat{g}(x) = \sum_{i=1}^{n} \widehat{\alpha}_i k(x_i, x), \quad \forall x \in \mathcal{X}$$

where $\widehat{\alpha}_i \in \mathbb{R}$, *for all* $1 \leq i \leq n$

Proof of the Representer theorem

Proof.

- $V = \mathrm{Vect}\left(k_{x_1}, \ldots, k_{x_n}\right)$

Proof of the Representer theorem

Proof.

- $V = \mathrm{Vect}\left(k_{x_1}, \ldots, k_{x_n}\right)$
- Any $f \in V$ satisfies

$$x \mapsto f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \in \mathbb{R}$$

Proof of the Representer theorem

Proof.

- $V = \mathrm{Vect}\left(k_{x_1}, \ldots, k_{x_n}\right)$
- Any $f \in V$ satisfies

$$x \mapsto f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \in \mathbb{R}$$

Let g be a solution to this problem. Then

- $$g(x_j) = g_V(x_j) + g_{V^\perp}(x_j) = g_V(x_j) + \underbrace{\left\langle g_{V^\perp}, k_{x_j} \right\rangle_{\mathcal{H}}}_{=0}$$

Proof of the Representer theorem

Proof.

- $V = \mathrm{Vect}\left(k_{x_1}, \ldots, k_{x_n}\right)$
- Any $f \in V$ satisfies

$$x \mapsto f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \in \mathbb{R}$$

Let g be a solution to this problem. Then

- $$g(x_j) = g_V(x_j) + g_{V^\perp}(x_j) = g_V(x_j) + \underbrace{\left\langle g_{V^\perp}, k_{x_j} \right\rangle_{\mathcal{H}}}_{=0}$$
- $$\|g\|_{\mathcal{H}}^2 = \|g_V\|_{\mathcal{H}}^2 + \|g_{V^\perp}\|_{\mathcal{H}}^2 \geq \|g_V\|_{\mathcal{H}}^2$$

Proof of the Representer theorem

Proof.

- $V = \mathrm{Vect}\left(k_{x_1}, \ldots, k_{x_n}\right)$
- Any $f \in V$ satisfies

$$x \mapsto f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) \in \mathbb{R}$$

Let g be a solution to this problem. Then

- $$g(x_j) = g_V(x_j) + g_{V^\perp}(x_j) = g_V(x_j) + \underbrace{\left\langle g_{V^\perp}, k_{x_j} \right\rangle_{\mathcal{H}}}_{=0}$$
- $$\|g\|_{\mathcal{H}}^2 = \|g_V\|_{\mathcal{H}}^2 + \|g_{V^\perp}\|_{\mathcal{H}}^2 \geq \|g_V\|_{\mathcal{H}}^2$$

Hence,

$$\Psi\left[g(x_1), \ldots, g(x_n), \|g\|_{\mathcal{H}}\right] \geq \Psi\left[g_V(x_1), \ldots, g_V(x_n), \|g_V\|_{\mathcal{H}}\right]$$

Representer thm and finite dimension

$$\widehat{g}(x) = \sum_{j=1}^{n} \widehat{\alpha}_j k(x_j, x), \quad \forall x \in \mathcal{X}$$

From inifinite to finite dimension

Solving

$$\widehat{f} = Arg \min_{g \in \mathcal{H}, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \|g\|_{\mathcal{H}}^2 \right\}$$

Representer thm and finite dimension

$$\widehat{g}(x) = \sum_{j=1}^{n} \widehat{\alpha}_j k(x_j, x), \quad \forall x \in \mathcal{X}$$

From inifinite to finite dimension

Solving

$$\widehat{f} = Arg \min_{g \in \mathcal{H}, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \|g\|_{\mathcal{H}}^2 \right\}$$

equivalent to solving (K: Gram matrix)

$$\widehat{\alpha} = Arg \min_{\alpha \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, [K\alpha]_i + \beta_0) + \lambda \alpha^\top K \alpha \right\}$$

Representer thm and finite dimension

$$\widehat{g}(x) = \sum_{j=1}^{n} \widehat{\alpha}_j k(x_j, x), \quad \forall x \in \mathcal{X}$$

From inifinite to finite dimension

Solving

$$\widehat{f} = Arg \min_{g \in \mathcal{H}, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \|g\|_{\mathcal{H}}^2 \right\}$$

equivalent to solving (K: Gram matrix)

$$\widehat{\alpha} = Arg \min_{\alpha \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, [K\alpha]_i + \beta_0) + \lambda \alpha^\top K \alpha \right\}$$

Remarks:

- Representer thm $\Rightarrow$ Finite dimensional space
- Dimension of the solution space at most n!
- Not yet any closed-form expression for the solution...

Introducing Slack variables ξ_is

Solving (fixed $\lambda > 0$)

$$\min_{\alpha \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \alpha^\top K \alpha \right\}$$

is equivalent to solving

Introducing Slack variables ξ_is

Solving (fixed $\lambda > 0$)

$$\min_{\alpha \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \alpha^\top K \alpha \right\}$$

is equivalent to solving

$$\min_{\alpha, \xi \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} \xi_i + \lambda \alpha^\top K \alpha \right\}$$

such that $\xi_i \geq h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0), \quad \forall 1 \leq i \leq n$

is equivalent to solving

Introducing Slack variables ξ_is

Solving (fixed $\lambda > 0$)

$$\min_{\alpha \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0) + \lambda \alpha^\top K \alpha \right\}$$

is equivalent to solving

$$\min_{\alpha, \xi \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} \xi_i + \lambda \alpha^\top K \alpha \right\}$$

such that $\xi_i \geq h(y_i, \langle g, \phi(x_i) \rangle_{\mathcal{H}} + \beta_0), \quad \forall 1 \leq i \leq n$

is equivalent to solving

$$\min_{\alpha, \xi \in \mathbb{R}^n, \beta_0 \in \mathbb{R}} \left\{ \sum_{i=1}^{n} \xi_i + \lambda \alpha^\top K \alpha \right\}$$

such that $\xi_i \geq 1 - y_i \cdot \left[\sum_{j=1}^{n} \alpha_j k(x_j, x_i) + \beta_0 \right], \quad \forall 1 \leq i \leq n$

$\xi_i \geq 0, \quad \forall 1 \leq i \leq n$

Lagrangian formulation

Lagrangian formulation

($\mu, \nu \geq 0$: Langrange multipliers)

$$L_P(\xi, \alpha; \mu, \nu) = \sum_{i=1}^{n} \xi_i + \lambda \alpha^\top \mathsf{K} \alpha$$

$$- \sum_{i=1}^{n} \mu_i \left[\xi_i - 1 + y_i \cdot \left[\sum_{j=1}^{n} \alpha_j k(x_j, x_i) + \beta_0 \right] \right] - \sum_{i=1}^{n} \nu_i \xi_i$$

Lagrangian formulation

Lagrangian formulation ($\mu, \nu \geq 0$: Langrange multipliers)

$$L_P(\xi, \alpha; \mu, \nu) = \sum_{i=1}^{n} \xi_i + \lambda \alpha^\top \mathsf{K} \alpha$$
$$- \sum_{i=1}^{n} \mu_i \left[\xi_i - 1 + y_i \cdot \left[\sum_{j=1}^{n} \alpha_j k(x_j, x_i) + \beta_0 \right] \right] - \sum_{i=1}^{n} \nu_i \xi_i$$

Primal problem

- Minimizing $L_P(\xi, \alpha; \mu, \nu)$ w.r.t. α, ξ (computing derivatives)
- Leads to the dual $L_D(\mu, \nu) = L_P(\widehat{\xi}(\mu, \nu), \widehat{\alpha}(\mu, \nu); \mu, \nu)$

Dual optimization problem

Optimizing the Dual

$$\max_{0 \leq \mu_i \leq 1/n} \overbrace{\left\{ \sum_{i=1}^{n} \mu_i - \frac{1}{4\lambda} \sum_{1 \leq i,j \leq n} y_i y_j \mu_i \mu_j k(x_i, x_j) \right\}}^{=L_D(\mu,\nu)}$$

solved for $\mu_i = 2\lambda\alpha_i y_i$, for all $1 \leq i \leq n$ $\qquad (\mu_i + \nu_i = 1/n)$

Dual optimization problem

Optimizing the Dual

$$\max_{0\leq\mu_i\leq 1/n} \overbrace{\left\{ \sum_{i=1}^{n} \mu_i - \frac{1}{4\lambda} \sum_{1\leq i,j\leq n} y_i y_j \mu_i \mu_j k(x_i, x_j) \right\}}^{=L_D(\mu,\nu)}$$

solved for $\mu_i = 2\lambda\alpha_i y_i$, for all $1 \leq i \leq n$ $\qquad (\mu_i + \nu_i = 1/n)$

Towards a simplified Primal expression

$$\max_{\alpha\in\mathbb{R}^n} \left\{ 2\sum_{i=1}^{n} \alpha_i y_i - \sum_{1\leq i,j\leq n} \alpha_i \alpha_j k(x_i, x_j) \right\}$$

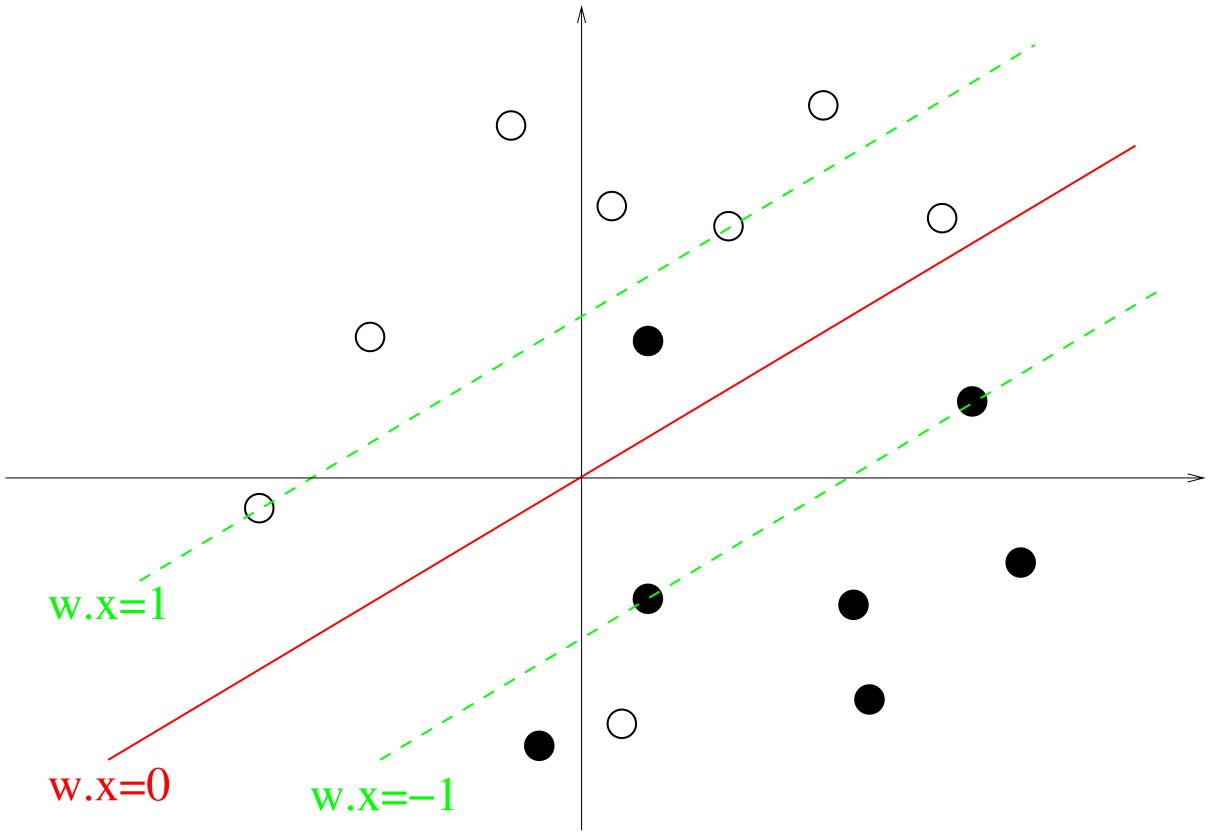
$$= \max_{\alpha\in\mathbb{R}^n} \left\{ 2\alpha^\top \mathsf{Y} - \alpha^\top \mathsf{K}\alpha \right\}$$

$$\text{such that} \quad 0 \leq \alpha_i y_i \leq \frac{1}{2n\lambda}, \qquad \forall i = 1, \ldots, n$$

Remark:

Easier to solve numerically (No closed-form expression)

Illustration: Where are support vectors?



KKT and support vectors

KKT conditions (when optimizing the Primal problem)

For all $1 \leq i \leq n$,

$$\mu_i \left[\xi_i - (1 - y_i f(x_i)) \right] = 0 \quad \text{and} \quad \nu_i \xi_i = 0$$

with $0 \leq \mu_i \leq 1/n$ and $\mu_i + \nu_i = 1/n$

KKT and support vectors

KKT conditions (when optimizing the Primal problem)

For all $1 \leq i \leq n$,

$$\mu_i \left[\xi_i - (1 - y_i f(x_i)) \right] = 0 \quad \text{and} \quad \nu_i \xi_i = 0$$

with $0 \leq \mu_i \leq 1/n$ and $\mu_i + \nu_i = 1/n$

Interpretation

1. $\mu_i = 0$:
   - $\nu_i = 1/n$ and $\xi_i = 0 \Rightarrow 1 < y_i f(x_i)$
   - x_i: well-classified

KKT and support vectors

KKT conditions (when optimizing the Primal problem)

For all $1 \leq i \leq n$,

$$\mu_i \left[\xi_i - (1 - y_i f(x_i)) \right] = 0 \quad \text{and} \quad \nu_i \xi_i = 0$$

with $0 \leq \mu_i \leq 1/n$ and $\mu_i + \nu_i = 1/n$

Interpretation

1. $\mu_i = 0$:
   - $\nu_i = 1/n$ and $\xi_i = 0 \Rightarrow 1 < y_i f(x_i)$
   - x_i: well-classified
2. $0 < \mu_i < 1/n$:
   - $0 < \nu_i < 1/n$ and $\xi_i = 0 = 1 - y_i f(x_i) \Rightarrow 1 = y_i f(x_i)$

KKT and support vectors

KKT conditions (when optimizing the Primal problem)

For all $1 \leq i \leq n$,

$$\mu_i \left[\xi_i - (1 - y_i f(x_i)) \right] = 0 \quad \text{and} \quad \nu_i \xi_i = 0$$

with $0 \leq \mu_i \leq 1/n$ and $\mu_i + \nu_i = 1/n$

Interpretation

1. $\mu_i = 0$:
   - $\nu_i = 1/n$ and $\xi_i = 0 \Rightarrow 1 < y_i f(x_i)$
   - x_i: well-classified
2. $0 < \mu_i < 1/n$:
   - $0 < \nu_i < 1/n$ and $\xi_i = 0 = 1 - y_i f(x_i) \Rightarrow 1 = y_i f(x_i)$
   - x_i on the slab boundary

KKT and support vectors

KKT conditions (when optimizing the Primal problem)

For all $1 \leq i \leq n$,

$$\mu_i \left[\xi_i - (1 - y_i f(x_i)) \right] = 0 \quad \text{and} \quad \nu_i \xi_i = 0$$

with $0 \leq \mu_i \leq 1/n$ and $\mu_i + \nu_i = 1/n$

Interpretation

1. $\mu_i = 0$:
   - $\nu_i = 1/n$ and $\xi_i = 0 \Rightarrow 1 < y_i f(x_i)$
   - x_i: well-classified
2. $0 < \mu_i < 1/n$:
   - $0 < \nu_i < 1/n$ and $\xi_i = 0 = 1 - y_i f(x_i) \Rightarrow 1 = y_i f(x_i)$
   - x_i on the slab boundary
3. $\mu_i = 1/n$:
   - $\nu_i = 0$ and $\xi_i \geq 0 \Rightarrow 1 \geq y_i f(x_i)$
   - x_i within the slab or misclassified

KKT and support vectors

KKT conditions (when optimizing the Primal problem)

For all $1 \leq i \leq n$,

$$\mu_i \left[\xi_i - (1 - y_i f(x_i)) \right] = 0 \quad \text{and} \quad \nu_i \xi_i = 0$$

with $0 \leq \mu_i \leq 1/n$ and $\mu_i + \nu_i = 1/n$

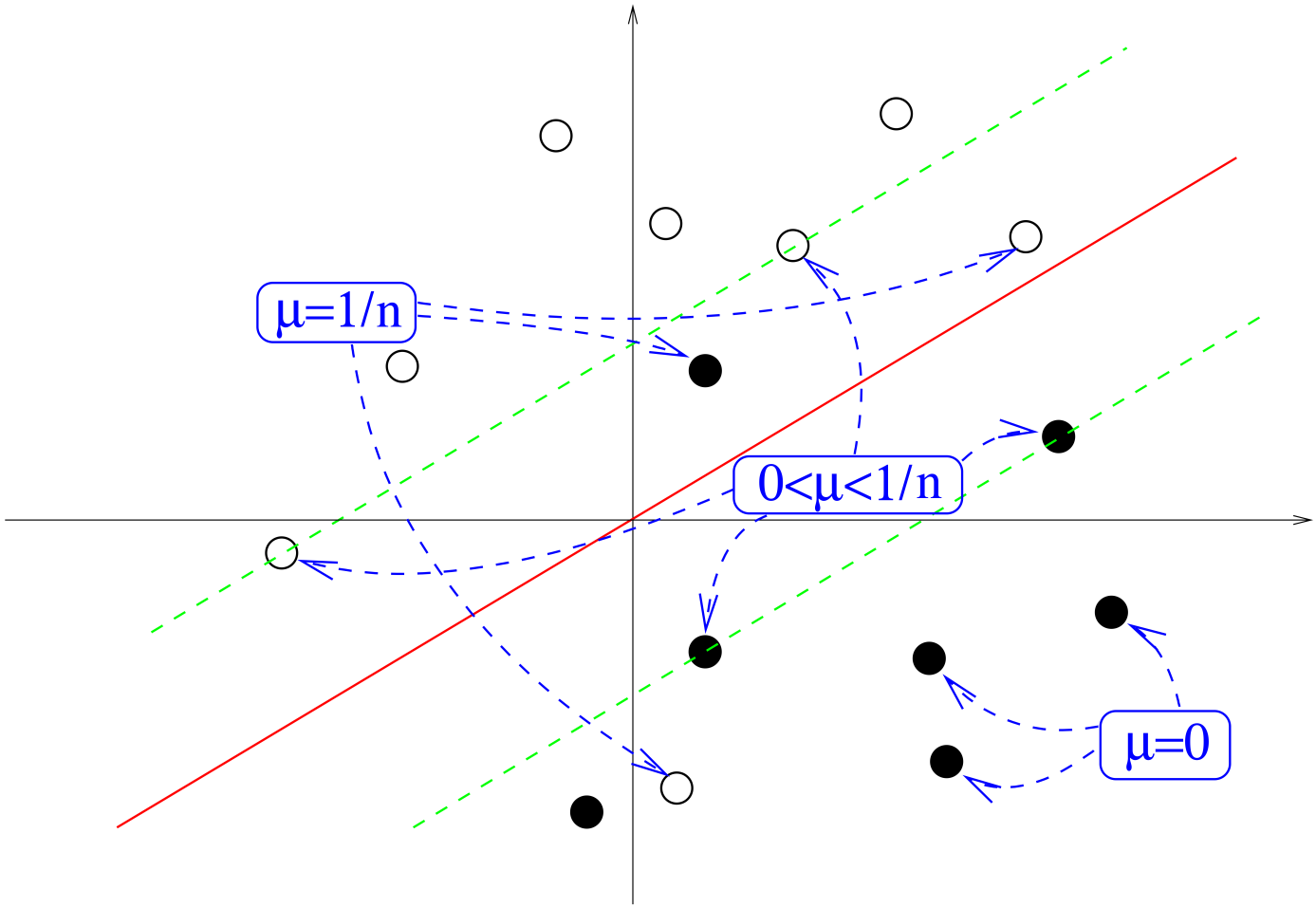
Interpretation

1. $\mu_i = 0$:
   - $\nu_i = 1/n$ and $\xi_i = 0 \Rightarrow 1 < y_i f(x_i)$
   - x_i: well-classified
2. $0 < \mu_i < 1/n$:
   - $0 < \nu_i < 1/n$ and $\xi_i = 0 = 1 - y_i f(x_i) \Rightarrow 1 = y_i f(x_i)$
   - x_i on the slab boundary
3. $\mu_i = 1/n$:
   - $\nu_i = 0$ and $\xi_i \geq 0 \Rightarrow 1 \geq y_i f(x_i)$
   - x_i within the slab or misclassified

Definition (SV)

Support vectors are the x_is such that $\mu_i = 2\lambda y_i \alpha_i > 0$

Illustration: Support vectors locations

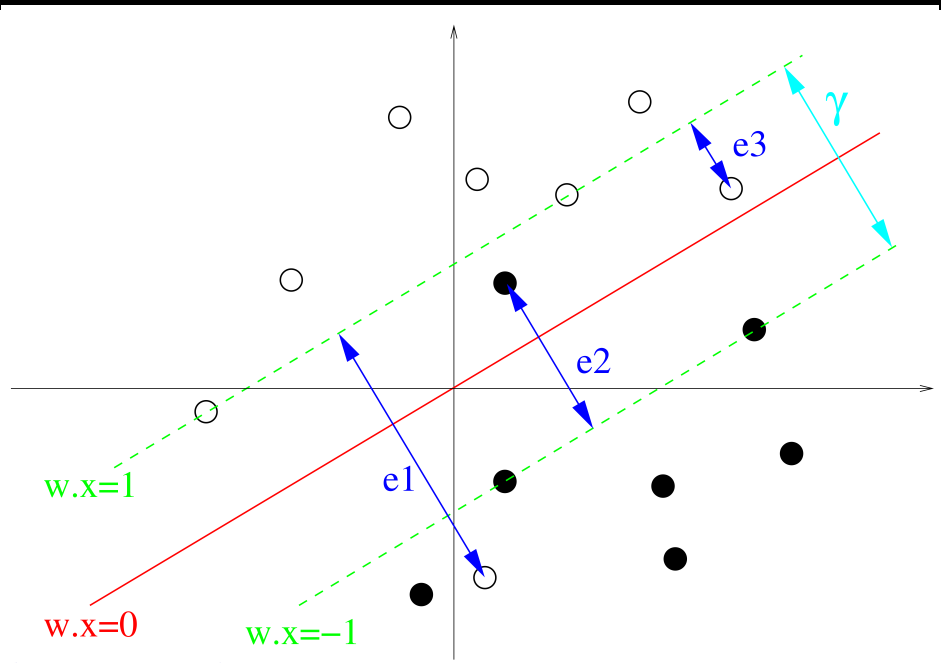


(from J.-Ph. Vert's slides)

Remark:

- $f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$, for all $x \in \mathcal{X}$
- Only a few α_is (μ_is) are non zero! (Support Vectors)

Margin



Definition (Margin)

In the SVM context, the margin $\gamma > 0$ denotes the width of the region within the RKHS between the hyperplanes of equations

$$\langle g, \phi(x) \rangle + \beta_0 = 1, \quad \text{and} \quad \langle g, \phi(x) \rangle + \beta_0 = -1$$

Remark:

$$\gamma = 2 \frac{y_i \left(\langle \widehat{g}, \phi(x_i) \rangle + \widehat{\beta}_0 \right)}{\|\widehat{g}\|_{\mathcal{H}}} = \frac{2}{\|\widehat{g}\|_{\mathcal{H}}}$$

SVM improvement