Random Matrices Meet Machine Learning: A Large Dimensional Analysis of LS-SVM

ICASSP'17

Zhenyu Liao, Romain Couillet

CentraleSupélec
Université Paris-Saclay
Paris, France

ICASSP'17, New Orleans, USA

CentraleSupélec

Outline

Outline

1. Motivation
2. Problem Statement
3. Main Results
4. Summary

Motivation

Performance analysis of SVM difficult:

- strongly data-driven

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form
- kernel non-linearity

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form
- kernel non-linearity

but important:

- a better understanding of algorithm

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form
- kernel non-linearity

but important:

- a better understanding of algorithm
- how to choose a good training set

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form
- kernel non-linearity

but important:

- a better understanding of algorithm
- how to choose a good training set
- how to choose a suitable kernel function f

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form
- kernel non-linearity

but important:

- a better understanding of algorithm
- how to choose a good training set
- how to choose a suitable kernel function f
- how to tune related hyperparameters?

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form
- kernel non-linearity

but important:

- a better understanding of algorithm
- how to choose a good training set
- how to choose a suitable kernel function f
- how to tune related hyperparameters?

Key interest: performance of SVM in the *Big Data* regime

- large dimension + large number of data

Motivation

Performance analysis of SVM difficult:

- strongly data-driven
- implicit form
- kernel non-linearity

but important:

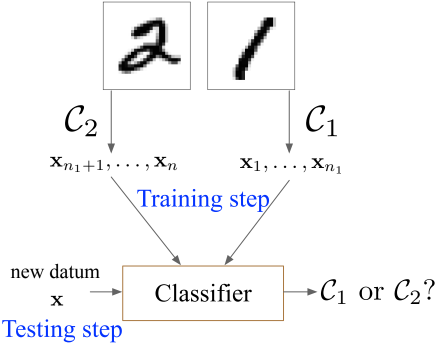
- a better understanding of algorithm
- how to choose a good training set
- how to choose a suitable kernel function f
- how to tune related hyperparameters?

Key interest: performance of SVM in the *Big Data* regime

- large dimension + large number of data
- more traceable thanks to large dimensional phenomenon

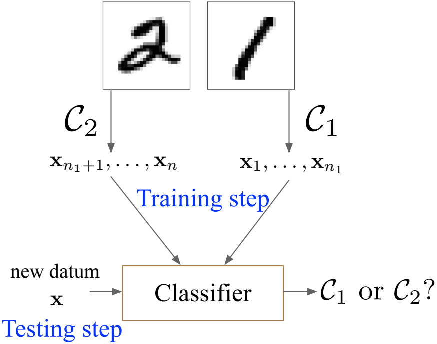
Outline

Binary Classification Problem



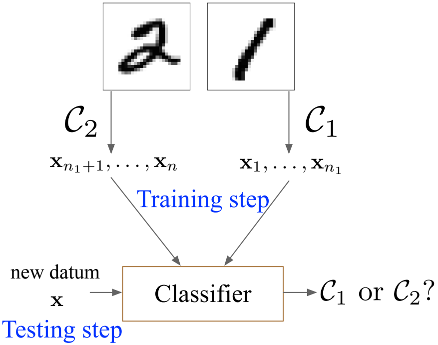
- **Training**:
  Training set: $\mathbf{x}_1, \ldots, \mathbf{x}_{n_1} \in \mathcal{C}_1$, $\mathbf{x}_{n_1+1}, \ldots, \mathbf{x}_n \in \mathcal{C}_2$.
  $\mathbf{x}_i \in \mathbb{R}^p, \forall i = 1, \ldots, n$.

Binary Classification Problem



- **Training**:
  Training set: $\mathbf{x}_1, \ldots, \mathbf{x}_{n_1} \in \mathcal{C}_1$, $\mathbf{x}_{n_1+1}, \ldots, \mathbf{x}_n \in \mathcal{C}_2$.
  $\mathbf{x}_i \in \mathbb{R}^p, \forall i = 1, \ldots, n.$
- **Test**:
  New datum $\mathbf{x} \Rightarrow$ which class?

Binary Classification Problem



- **Training**:
  Training set: $\mathbf{x}_1, \ldots, \mathbf{x}_{n_1} \in \mathcal{C}_1$, $\mathbf{x}_{n_1+1}, \ldots, \mathbf{x}_n \in \mathcal{C}_2$.
  $\mathbf{x}_i \in \mathbb{R}^p, \forall i = 1, \ldots, n$.
- **Test**:
  New datum $\mathbf{x} \Rightarrow$ which class?

Ideas

- statistical machine learning $\Rightarrow$ same distribution
- data non-linearly separable in data space $\Rightarrow$ kernel methods

Least Squares Support Vector Machines (1)

When $\mathcal{C}_1, \mathcal{C}_2$ are linearly separable.

Least Squares Support Vector Machines (1)

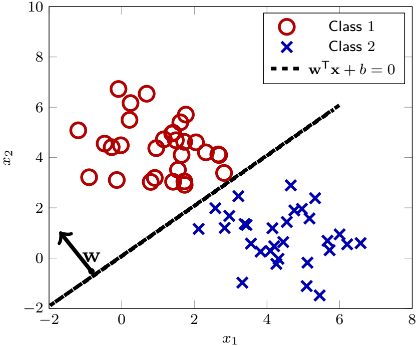
When $\mathcal{C}_1, \mathcal{C}_2$ are linearly separable.

To solve the optimization problem:

$$\arg\min_{\mathbf{w}} \quad J(\mathbf{w}, e) = \|\mathbf{w}\|^2 + \frac{\gamma}{n} \sum_{i=1}^{n} e_i^2$$

$$\text{such that} \quad y_i = \mathbf{w}^\mathsf{T} \mathbf{x}_i + b + e_i$$

$$\text{for } i = 1, \ldots, n$$



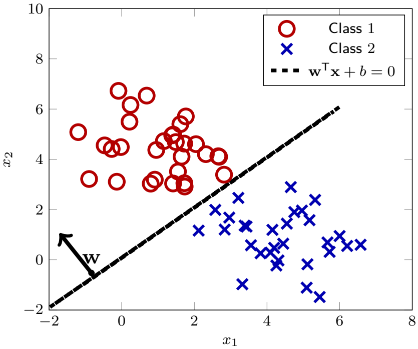
Least Squares Support Vector Machines (1)

When $\mathcal{C}_1, \mathcal{C}_2$ are linearly separable.

To solve the optimization problem:

$$\arg\min_{\mathbf{w}} \quad J(\mathbf{w}, e) = \|\mathbf{w}\|^2 + \frac{\gamma}{n} \sum_{i=1}^{n} e_i^2$$

$$\text{such that} \quad y_i = \mathbf{w}^\mathsf{T} \mathbf{x}_i + b + e_i \quad \text{for } i = 1, \ldots, n$$



In need of a transformation

- find proper features to classify
- when linear separation is impossible

Kernel trick

$$\mathbf{x} = [x_1,\ x_2]$$

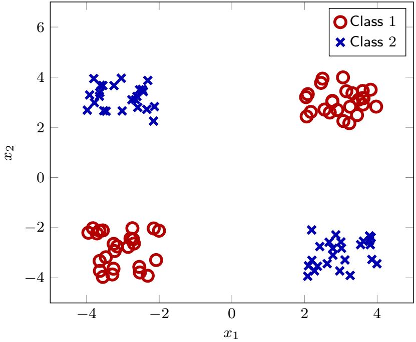


Figure: XOR example 2D visualization

Kernel trick

$$\mathbf{x} = [x_1,\ x_2] \overset{\boldsymbol{\varphi}(\cdot)}{\to} \boldsymbol{\varphi}(\mathbf{x}) = [x_1,\ x_2,\ x_1 x_2].$$

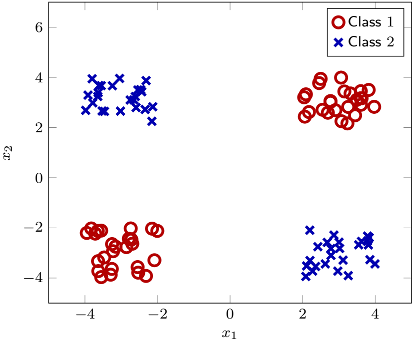


Figure: XOR example 2D visualization

Kernel trick

$$\mathbf{x} = [x_1,\ x_2] \overset{\boldsymbol{\varphi}(\cdot)}{\to} \boldsymbol{\varphi}(\mathbf{x}) = [x_1,\ x_2,\ x_1 x_2].$$

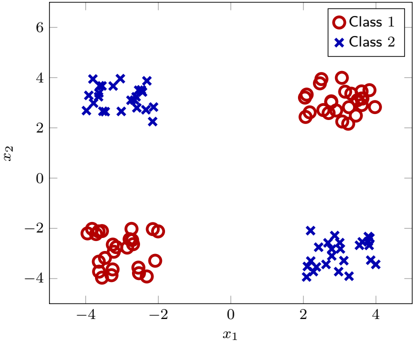


Figure: XOR example 2D visualization

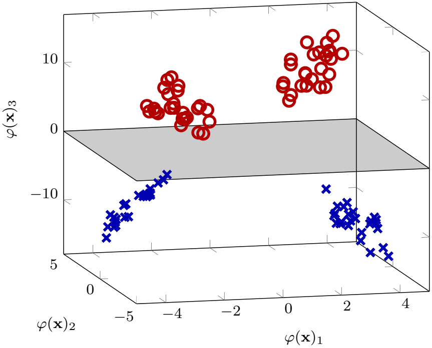


Figure: XOR example 3D visualization

Least Squares Support Vector Machines (2)

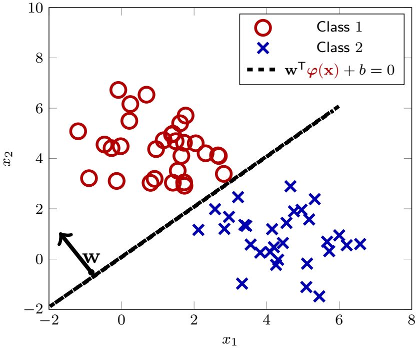
More generally, when linear separable impossible.

To solve the optimization problem:

$$\arg\min_{\mathbf{w}} \quad J(\mathbf{w}, e) = \|\mathbf{w}\|^2 + \frac{\gamma}{n} \sum_{i=1}^{n} e_i^2$$

such that $\quad y_i = \mathbf{w}^\mathsf{T} \varphi(\mathbf{x}_i) + b + e_i$

for $i = 1, \ldots, n$



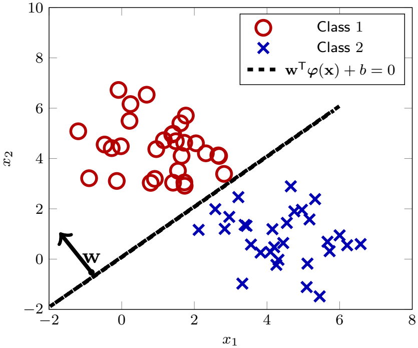
More generally, when linear separable impossible.

To solve the optimization problem:

$$\arg\min_{\mathbf{w}} \quad J(\mathbf{w}, e) = \|\mathbf{w}\|^2 + \frac{\gamma}{n}\sum_{i=1}^{n} e_i^2$$

$$\text{such that} \quad y_i = \mathbf{w}^\mathsf{T}\varphi(\mathbf{x}_i) + b + e_i$$

$$\text{for } i = 1, \ldots, n$$

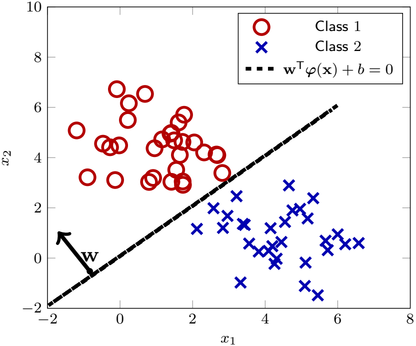


Idea

Structural risk Versus

Least Squares Support Vector Machines (2)

More generally, when linear separable impossible.



To solve the optimization problem:

$$\arg\min_{\mathbf{w}} \quad J(\mathbf{w}, e) = \|\mathbf{w}\|^2 + \frac{\gamma}{n} \sum_{i=1}^{n} e_i^2$$

$$\text{such that} \quad y_i = \mathbf{w}^\mathsf{T}\varphi(\mathbf{x}_i) + b + e_i$$

$$\text{for } i = 1, \ldots, n$$

Idea

Structural risk Versus Empirical risk

Least Squares Support Vector Machines (3)

- **Training**: solution given by

$$\begin{cases} \boldsymbol{\alpha} & = \mathbf{S}\left(\mathbf{I}_n - \frac{\mathbf{1}_n \mathbf{1}_n^\mathsf{T} \mathbf{S}}{\mathbf{1}_n^\mathsf{T} \mathbf{S} \mathbf{1}_n}\right)\mathbf{y} = \mathbf{S}\left(\mathbf{y} - b\mathbf{1}_n\right) \\ b & = \frac{\mathbf{1}_n^\mathsf{T} \mathbf{S} \mathbf{y}}{\mathbf{1}_n^\mathsf{T} \mathbf{S} \mathbf{1}_n} \end{cases} \tag{1}$$

with $\mathbf{S} \equiv \left(\mathbf{K} + \frac{n}{\gamma}\mathbf{I}_n\right)^{-1}$ the resolvent of the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$

$$\mathbf{K} \equiv \left\{\boldsymbol{\varphi}(\mathbf{x}_i)^\mathsf{T} \boldsymbol{\varphi}(\mathbf{x}_j)\right\}_{i,j=1}^{n} = \left\{ f\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{p}\right)\right\}_{i,j=1}^{n}, \tag{2}$$

for some *translation invariant* kernel function $f : \mathbb{R}_+ \mapsto \mathbb{R}_+$, $\mathbf{y} \equiv [y_1, \ldots, y_n]^\mathsf{T}$ and $\boldsymbol{\alpha} \equiv [\alpha_1, \ldots, \alpha_n]^\mathsf{T}$.

Least Squares Support Vector Machines (3)

- **Training**: solution given by

$$\begin{cases} \boldsymbol{\alpha} & = \mathbf{S}\left(\mathbf{I}_n - \frac{\mathbf{1}_n \mathbf{1}_n^\mathsf{T} \mathbf{S}}{\mathbf{1}_n^\mathsf{T} \mathbf{S} \mathbf{1}_n}\right)\mathbf{y} = \mathbf{S}\left(\mathbf{y} - b\mathbf{1}_n\right) \\ b & = \frac{\mathbf{1}_n^\mathsf{T} \mathbf{S} \mathbf{y}}{\mathbf{1}_n^\mathsf{T} \mathbf{S} \mathbf{1}_n} \end{cases} \tag{1}$$

with $\mathbf{S} \equiv \left(\mathbf{K} + \frac{n}{\gamma}\mathbf{I}_n\right)^{-1}$ the resolvent of the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$

$$\mathbf{K} \equiv \left\{\boldsymbol{\varphi}(\mathbf{x}_i)^\mathsf{T} \boldsymbol{\varphi}(\mathbf{x}_j)\right\}_{i,j=1}^n = \left\{ f\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{p}\right)\right\}_{i,j=1}^n, \tag{2}$$

for some *translation invariant* kernel function $f : \mathbb{R}_+ \mapsto \mathbb{R}_+$, $\mathbf{y} \equiv [y_1, \ldots, y_n]^\mathsf{T}$ and $\boldsymbol{\alpha} \equiv [\alpha_1, \ldots, \alpha_n]^\mathsf{T}$.

- **Test**: the decision function for a new $\mathbf{x}$

$$g(\mathbf{x}) = \boldsymbol{\alpha}^\mathsf{T} \mathbf{k}(\mathbf{x}) + b \tag{3}$$

where $\mathbf{k}(\mathbf{x}) = \left\{ f\left(\|\mathbf{x}_j - \mathbf{x}\|^2/p\right)\right\}_{j=1}^n \in \mathbb{R}^n$.
$\Rightarrow$ In practice, $\mathrm{sign}(g(\mathbf{x}))$ to predict the class.

Least Squares Support Vector Machines (3)

- **Training**: solution given by

$$\begin{cases} \boldsymbol{\alpha} & = \mathbf{S}\left(\mathbf{I}_n - \frac{\mathbf{1}_n\mathbf{1}_n^\mathsf{T}\mathbf{S}}{\mathbf{1}_n^\mathsf{T}\mathbf{S}\mathbf{1}_n}\right)\mathbf{y} = \mathbf{S}\left(\mathbf{y} - b\mathbf{1}_n\right) \\ b & = \frac{\mathbf{1}_n^\mathsf{T}\mathbf{S}\mathbf{y}}{\mathbf{1}_n^\mathsf{T}\mathbf{S}\mathbf{1}_n} \end{cases} \tag{1}$$

with $\mathbf{S} \equiv \left(\mathbf{K} + \frac{n}{\gamma}\mathbf{I}_n\right)^{-1}$ the resolvent of the kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$

$$\mathbf{K} \equiv \left\{\boldsymbol{\varphi}(\mathbf{x}_i)^\mathsf{T}\boldsymbol{\varphi}(\mathbf{x}_j)\right\}_{i,j=1}^n = \left\{f\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{p}\right)\right\}_{i,j=1}^n, \tag{2}$$

for some *translation invariant* kernel function $f : \mathbb{R}_+ \mapsto \mathbb{R}_+$, $\mathbf{y} \equiv [y_1, \ldots, y_n]^\mathsf{T}$ and $\boldsymbol{\alpha} \equiv [\alpha_1, \ldots, \alpha_n]^\mathsf{T}$.

- **Test**: the decision function for a new $\mathbf{x}$

$$g(\mathbf{x}) = \boldsymbol{\alpha}^\mathsf{T}\mathbf{k}(\mathbf{x}) + b \tag{3}$$

where $\mathbf{k}(\mathbf{x}) = \left\{f\left(\|\mathbf{x}_j - \mathbf{x}\|^2/p\right)\right\}_{j=1}^n \in \mathbb{R}^n$.

$\Rightarrow$ In practice, $\mathrm{sign}(g(\mathbf{x}))$ to predict the class.

Advantage

explicit form of the kernel matrix $\mathbf{K}$ and the vector $\mathbf{k}$

Outline

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1, 2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p \times p}$

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1, 2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

  for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p \times p}$
- **Non-trivial regime**: the asymptotic classification rate neither 1 nor 0

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1,2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

  for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p\times p}$
- **Non-trivial regime**: the asymptotic classification rate neither 1 nor 0
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1, 2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

  for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p \times p}$
- **Non-trivial regime**: the asymptotic classification rate neither 1 nor 0
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1, 2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

  for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p \times p}$
- **Non-trivial regime**: the asymptotic classification rate neither 1 nor 0
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$
- **Technical assumptions**:

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1, 2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

  for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p \times p}$
- **Non-trivial regime**: the asymptotic classification rate neither 1 nor 0
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$
- **Technical assumptions**:
  - $\mathbf{C}^\circ \equiv c_1 \mathbf{C}_1 + c_2 \mathbf{C}_2$, $c_1 \equiv \frac{n_1}{n}$ and $c_2 \equiv \frac{n_2}{n} = 1 - c_1$

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1, 2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

  for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p \times p}$
- **Non-trivial regime**: the asymptotic classification rate neither 1 nor 0
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$
- **Technical assumptions**:
  - $\mathbf{C}^\circ \equiv c_1 \mathbf{C}_1 + c_2 \mathbf{C}_2$, $c_1 \equiv \frac{n_1}{n}$ and $c_2 \equiv \frac{n_2}{n} = 1 - c_1$
  - as $n \to \infty$, we have $\frac{2}{p} \operatorname{tr} C^\circ \to \tau > 0$

Asymptotic Regime: Growth Rate Assumptions

- **Large dimension**: $n, p \to \infty$ and $\frac{p}{n} \to c_0$
- **Gaussian mixture model**: for $a \in \{1, 2\}$:

$$\mathbf{x}_i \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$$

  for some $\boldsymbol{\mu}_a \in \mathbb{R}^p$ and nonnegative definite $\mathbf{C}_a \in \mathbb{R}^{p \times p}$
- **Non-trivial regime**: the asymptotic classification rate neither 1 nor 0
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$
- **Technical assumptions**:
  - $\mathbf{C}^\circ \equiv c_1\mathbf{C}_1 + c_2\mathbf{C}_2$, $c_1 \equiv \frac{n_1}{n}$ and $c_2 \equiv \frac{n_2}{n} = 1 - c_1$
  - as $n \to \infty$, we have $\frac{2}{p} \operatorname{tr} C^\circ \to \tau > 0$
  - the kernel function f is three-times differentiable

Motivation: recall

Performance analysis of SVM difficult:

- strongly data-driven

Performance analysis of SVM difficult:

- strongly data-driven
  - $\Rightarrow$ data from Gaussian mixture model

Motivation: recall

Performance analysis of SVM difficult:

- strongly data-driven
  - $\Rightarrow$ data from Gaussian mixture model
- implicit form

Motivation: recall

Performance analysis of SVM difficult:

- strongly data-driven
  - $\Rightarrow$ data from Gaussian mixture model
- implicit form
  - $\Rightarrow$ study of LS-SVM as a first step

Performance analysis of SVM difficult:

- strongly data-driven
  - $\Rightarrow$ data from Gaussian mixture model
- implicit form
  - $\Rightarrow$ study of LS-SVM as a first step
- tkernel non-linearity

Motivation: recall

Performance analysis of SVM difficult:

- strongly data-driven
  - $\Rightarrow$ data from Gaussian mixture model
- implicit form
  - $\Rightarrow$ study of LS-SVM as a first step
- tkernel non-linearity
  - $\Rightarrow$ under appropriate growth rate condition, the kernel matrix $\mathbf{K}$ can be linearized

Linearization of the kernel matrix (1)

Recall

- kernel matrix $\mathbf{K}$: $\mathbf{K}_{i,j} = f\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{p}\right)$
- growth rate assumptions
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$
- Gaussian data: $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$ or $\mathbf{x}_i = \boldsymbol{\mu}_a + \sqrt{p}\mathbf{w}_i$ where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \frac{\mathbf{C}_a}{p})$

For $\mathbf{x}_i \in \mathcal{C}_a$ and $\mathbf{x}_j \in \mathcal{C}_b$

$$\frac{1}{p}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{w}_i - \mathbf{w}_j\|^2 + \underbrace{\frac{1}{p}\|\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\|^2}_{O(n^{-1})} + \underbrace{\frac{2}{\sqrt{p}}(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b)^\mathsf{T}(\mathbf{w}_i - \mathbf{w}_j)}_{O(n^{-1})}$$

$$\|\mathbf{w}_i - \mathbf{w}_j\|^2 = \|\mathbf{w}_i\|^2 + \|\mathbf{w}_j\|^2 - \underbrace{2\mathbf{w}_i^\mathsf{T}\mathbf{w}_j}_{O(n^{-1/2})}$$

$$= \mathbb{E}[\|\mathbf{w}_i\|^2] + \mathbb{E}[\|\mathbf{w}_j\|^2] + \underbrace{\|\mathbf{w}_i\|^2 - \mathbb{E}[\|\mathbf{w}_i\|^2] + \|\mathbf{w}_j\|^2 - \mathbb{E}[\|\mathbf{w}_j\|^2] - 2\mathbf{w}_i^\mathsf{T}\mathbf{w}_j}_{O(n^{-1/2})}$$

$$= \frac{1}{p}\operatorname{tr}\mathbf{C}_a + \frac{1}{p}\operatorname{tr}\mathbf{C}_a + O(n^{-1/2})$$

Linearization of the kernel matrix (2)

Recall: growth rate assumptions

- $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
- $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$

For $\mathbf{x}_i \in \mathcal{C}_a$ and $\mathbf{x}_j \in \mathcal{C}_b$

$$
\begin{aligned}
\|\mathbf{w}_i - \mathbf{w}_j\|^2 &= \|\mathbf{w}_i\|^2 + \|\mathbf{w}_j\|^2 - \underbrace{2\mathbf{w}_i^\mathsf{T}\mathbf{w}_j}_{O(n^{-1/2})} \\
&= \mathbb{E}[\|\mathbf{w}_i\|^2] + \mathbb{E}[\|\mathbf{w}_j\|^2] + \underbrace{\|\mathbf{w}_i\|^2 - \mathbb{E}[\|\mathbf{w}_i\|^2] + \|\mathbf{w}_j\|^2 - \mathbb{E}[\|\mathbf{w}_j\|^2] - 2\mathbf{w}_i^\mathsf{T}\mathbf{w}_j}_{O(n^{-1/2})} \\
&= \frac{1}{p}\operatorname{tr}\mathbf{C}_a + \frac{1}{p}\operatorname{tr}\mathbf{C}_a + O(n^{-1/2}) \\
&= \underbrace{\frac{2}{p}\operatorname{tr}\mathbf{C}^\circ}_{\equiv\tau=O(1)} + \underbrace{\frac{1}{p}\operatorname{tr}(\mathbf{C}_a - \mathbf{C}^\circ) + \frac{1}{p}\operatorname{tr}(\mathbf{C}_b - \mathbf{C}^\circ)}_{O(n^{-1/2})} + O(n^{-1/2})
\end{aligned}
$$

thus

$$\frac{1}{p}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \tau + O(n^{-1/2}).$$

Linearization of the kernel matrix (3)

Recall: kenel matrix

$$\mathbf{K}_{i,j} = f\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{p}\right)$$

For $\mathbf{x}_i \in \mathcal{C}_a$ and $\mathbf{x}_j \in \mathcal{C}_b$:

$$\frac{1}{p}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \tau + O(n^{-1/2}).$$

thus for $\mathbf{K}_{i,j}$

$$\begin{aligned}\mathbf{K}_{i,j} &= f\left(\tau + O(n^{-1/2})\right)\\ &= f(\tau) + f'(\tau)[\ldots] + f''(\tau)[\ldots]\ldots\end{aligned}$$

or in matrix form

$$\mathbf{K} = f(\tau)\mathbf{1}_n\mathbf{1}_n^{\mathsf{T}} + f'(\tau)[\ldots] + f''(\tau)[\ldots] + \ldots$$

Linearization of the kernel matrix (3)

Recall: kenel matrix

$$\mathbf{K}_{i,j} = f\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{p}\right)$$

For $\mathbf{x}_i \in \mathcal{C}_a$ and $\mathbf{x}_j \in \mathcal{C}_b$:

$$\frac{1}{p}\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \tau + O(n^{-1/2}).$$

thus for $\mathbf{K}_{i,j}$

$$\begin{aligned}\mathbf{K}_{i,j} &= f\left(\tau + O(n^{-1/2})\right)\\ &= f(\tau) + f'(\tau)[\ldots] + f''(\tau)[\ldots]\ldots\end{aligned}$$

or in matrix form

$$\mathbf{K} = f(\tau)\mathbf{1}_n\mathbf{1}_n^\mathsf{T} + f'(\tau)[\ldots] + f''(\tau)[\ldots] + \ldots$$

What is coming next?

key object decision function $g(\mathbf{x})$

$$g(\mathbf{x}) = \boldsymbol{\alpha}^\mathsf{T}\mathbf{k}(\mathbf{x}) + b$$

depend explicitly on $\mathbf{K}$ and vector $\mathbf{k}$.

Asymptotic Behavior of the Decision Function

Theorem

Under previous assumptions, for $\mathbf{x} \in \mathcal{C}_a$, $a \in \{1,2\}$, *we have*

$$n\left(g(\mathbf{x}) - G_a\right) \xrightarrow{d} 0$$

where $G_a \sim \mathcal{N}(\mathrm{E}_a, \mathrm{Var}_a)$ *with*

$$\mathrm{E}_a = \begin{cases} c_2 - c_1 - 2c_2 \cdot c_1 c_2 \gamma \mathfrak{D} \ , & a = 1 \\ c_2 - c_1 + 2c_1 \cdot c_1 c_2 \gamma \mathfrak{D} \ , & a = 2 \end{cases}$$

$$\mathrm{Var}_a = 8\gamma^2 c_1^2 c_2^2 \left(\mathcal{V}_1^a + \mathcal{V}_2^a + \mathcal{V}_3^a\right)$$

and

$$\mathfrak{D} = -\frac{2f'(\tau)}{p}\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2 + \frac{f''(\tau)}{p^2}\left(\mathrm{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 + \frac{2f''(\tau)}{p^2}\,\mathrm{tr}\left(\left(\mathbf{C}_2 - \mathbf{C}_1\right)^2\right)$$

$$\mathcal{V}_1^a = \frac{\left(f''(\tau)\right)^2}{p^4}\left(\mathrm{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 \mathrm{tr}\mathbf{C}_a^2$$

$$\mathcal{V}_2^a = \frac{2\left(f'(\tau)\right)^2}{p^2}\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)^\mathsf{T} \mathbf{C}_a \left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)$$

$$\mathcal{V}_3^a = \frac{2\left(f'(\tau)\right)^2}{np^2}\left(\frac{\mathrm{tr}\,\mathbf{C}_1\mathbf{C}_a}{c_1} + \frac{\mathrm{tr}\,\mathbf{C}_2\mathbf{C}_a}{c_2}\right)$$

Simulations on Gaussian data

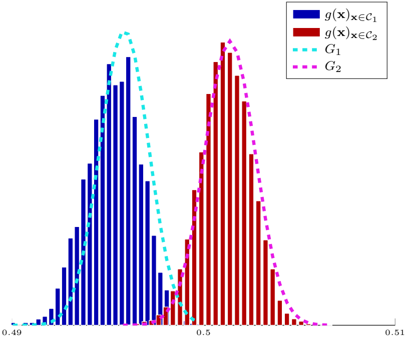


Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 512$, $c_1 = 1/4, c_2 = 3/4$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$ with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 3; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{5}{\sqrt{p}})$.

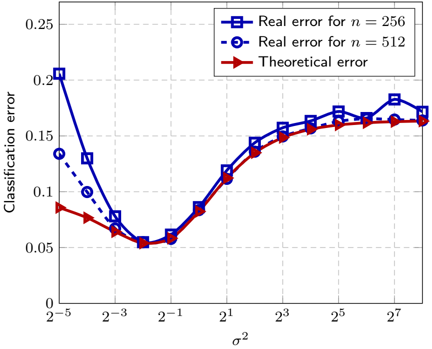


Figure: Performance of LS-SVM, $c_0 = 2$, $c_1 = c_2 = 1/2$, $\gamma = 1$, Gaussian kernel. $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$, with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{4}{\sqrt{p}})$.

More discussions

Several remarks:

1. imbalanced training data:
   $c_2 - c_1 \neq 0$
   $\Rightarrow$ catastrophe!

Theorem

$n\left(g(\mathbf{x}) - G_a\right) \xrightarrow{d} 0$ *and* $G_a \sim \mathcal{N}(\mathrm{E}_a, \mathrm{Var}_a)$ *with*

$$\mathrm{E}_a = \begin{cases} c_2 - c_1 - 2c_2 \cdot c_1 c_2 \gamma \mathfrak{D} \ , & a = 1 \\ c_2 - c_1 + 2c_1 \cdot c_1 c_2 \gamma \mathfrak{D} \ , & a = 2 \end{cases}$$

$$\mathrm{Var}_a = 8\gamma^2 c_1^2 c_2^2 \left(\mathcal{V}_1^a + \mathcal{V}_2^a + \mathcal{V}_3^a\right)$$

and

$$\mathfrak{D} = -\frac{2f'(\tau)}{p}\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2 + \frac{f''(\tau)}{p^2}\left(\mathrm{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 + \frac{2f''(\tau)}{p^2}\mathrm{tr}\left(\left(\mathbf{C}_2 - \mathbf{C}_1\right)^2\right)$$

$$\mathcal{V}_1^a = \frac{\left(f''(\tau)\right)^2}{p^4}\left(\mathrm{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 \mathrm{tr}\mathbf{C}_a^2$$

$$\mathcal{V}_2^a = \frac{2\left(f'(\tau)\right)^2}{p^2}\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)^{\mathsf{T}} \mathbf{C}_a \left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)$$

$$\mathcal{V}_3^a = \frac{2\left(f'(\tau)\right)^2}{np^2}\left(\frac{\mathrm{tr}\,\mathbf{C}_1\mathbf{C}_a}{c_1} + \frac{\mathrm{tr}\,\mathbf{C}_2\mathbf{C}_a}{c_2}\right)$$

More discussions

Several remarks:

1. imbalanced training data: $c_2 - c_1 \neq 0$ $\Rightarrow$ catastrophe!
2. $\mathfrak{D}$ as large as possible: conditions of signs of the kernel function f $\Rightarrow f'(\tau) < 0$ and $f''(\tau) > 0$

Theorem

$n\left(g(\mathbf{x}) - G_a\right) \xrightarrow{d} 0$ *and* $G_a \sim \mathcal{N}(\mathrm{E}_a, \mathrm{Var}_a)$ *with*

$$\mathrm{E}_a = \begin{cases} c_2 - c_1 - 2c_2 \cdot c_1 c_2 \gamma \mathfrak{D} \,, & a = 1 \\ c_2 - c_1 + 2c_1 \cdot c_1 c_2 \gamma \mathfrak{D} \,, & a = 2 \end{cases}$$

$$\mathrm{Var}_a = 8\gamma^2 c_1^2 c_2^2 \left(\mathcal{V}_1^a + \mathcal{V}_2^a + \mathcal{V}_3^a\right)$$

and

$$\mathfrak{D} = -\frac{2f'(\tau)}{p}\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2 + \frac{f''(\tau)}{p^2}\left(\operatorname{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 + \frac{2f''(\tau)}{p^2}\operatorname{tr}\left(\left(\mathbf{C}_2 - \mathbf{C}_1\right)^2\right)$$

$$\mathcal{V}_1^a = \frac{\left(f''(\tau)\right)^2}{p^4}\left(\operatorname{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 \operatorname{tr}\mathbf{C}_a^2$$

$$\mathcal{V}_2^a = \frac{2\left(f'(\tau)\right)^2}{p^2}\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)^{\mathsf{T}} \mathbf{C}_a \left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)$$

$$\mathcal{V}_3^a = \frac{2\left(f'(\tau)\right)^2}{np^2}\left(\frac{\operatorname{tr}\mathbf{C}_1\mathbf{C}_a}{c_1} + \frac{\operatorname{tr}\mathbf{C}_2\mathbf{C}_a}{c_2}\right)$$

More discussions

Several remarks:

1. imbalanced training data: $c_2 - c_1 \neq 0$ $\Rightarrow$ catastrophe!
2. $\mathfrak{D}$ as large as possible: conditions of signs of the kernel function f $\Rightarrow f'(\tau) < 0$ and $f''(\tau) > 0$
3. the influence of γ: $\Rightarrow$ (asymptotically) not important!

Theorem

$n\left(g(\mathbf{x}) - G_a\right) \xrightarrow{d} 0$ and $G_a \sim \mathcal{N}(\mathrm{E}_a, \mathrm{Var}_a)$ with

$$\mathrm{E}_a = \begin{cases} c_2 - c_1 - 2c_2 \cdot c_1 c_2 \gamma \mathfrak{D} \ , & a = 1 \\ c_2 - c_1 + 2c_1 \cdot c_1 c_2 \gamma \mathfrak{D} \ , & a = 2 \end{cases}$$

$$\mathrm{Var}_a = 8\gamma^2 c_1^2 c_2^2 \left(\mathcal{V}_1^a + \mathcal{V}_2^a + \mathcal{V}_3^a\right)$$

and

$$\mathfrak{D} = -\frac{2f'(\tau)}{p}\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2 + \frac{f''(\tau)}{p^2}\left(\mathrm{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 + \frac{2f''(\tau)}{p^2}\,\mathrm{tr}\left(\left(\mathbf{C}_2 - \mathbf{C}_1\right)^2\right)$$

$$\mathcal{V}_1^a = \frac{\left(f''(\tau)\right)^2}{p^4}\left(\mathrm{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2 \mathrm{tr}\mathbf{C}_a^2$$

$$\mathcal{V}_2^a = \frac{2\left(f'(\tau)\right)^2}{p^2}\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)^{\mathsf{T}} \mathbf{C}_a \left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)$$

$$\mathcal{V}_3^a = \frac{2\left(f'(\tau)\right)^2}{np^2}\left(\frac{\mathrm{tr}\,\mathbf{C}_1\mathbf{C}_a}{c_1} + \frac{\mathrm{tr}\,\mathbf{C}_2\mathbf{C}_a}{c_2}\right)$$

More discussions

Several remarks:

1. imbalanced training data: $c_2 - c_1 \neq 0$
   $\Rightarrow$ catastrophe!
2. $\mathfrak{D}$ as large as possible: conditions of signs of the kernel function f
   $\Rightarrow f'(\tau) < 0$ and $f''(\tau) > 0$
3. the influence of γ:
   $\Rightarrow$ (asymptotically) not important!
4. dominant difference in means
   $\Rightarrow$ even the choice of kernel irrelevant!

Theorem

$n\,(g(\mathbf{x}) - G_a) \xrightarrow{d} 0$ *and* $G_a \sim \mathcal{N}(\mathrm{E}_a, \mathrm{Var}_a)$ *with*

$$\mathrm{E}_a = \begin{cases} c_2 - c_1 - 2c_2 \cdot c_1 c_2 \gamma \mathfrak{D}\ , & a = 1 \\ c_2 - c_1 + 2c_1 \cdot c_1 c_2 \gamma \mathfrak{D}\ , & a = 2 \end{cases}$$

$$\mathrm{Var}_a = 8\gamma^2 c_1^2 c_2^2 \left(\mathcal{V}_1^a + \mathcal{V}_2^a + \mathcal{V}_3^a\right)$$

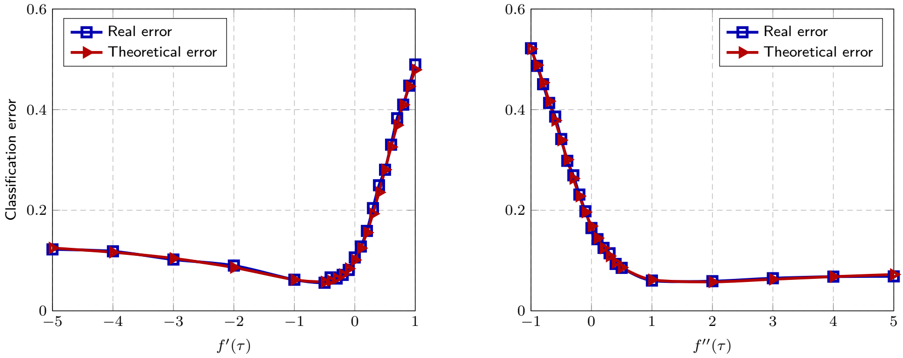
and

$$\mathfrak{D} = -\frac{2f'(\tau)}{p}\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2 + \frac{f''(\tau)}{p^2}\left(\mathrm{tr}\,(\mathbf{C}_2 - \mathbf{C}_1)\right)^2 + \frac{2f''(\tau)}{p^2}\,\mathrm{tr}\left((\mathbf{C}_2 - \mathbf{C}_1)^2\right)$$

$$\mathcal{V}_1^a = \frac{(f''(\tau))^2}{p^4}\left(\mathrm{tr}\,(\mathbf{C}_2 - \mathbf{C}_1)\right)^2 \mathrm{tr}\mathbf{C}_a^2$$

$$\mathcal{V}_2^a = \frac{2\,(f'(\tau))^2}{p^2}\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)^{\mathsf{T}} \mathbf{C}_a \left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)$$

$$\mathcal{V}_3^a = \frac{2\,(f'(\tau))^2}{np^2}\left(\frac{\mathrm{tr}\,\mathbf{C}_1\mathbf{C}_a}{c_1} + \frac{\mathrm{tr}\,\mathbf{C}_2\mathbf{C}_a}{c_2}\right)$$



(a) $f(\tau) = 4, \; f'' = 1$

(b) $f(\tau) = 4, \; f'(\tau) = -1$

Figure: Performance of LS-SVM, $n = 256, p = 512, c_1 = c_2 = 1/2, \gamma = 1$, polynomial kernel. $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$, with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{4}{\sqrt{p}})$.

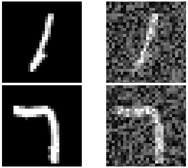
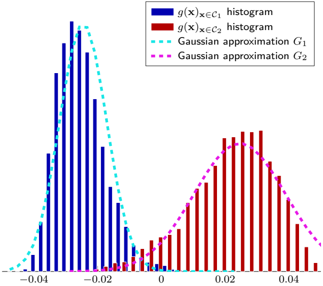
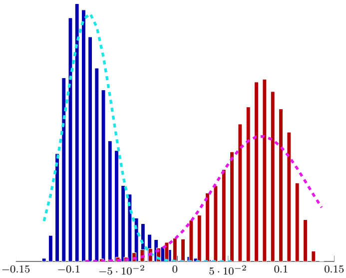


Figure: Samples from the MNIST database, without and with 0dB noise.



(a) with 0dB noise



(b) without noise

Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 784$, $c_1 = c_2 = 1/2$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, MNIST data (numbers 1 and 7) without and with 0dB noise.

Simulations on MNIST data: influence of γ

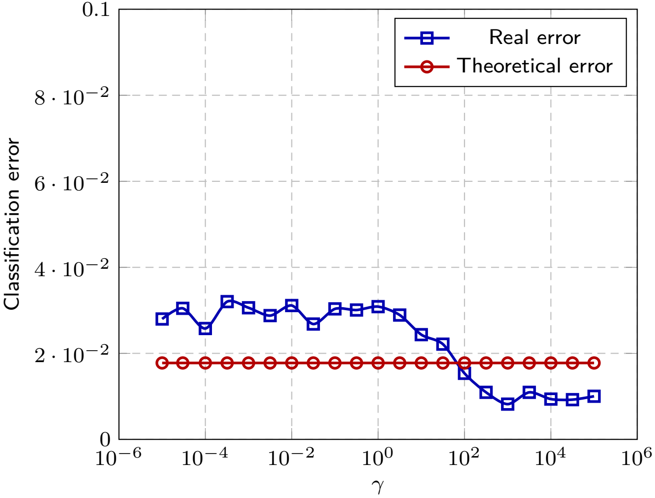


Figure: Performance of LS-SVM, $n = 256, p = 784$, $c_1 = c_2 = \frac{1}{2}$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, MNIST data (ones and sevens).

Kernel evaluation for MNIST data

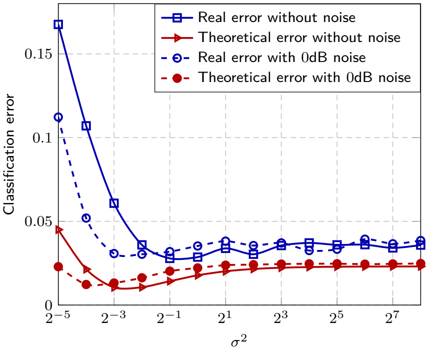


Figure: Performance of LS-SVM, $n = 256, p = 784$, $c_1 = c_2 = \frac{1}{2}, \gamma = 1$, Gaussian kernel, MNIST data with & without noise.

Kernel evaluation for MNIST data

Table: Empirical estimation of (normalized) differences in means and covariances of MNIST data.

	Without noise	With 0dB noise
$\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_2\|^2$	429	178
$\left(\operatorname{tr}\left(\mathbf{C}_2 - \mathbf{C}_1\right)\right)^2/p$	63	11
$\operatorname{tr}\left((\mathbf{C}_2 - \mathbf{C}_1)^2\right)/p$	35	6

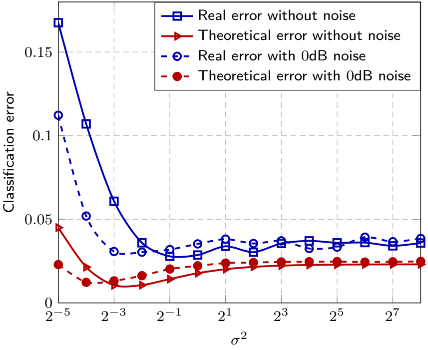
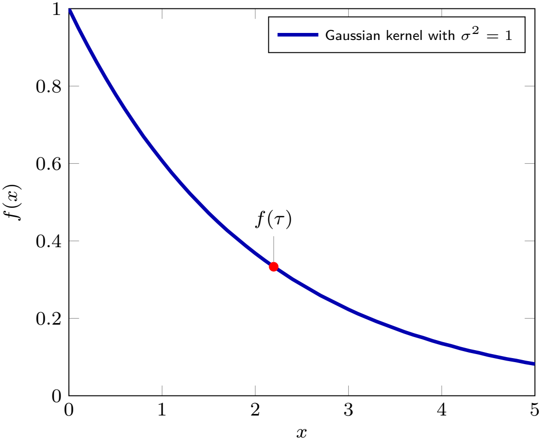
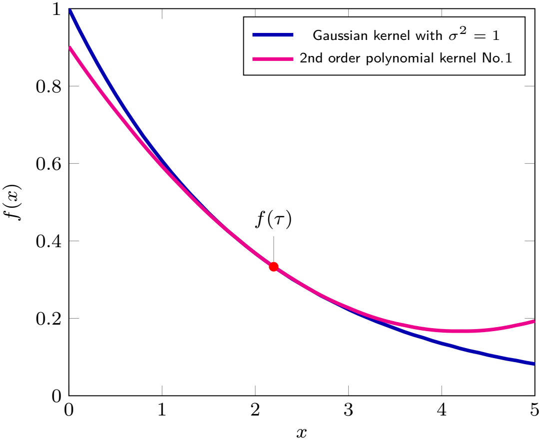


Figure: Performance of LS-SVM, $n = 256, p = 784$, $c_1 = c_2 = \frac{1}{2}, \gamma = 1$, Gaussian kernel, MNIST data with & without noise.

Kernel comparison for Gaussian data (1)

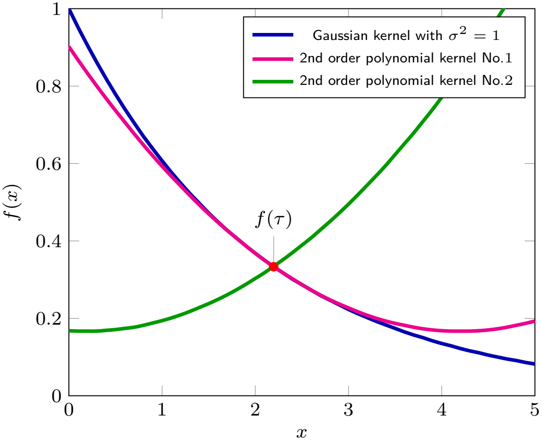


Kernel comparison for Gaussian data (1)



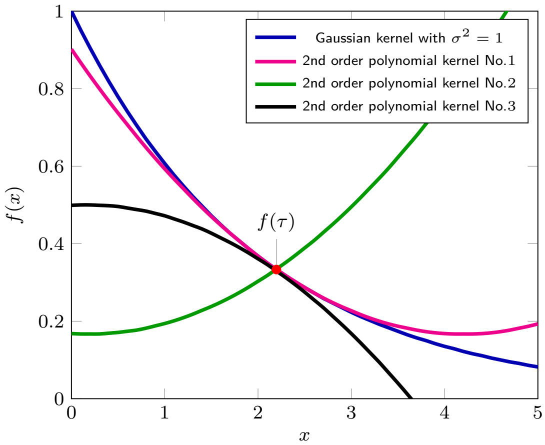
- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.

Kernel comparison for Gaussian data (1)



- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.

Kernel comparison for Gaussian data (1)



- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

Kernel comparison for Gaussian data (2)

Table: Performance of different kernels [1]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	91.4%
2nd-order polynomial kernel No.1	
2nd-order polynomial kernel No.2	
2nd-order polynomial kernel No.3	

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[1] Gaussian mixture data with $\boldsymbol{\mu}_a = \left[\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}\right]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{4}{\sqrt{p}})$. $n_{\text{test}} = n = 256, p = 512$, $\gamma = 1$.

Kernel comparison for Gaussian data (2)

Table: Performance of different kernels [1]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	91.4%
2nd-order polynomial kernel No.1	91.2%
2nd-order polynomial kernel No.2	
2nd-order polynomial kernel No.3	

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[1] Gaussian mixture data with $\boldsymbol{\mu}_a = \left[\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}\right]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{4}{\sqrt{p}})$. $n_{\text{test}} = n = 256$, $p = 512$, $\gamma = 1$.

Kernel comparison for Gaussian data (2)

Table: Performance of different kernels [1]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	91.4%
2nd-order polynomial kernel No.1	91.2%
2nd-order polynomial kernel No.2	33.6%
2nd-order polynomial kernel No.3	

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[1] Gaussian mixture data with $\boldsymbol{\mu}_a = \left[\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}\right]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{4}{\sqrt{p}})$. $n_{\text{test}} = n = 256, p = 512, \gamma = 1$.

Kernel comparison for Gaussian data (2)

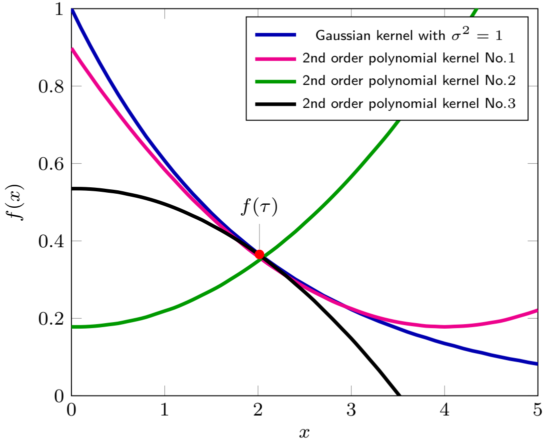
Table: Performance of different kernels [1]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	91.4%
2nd-order polynomial kernel No.1	91.2%
2nd-order polynomial kernel No.2	33.6%
2nd-order polynomial kernel No.3	67.1%

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[1] Gaussian mixture data with $\boldsymbol{\mu}_a = \left[\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}\right]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{4}{\sqrt{p}})$. $n_{\text{test}} = n = 256$, $p = 512$, $\gamma = 1$.

Kernel comparison for MNIST data (1)



- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

Kernel comparison for MNIST data (2)

Table: Performance of different kernels[2]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	97.1%
2nd-order polynomial kernel No.1	
2nd-order polynomial kernel No.2	
2nd-order polynomial kernel No.3	

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[2] MNIST data (number 1 and 7), $n_{\text{test}} = n = 256$, $p = 784$, $\gamma = 1$.

Kernel comparison for MNIST data (2)

Table: Performance of different kernels[2]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	97.1%
2nd-order polynomial kernel No.1	97.3%
2nd-order polynomial kernel No.2	
2nd-order polynomial kernel No.3	

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[2]MNIST data (number 1 and 7), $n_{\text{test}} = n = 256, p = 784$, $\gamma = 1$.

Kernel comparison for MNIST data (2)

Table: Performance of different kernels[2]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	97.1%
2nd-order polynomial kernel No.1	97.3%
2nd-order polynomial kernel No.2	4.9%
2nd-order polynomial kernel No.3	

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[2] MNIST data (number 1 and 7), $n_{\text{test}} = n = 256, p = 784$, $\gamma = 1$.

Kernel comparison for MNIST data (2)

Table: Performance of different kernels[2]

	Classification success rate
Gaussian kernel with $\sigma^2 = 1$	97.1%
2nd-order polynomial kernel No.1	97.3%
2nd-order polynomial kernel No.2	4.9%
2nd-order polynomial kernel No.3	95.0%

- No.1: same $f(\tau)$, $f'(\tau)$, $f''(\tau)$ as Gaussian kernel.
- No.2: same $f(\tau)$ and $f''(\tau)$, while $f'(\tau)$ of opposite sign.
- No.3: same $f(\tau)$ and $f'(\tau)$, while $f''(\tau)$ of opposite sign.

[2] MNIST data (number 1 and 7), $n_{\text{test}} = n = 256$, $p = 784$, $\gamma = 1$.

Outline

Summary

Take-away messages:

- Balanced training data

Summary

Take-away messages:

- Balanced training data
- User-defined kernel function for higher-order information (e.g., covariance) in difficult tasks

Summary

Take-away messages:

- Balanced training data
- User-defined kernel function for higher-order information (e.g., covariance) in difficult tasks
- Sometimes tuning parameters can be (asymptotically) trivial

Summary

Take-away messages:

- Balanced training data
- User-defined kernel function for higher-order information (e.g., covariance) in difficult tasks
- Sometimes tuning parameters can be (asymptotically) trivial

Summary

Take-away messages:

- Balanced training data
- User-defined kernel function for higher-order information (e.g., covariance) in difficult tasks
- Sometimes tuning parameters can be (asymptotically) trivial

Future work:

- Extension to SVM: more complex structure from the implicit form

[3] a random-connected single-layer feed-forward network.

Summary

Take-away messages:

- Balanced training data
- User-defined kernel function for higher-order information (e.g., covariance) in difficult tasks
- Sometimes tuning parameters can be (asymptotically) trivial

Future work:

- Extension to SVM: more complex structure from the implicit form
- link to Neural Networks: through Extreme Learning Machine[3]

[3] a random-connected single-layer feed-forward network.

Summary

Take-away messages:

- Balanced training data
- User-defined kernel function for higher-order information (e.g., covariance) in difficult tasks
- Sometimes tuning parameters can be (asymptotically) trivial

Future work:

- Extension to SVM: more complex structure from the implicit form
- link to Neural Networks: through Extreme Learning Machine[3]
- A new way of understanding the magic of machine learning methods

[3] a random-connected single-layer feed-forward network.

Thank you!