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

ICASSP'17

Zhenyu Liao, Romain Couillet

CentraleSupélec
Université Paris-Saclay
Paris, France

CentraleSupélec

Outline

Outline

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

In addition:

- results only available for *number of data* $n \to \infty$ [Van Gestel et al. 2002]

Motivation

Performance analysis of SVM difficult:

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

In addition:

- results only available for *number of data* $n \to \infty$ [Van Gestel et al. 2002]
- no prediction so far when *dimension of data* $p \sim n$

Motivation

Performance analysis of SVM difficult:

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

In addition:

- results only available for *number of data* $n \to \infty$ [Van Gestel et al. 2002]
- no prediction so far when *dimension of data* $p \sim n$
- when $n, p \to \infty$, completely different behavior of kernels

Motivation

Performance analysis of SVM difficult:

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

In addition:

- results only available for *number of data* $n \to \infty$ [Van Gestel et al. 2002]
- no prediction so far when *dimension of data* $p \sim n$
- when $n, p \to \infty$, completely different behavior of kernels

$\Rightarrow$ SVM for BigData not understood

Motivation

Performance analysis of SVM difficult:

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

In addition:

- results only available for *number of data* $n \to \infty$ [Van Gestel et al. 2002]
- no prediction so far when *dimension of data* $p \sim n$
- when $n, p \to \infty$, completely different behavior of kernels

$\Rightarrow$ SVM for BigData not understood

In this work:

- new random matrix approach to linearize kernels

Motivation

Performance analysis of SVM difficult:

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

In addition:

- results only available for *number of data* $n \to \infty$ [Van Gestel et al. 2002]
- no prediction so far when *dimension of data* $p \sim n$
- when $n, p \to \infty$, completely different behavior of kernels

$\Rightarrow$ SVM for BigData not understood

In this work:

- new random matrix approach to linearize kernels
- asymptotic analysis of LS-SVM for $n, p \to \infty$

Motivation

Performance analysis of SVM difficult:

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

In addition:

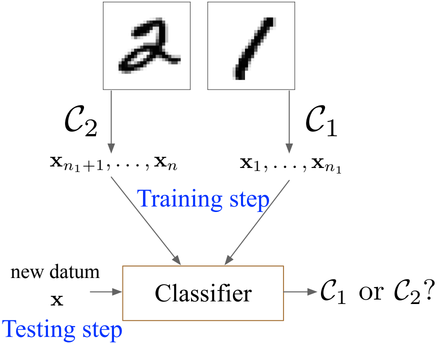
- results only available for *number of data* $n \to \infty$ [Van Gestel et al. 2002]
- no prediction so far when *dimension of data* $p \sim n$
- when $n, p \to \infty$, completely different behavior of kernels

$\Rightarrow$ SVM for BigData not understood

In this work:

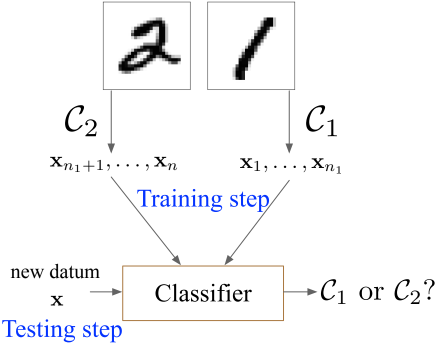
- new random matrix approach to linearize kernels
- asymptotic analysis of LS-SVM for $n, p \to \infty$
- new insights

Outline



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

Least Squares Support Vector Machines (1)

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

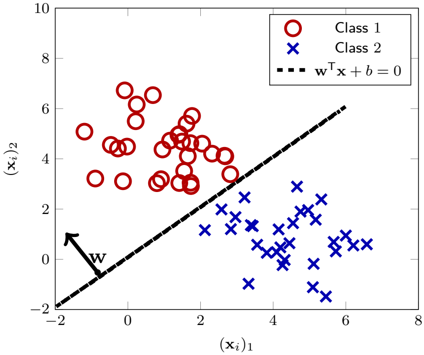
Least Squares Support Vector Machines (1)

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

Optimization problem: find separating hyperplane

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

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



Least Squares Support Vector Machines (2)

When no linear separability:
$\Rightarrow$ Kernel method

Least Squares Support Vector Machines (2)

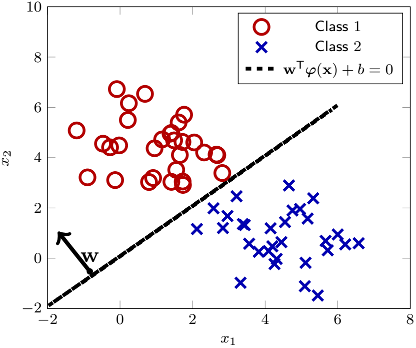
When no linear separability:
⇒ Kernel method

To solve the optimization problem:

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

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

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



Least Squares Support Vector Machines (3)

- **Training**: Solution given by $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\varphi}(\mathbf{x}_i)$, where

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

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

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

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 $\mathbf{w} = \sum_{i=1}^n \alpha_i \boldsymbol{\varphi}(\mathbf{x}_i)$, where

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

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

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

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

- **bing**: Decision for new $\mathbf{x}$

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

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

Least Squares Support Vector Machines (3)

- **Training**: Solution given by $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\varphi}(\mathbf{x}_i)$, where

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

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

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

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

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

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

Least Squares Support Vector Machines (3)

- **Training**: Solution given by $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\varphi}(\mathbf{x}_i)$, where

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

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

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

  for some *translation invariant* kernel function $f : \mathbb{R}_+ \mapsto \mathbb{R}_+$, $\mathbf{y} \equiv [y_1, \ldots, y_n]^\mathsf{T}$ and $\boldsymbol{\alpha} \equiv [\alpha_1, \ldots, \alpha_n]^\mathsf{T}$.
- **bing**: Decision for 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 $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\varphi}(\mathbf{x}_i)$, where

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

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

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

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

- **bing**: Decision for 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, as opposed to SVM $\Rightarrow$ easier to analyze.

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 $a \in \{1, 2\}$:

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

Asymptotic Regime: Growth Rate Assumptions

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

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

- **Non-trivial regime**: to ensure $P(\mathbf{x}_i \to \mathcal{C}_b \mid \mathbf{x}_i \in \mathcal{C}_a) \not\to 0$ nor 1

Asymptotic Regime: Growth Rate Assumptions

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

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

- **Non-trivial regime**: to ensure $P(\mathbf{x}_i \to \mathcal{C}_b \mid \mathbf{x}_i \in \mathcal{C}_a) \not\to 0$ nor 1
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$

Asymptotic Regime: Growth Rate Assumptions

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

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

- **Non-trivial regime**: to ensure $P(\mathbf{x}_i \to \mathcal{C}_b \mid \mathbf{x}_i \in \mathcal{C}_a) \not\to 0$ nor 1
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ 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 $a \in \{1, 2\}$:

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

- **Non-trivial regime**: to ensure $P(\mathbf{x}_i \to \mathcal{C}_b \mid \mathbf{x}_i \in \mathcal{C}_a) \not\to 0$ nor 1
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$

  $\Rightarrow$ If relaxed, perfect classification from $\|\mathbf{x}_i\|$

Asymptotic Regime: Growth Rate Assumptions

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

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

- **Non-trivial regime**: to ensure $P(\mathbf{x}_i \to \mathcal{C}_b \mid \mathbf{x}_i \in \mathcal{C}_a) \not\to 0$ nor 1
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$

  $\Rightarrow$ If relaxed, perfect classification from $\|\mathbf{x}_i\|$
- **Notations**:

Asymptotic Regime: Growth Rate Assumptions

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

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

- **Non-trivial regime**: to ensure $P(\mathbf{x}_i \to \mathcal{C}_b \mid \mathbf{x}_i \in \mathcal{C}_a) \not\to 0$ nor 1
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$

  $\Rightarrow$ If relaxed, perfect classification from $\|\mathbf{x}_i\|$
- **Notations**:
  - $\mathbf{C}^\circ \equiv c_1\mathbf{C}_1 + c_2\mathbf{C}_2$, $c_1 \equiv \frac{n_1}{n}$ 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 $a \in \{1, 2\}$:

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

- **Non-trivial regime**: to ensure $P(\mathbf{x}_i \to \mathcal{C}_b \mid \mathbf{x}_i \in \mathcal{C}_a) \not\to 0$ nor 1
  - $\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\| = O(1)$
  - $\|\mathbf{C}_a\| = O(1)$ and $\operatorname{tr}(\mathbf{C}_2 - \mathbf{C}_1) = O(\sqrt{n})$

  $\Rightarrow$ If relaxed, perfect classification from $\|\mathbf{x}_i\|$
- **Notations**:
  - $\mathbf{C}^\circ \equiv c_1 \mathbf{C}_1 + c_2 \mathbf{C}_2$, $c_1 \equiv \frac{n_1}{n}$ and $c_2 \equiv \frac{n_2}{n} = 1 - c_1$
  - **Key Notation**: $\tau \equiv \frac{2}{p} \operatorname{tr} C^\circ$

Kernel linearization (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 $\mathbf{x}_i = \boldsymbol{\mu}_a + \mathbf{w}_i$ where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a)$

Kernel linearization (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 $\mathbf{x}_i = \boldsymbol{\mu}_a + \mathbf{w}_i$ where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a)$

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

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

Kernel linearization (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 $\mathbf{x}_i = \boldsymbol{\mu}_a + \mathbf{w}_i$ where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a)$

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

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

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

Kernel linearization (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 $\mathbf{x}_i = \boldsymbol{\mu}_a + \mathbf{w}_i$ where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a)$

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

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

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

$$= \frac{1}{p}\operatorname{tr}\mathbf{C}_a + \frac{1}{p}\operatorname{tr}\mathbf{C}_b + O\left(\frac{1}{\sqrt{n}}\right) = \underbrace{\frac{2}{p}\operatorname{tr}\mathbf{C}^\circ}_{\equiv\tau=O(1)} + \underbrace{\frac{1}{p}\operatorname{tr}(\mathbf{C}_a - \mathbf{C}^\circ) + \frac{1}{p}\operatorname{tr}(\mathbf{C}_b - \mathbf{C}^\circ)}_{O(n^{-1/2})} + O\left(\frac{1}{\sqrt{n}}\right)$$

Kernel linearization (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 $\mathbf{x}_i = \boldsymbol{\mu}_a + \mathbf{w}_i$ where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_a)$

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

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

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

$$= \frac{1}{p}\operatorname{tr}\mathbf{C}_a + \frac{1}{p}\operatorname{tr}\mathbf{C}_b + O(\frac{1}{\sqrt{n}}) = \underbrace{\frac{2}{p}\operatorname{tr}\mathbf{C}^\circ}_{\equiv\tau=O(1)} + \underbrace{\frac{1}{p}\operatorname{tr}(\mathbf{C}_a - \mathbf{C}^\circ) + \frac{1}{p}\operatorname{tr}(\mathbf{C}_b - \mathbf{C}^\circ)}_{O(n^{-1/2})} + O(\frac{1}{\sqrt{n}})$$

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

Kernel linearization (2)

Recall: kernel 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 $\mathbf{K}_{i,j}$

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

or in matrix form

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

Kernel linearization (2)

Recall: kernel 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 $\mathbf{K}_{i,j}$

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

or in matrix form

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

Non trivial RMT calculus: $\mathbf{A}_{ij} \to 0 \not\Leftrightarrow \|\mathbf{A}\| \to 0$

Kernel linearization (2)

Recall: kernel 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 $\mathbf{K}_{i,j}$

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

or in matrix form

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

Non trivial RMT calculus: $\mathbf{A}_{ij} \to 0 \not\Leftrightarrow \|\mathbf{A}\| \to 0$

Consequence

Asymptotic statistics of $\mathbf{K}$, thus of

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

Asymptotic Behavior of the Decision Function

Theorem

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

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

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

Asymptotic Behavior of the Decision Function

Theorem

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

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

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

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

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

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

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

Simulations on Gaussian data

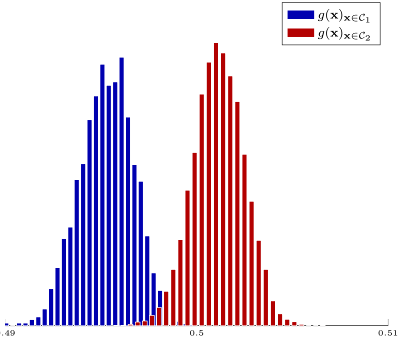


Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 512$, $c_1 = 1/4, c_2 = 3/4$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$ with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 3; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$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

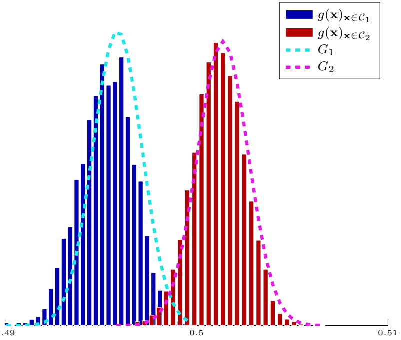


Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 512$, $c_1 = 1/4, c_2 = 3/4$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$ with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 3; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$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

Simulations on Gaussian data

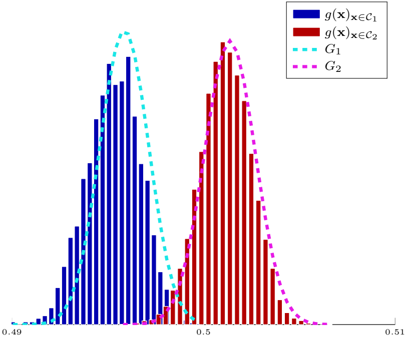


Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 512$, $c_1 = 1/4, c_2 = 3/4$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$ with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 3; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$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

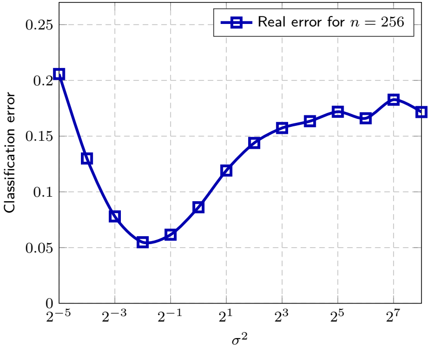


Figure: Performance of LS-SVM, $c_0 = 2$, $c_1 = c_2 = 1/2$, $\gamma = 1$, Gaussian kernel $f(x) = \exp(\frac{x}{2\sigma^2})$. $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$, with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j}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

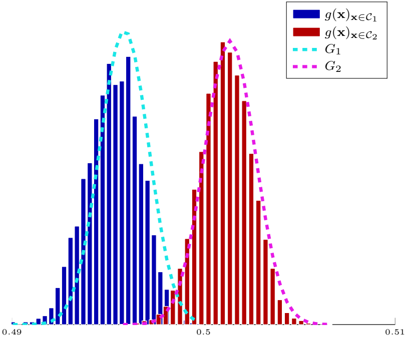


Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 512$, $c_1 = 1/4, c_2 = 3/4$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$ with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 3; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$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

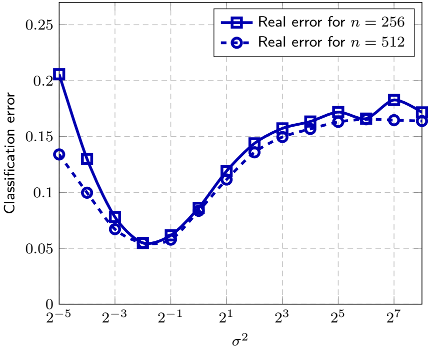


Figure: Performance of LS-SVM, $c_0 = 2$, $c_1 = c_2 = 1/2$, $\gamma = 1$, Gaussian kernel $f(x) = \exp(\frac{x}{2\sigma^2})$. $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$, with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j}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

Simulations on Gaussian data

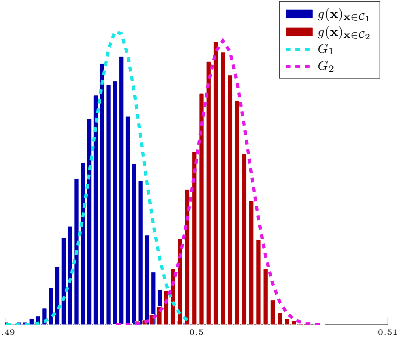


Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 512$, $c_1 = 1/4, c_2 = 3/4$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$ with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 3; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$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

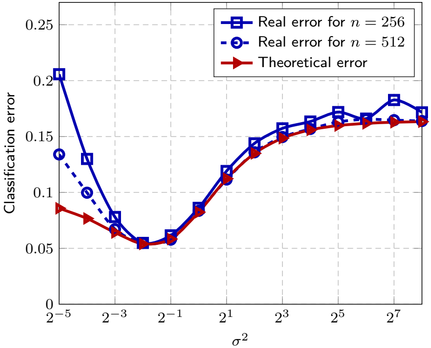


Figure: Performance of LS-SVM, $c_0 = 2$, $c_1 = c_2 = 1/2$, $\gamma = 1$, Gaussian kernel $f(x) = \exp(\frac{x}{2\sigma^2})$. $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}_a, \mathbf{C}_a)$, with $\boldsymbol{\mu}_a = [\mathbf{0}_{a-1}; 2; \mathbf{0}_{p-a}]$, $\mathbf{C}_1 = \mathbf{I}_p$ and $\{\mathbf{C}_2\}_{i,j}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

Simulations on MNIST data

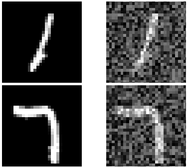


Figure: Samples from the MNIST database, without and with 0dB noise.

Simulations on MNIST data

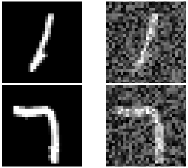


Figure: Samples from the MNIST database, without and with 0dB noise.

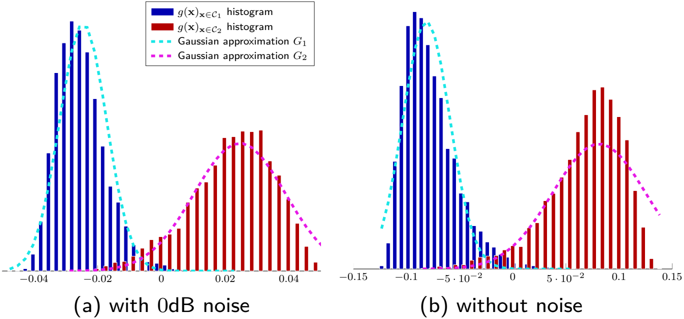


Figure: Gaussian approximation of $g(\mathbf{x})$, $n = 256, p = 784$, $c_1 = c_2 = 1/2$, $\gamma = 1$, Gaussian kernel with $\sigma^2 = 1$, MNIST data (numbers 1 and 7) without and with 0dB noise.

Discussion

Some consequences:

1. imbalanced training data: $c_2 - c_1 \neq 0$
   $\Rightarrow$ Decision boundary $c_2 - c_1$ instead of 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

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

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

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

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

Discussion

Some consequences:

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

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

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

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

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

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

Discussion

Some consequences:

1. imbalanced training data: $c_2 - c_1 \neq 0$ $\Rightarrow$ Decision boundary $c_2 - c_1$ instead of 0!
2. $\mathfrak{D}$ as large as possible: conditions of f $\Rightarrow f'(\tau) < 0$ and $f''(\tau) > 0$
3. 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

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

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

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

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

Discussion

Some consequences:

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

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

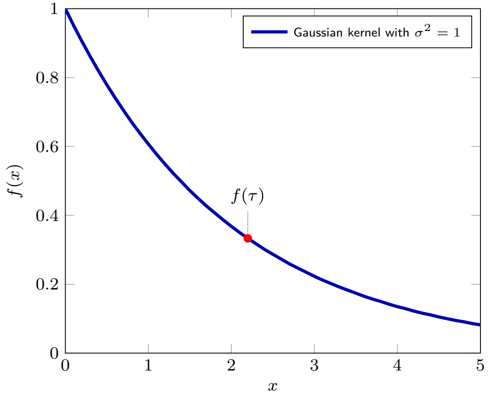
$$\mathfrak{D} = -\frac{2f'(\tau)}{p}\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2 + \frac{f''(\tau)}{p^2}\left(\mathrm{tr}\,(\mathbf{C}_2 - \mathbf{C}_1)\right)^2 + \frac{2f''(\tau)}{p^2}\,\mathrm{tr}\left((\mathbf{C}_2 - \mathbf{C}_1)^2\right)$$

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

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

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

Kernel comparison[1]



Recall: kernel matrix

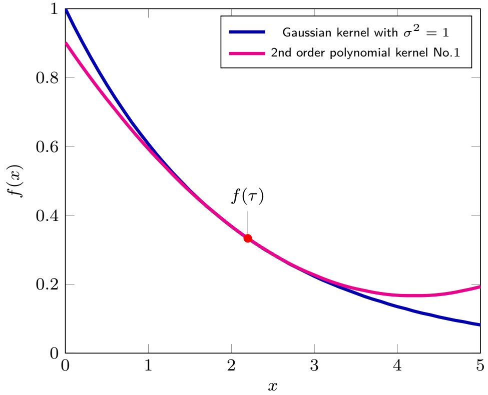
$$\mathbf{K}_{i,j} = f\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{p}\right)$$

Table: Performance of different kernels

Kernel	Success rate
▬	91.4%

[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[1]



Recall: kernel matrix

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

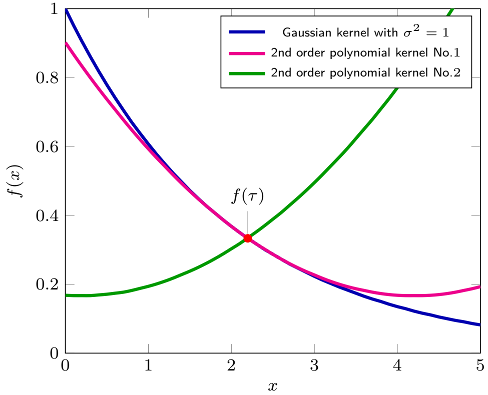
Table: Performance of different kernels

Kernel	Success rate
(blue: Gaussian)	91.4%
(magenta: polynomial No.1)	91.2%

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

[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[1]



Recall: kernel matrix

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

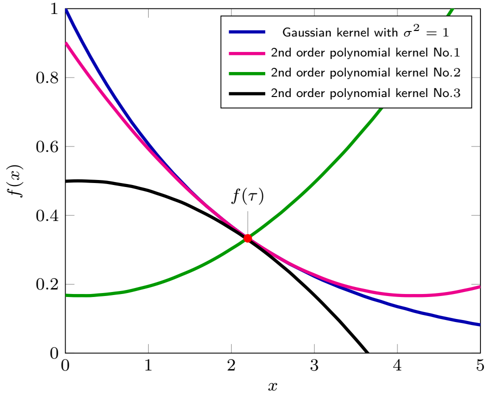
Table: Performance of different kernels

Kernel	Success rate
blue (Gaussian)	91.4%
magenta (No.1)	91.2%
green (No.2)	33.6%

- 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.

[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[1]



Recall: kernel matrix

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

Table: Performance of different kernels

Kernel	Success rate
(blue)	91.4%
(magenta)	91.2%
(green)	33.6%
(black)	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 $\{\mathbf{C}_2\}_{i,j} = .4^{|i-j|}(1 + \frac{4}{\sqrt{p}})$. $n_{\text{test}} = n = 256$, $p = 512$, $\gamma = 1$.

Outline

Summary

Take-away messages:

- New random matrix framework for SVM analysis

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

$\Rightarrow$ Key parameters are $f^{(k)}(\tau)$, not σ! (of Gaussian kernel)

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

$\Rightarrow$ Key parameters are $f^{(k)}(\tau)$, not σ! (of Gaussian kernel)

- Allows for analysis of other kernel methods: kernel PCA, clustering, etc

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

$\Rightarrow$ Key parameters are $f^{(k)}(\tau)$, not σ! (of Gaussian kernel)

- Allows for analysis of other kernel methods: kernel PCA, clustering, etc

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

$\Rightarrow$ Key parameters are $f^{(k)}(\tau)$, not σ! (of Gaussian kernel)

- Allows for analysis of other kernel methods: kernel PCA, clustering, etc

Future work:

- Extension to SVM: difficulty due to implicit formulation

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

$\Rightarrow$ Key parameters are $f^{(k)}(\tau)$, not σ! (of Gaussian kernel)

- Allows for analysis of other kernel methods: kernel PCA, clustering, etc

Future work:

- Extension to SVM: difficulty due to implicit formulation
- Possible extension beyond kernels: neural networks (shallow, deep, recurrent...)

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

$\Rightarrow$ Key parameters are $f^{(k)}(\tau)$, not σ! (of Gaussian kernel)

- Allows for analysis of other kernel methods: kernel PCA, clustering, etc

Future work:

- Extension to SVM: difficulty due to implicit formulation
- Possible extension beyond kernels: neural networks (shallow, deep, recurrent...)

Summary

Take-away messages:

- New random matrix framework for SVM analysis
- Kernel with same $f(\tau), f'(\tau), f''(\tau)$ asymptotically equivalent

$\Rightarrow$ Key parameters are $f^{(k)}(\tau)$, not σ! (of Gaussian kernel)

- Allows for analysis of other kernel methods: kernel PCA, clustering, etc

Future work:

- Extension to SVM: difficulty due to implicit formulation
- Possible extension beyond kernels: neural networks (shallow, deep, recurrent...)

References:

- Z. Liao, R. Couillet, "**A Large Dimensional Analysis of Least Squares Support Vector Machines**", (submitted to) Journal of Machine Learning Research, 2016.
- C. Louart, Z. Liao, R. Couillet, "**A Random Matrix Approach to Neural Networks**", (submitted to) Annals of Applied Probability, 2017.

Thank you!