IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on YYYY–MM–DD.
Submitted on YYYY–MM–DD, accepted on YYYY–MM–DD.
ISSN 2105–1232
This article is available online with supplementary materials, software, datasets and online demo at
https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000527

Double Descent in the case of Multilayer Perceptron

Emett Haddad

ENS Paris-Saclay, 91190, Gif-Sur-Yvette, France
emett.haddad@ens-paris-saclay.fr

PREPRINT June 10, 2024

Abstract

We consider $y^\star : x \in \mathcal{X} = \mathbb{R}^D \to \mathbb{E}_{(x,y)\sim\mathcal{P}}(y|x) \in \mathcal{Y} = \mathbb{R}$, with x and y random variable with joint distribution $(x, y) \hookrightarrow \mathcal{P}$. We have a **dataset** $\mathcal{D} := \{(x_n, y_n) \in \mathcal{X} \times \mathcal{Y}\}_{n=1}^N$ where $y_n := y(x_n)$, and $y = y^\star + \epsilon$ where ϵ represents the noise as $\mathbb{E}(\epsilon|x) = 0$, et $\mathbb{V}(\epsilon) = \sigma^2$. We want to create an estimator $\hat{y}_\mathcal{D} : \mathcal{X} \to \mathcal{Y}$ which minimises the empirical risk in order to minimise the true risk $\mathcal{R}(\hat{y}) = \mathbb{E}_{(x,y)\sim\mathcal{P}}((y - \hat{y}(x))^2)$. Working in the context of [1]

Source Code

The reviewed source code and documentation for this algorithm are available from the web page of this article[1]. Usage instruction are included in the README.txt file of the archive.

1 Model

- $\mathcal{F} := \{f_p, 1 \leq p \leq P\}$ a feature set.
- $X = [x_1, \cdots, x_N]^T \in \mathcal{M}_{N,D}(\mathbb{R})$, $Y = [y_1, \cdots y_N]^T$
- $\Phi_P(x) = [f_1(x), \cdots, f_P(x)]^T$ and $Z_P = [\Phi_P^T(x_1), \cdots, \Phi_P^T(x_N)] = [f_j(x_i)] \in \mathcal{M}_{N,P}(\mathbb{R})$
- $\hat{y}_\mathcal{D}(x) = \Phi_P^T(x)\hat{\beta}_P$, with $\hat{\beta}_P = \begin{bmatrix} Z_P \\ \sqrt{N\lambda} I_P \end{bmatrix}^\dagger \begin{bmatrix} Y \\ O_P \end{bmatrix}$
- Loss function $\mathcal{L}(y) = \frac{1}{N}||Y - Z\beta||_2^2 + \lambda||\beta||_2^2$ with $\lambda \geq 0$

[1] https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000527

2 Algorithms

Algorithm 1: Generation of the orthonormal polynomial basis (Gram- Schmidt)

function generate_orthonormal_basis(D, Deg, C)

Input D, Deg, C**:** $D \in \mathbb{N}^\star$,$Deg \in \mathbb{N}^\star$, $C \subset \mathbb{R}^D$

$\mathrm{Basis_{D'}} = [\Pi_{\mathrm{d=1}}^{\mathrm{D}} \mathrm{X_d^{\alpha_d}}, \sum \alpha_\mathrm{d} \leq \mathrm{D'}]$

$\mathrm{P = LENGHT(Basis_{Deg})}$

for $1 \leq p \leq P$ **do**

$f'_p = \mathrm{Basis_{Deg}[p]} - \sum_{\mathrm{i=1}}^{\mathrm{p-1}} [\int_{\mathrm{C}} \mathrm{Basis\_ortho_{Deg}[i]} \cdot \mathrm{Basis_{Deg}[i]}] \times \mathrm{Basis\_ortho_{Deg}[i]}$

$\mathrm{Basis\_ortho_{Deg}[p]} = \frac{\mathrm{f'_p}}{||\mathrm{f'_p}||_2}$

return $\mathrm{Basis\_ortho_{Deg}}$

Algorithm 2: Dataset initialisation

function dataset_initialisation$(f, C, M, ratio_data)$

Input y, C, M, ratio_data**:** $y : \mathbb{R}^D \to \mathbb{R}$, $C = [[a_d, b_d], 1 \leq d \leq D]$, ratio_data $\in [0, 1]$

$U = \mathcal{U}([0,1]^{(M,D)})$

$X = Diag(a_d) + UDiag(b_d - a_d)$ # $X \hookrightarrow \mathcal{U}(C^M)$

$Y = y(X)$

$X_{train}, X_{test}, Y_{train}, Y_{test} = \mathrm{test\_split}(X, Y, \mathrm{ratio\_data})$

return $\mathrm{X_{train}, X_{test}, Y_{train}, Y_{test}}$

Algorithm 3: Ridge Regression

function ridge_regression($\mathrm{P_{min}, P_{max}}$, Features, λ, Data, Data_global)

$\mathrm{X_{train}, X_{test}, Y_{train}, Y_{test}} = \mathrm{Data}$

$\mathrm{X_{global}, Y_{global}} = \mathrm{Data\_global}$

for $P_{min} \leq p \leq P_{max}$ **do**

$\Phi_p = [f_p, 1 \leq p \leq P]^T$ # (f_p) an orthonormal basis for p.s. on C

$Z_p = \Phi_p^T(X_{train})$ # $Z_p \in \mathcal{M}_{N,P}(\mathbb{R})$

$$\hat{\beta}_p = \begin{bmatrix} Z_p \\ \sqrt{N\lambda I_p} \end{bmatrix}^\dagger \begin{bmatrix} Y_{train} \\ O_p \end{bmatrix}$$

$\mathrm{Train\_error[p]} = \log(1 + \mathrm{MSE(Y_{train}, Z_p \hat{\beta}_p)})$

$\mathrm{Test\_error[p]} = \log(1 + \mathrm{MSE(Y_{test}, \Phi_p^T(X_{test})\hat{\beta}_p)})$

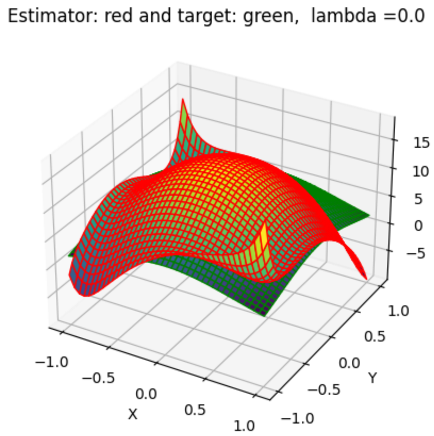
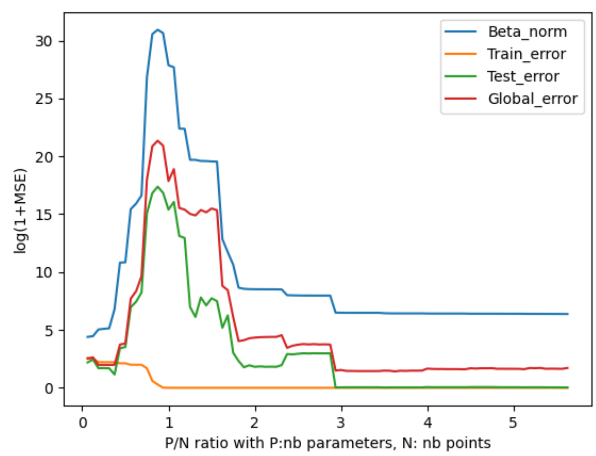
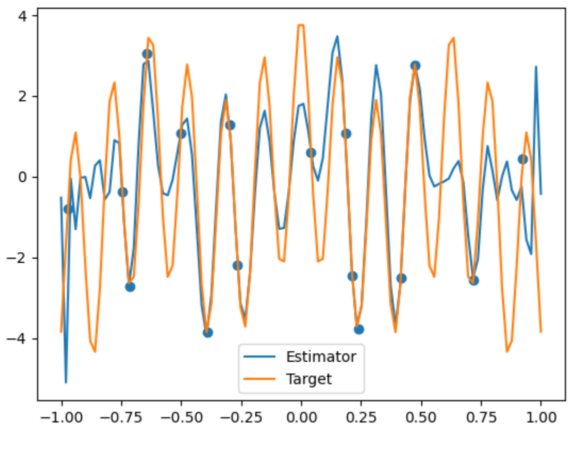
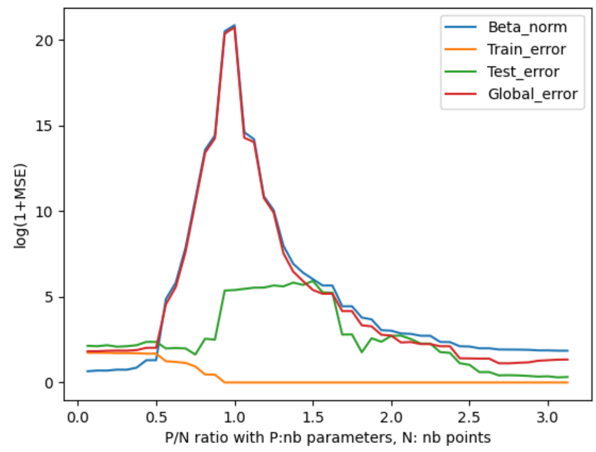
$\mathrm{Global\_error[p]} = \log(1 + \mathrm{MSE(Y_{global}, \Phi_p^T(X_{global})\hat{\beta}_p)})$

$\mathrm{Beta\_norm[p]} = \log(1 + ||\hat{\beta}_\mathrm{p}||_2^2)$

$\hat{y} = \Phi_p \cdot \hat{\beta}_{P_{max}}$

return Train_error, Test_error, Beta_norm, Global_error, $\hat{\mathrm{y}}$

3 Examples (random seed : 23334)

Function plot	Graph error
 $y(x) = 2e^{2-(x_1^2+x_2^2)}, Basis : cano$	 $M = 20, r = 0.2, n = 40, a = 1, Deg = 12, D = 2, \lambda = 0$
 $y(x) = 3cos(40x) + cos(10x) - x^2, Basis : ortho$	 $M = 20, r = 0.2, n = 100, a = 1, Deg = 50, D = 1, \lambda = 0$

References

[1] Emett Haddad, *Github Double Descent*. https://github.com/EmettGabrielH/Double-descent---Emett-Haddad, 2024. [Online].