



Published in Image Processing On Line on YYYY-MM-DD.
 Submitted on YYYY-MM-DD, accepted on YYYY-MM-DD.
 ISSN 2105-1232 © YYYY IPOL & the authors CC-BY-NC-SA
 This article is available online with supplementary materials,
 software, datasets and online demo at
<https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000515>

Double Descent in the case of ridge regression

Emett Haddad

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

PREPRINT June 11, 2024

Abstract

We consider $y^* : x \in \mathcal{X} = \mathbb{R}^D \rightarrow \mathbb{E}_{(x,y) \sim \mathcal{P}}(y|x) \in \mathcal{Y} = \mathbb{R}$, 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 $y = y^* + \epsilon$ where ϵ represents the noise as $\mathbb{E}(\epsilon|x) = 0$, et $\mathbb{V}(\epsilon) = \sigma^2$. We want to create an estimator $\hat{y}_{\mathcal{D}} : \mathcal{X} \rightarrow \mathcal{Y}$ 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], but with $\epsilon = 0$.

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.
- $X = [x_1, \dots, x_N]^T \in \mathcal{M}_{N,D}(\mathbb{R})$, $Y = [y_1, \dots, y_N]^T$
- $\Phi_P(x) = [f_1(x), \dots, f_P(x)]^T$ and $Z_P = [\Phi_P^T(x_1), \dots, \Phi_P^T(x_N)] = [f_j(x_i)] \in \mathcal{M}_{N,P}(\mathbb{R})$
- $\hat{y}_{\mathcal{D}}(x) = \Phi_P^T(x) \hat{\beta}_P$, with $\hat{\beta}_P = \left[\frac{Z_P}{\sqrt{N\lambda}I_P} \right]^\dagger \begin{bmatrix} Y \\ O_P \end{bmatrix}$
- 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=77777000515>

2 Algorithms

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

```

1 function generate_orthonormal_basis(D, Deg, C)
  Input  $D, Deg, C$ :  $D \in \mathbb{N}^*, Deg \in \mathbb{N}^*, C \subset \mathbb{R}^D$ 
2   $\text{Basis}_{D'} = [\Pi_{d=1}^D X_d^{\alpha_d}, \sum \alpha_d \leq D']$ 
3   $P = \text{LENGHT}(\text{Basis}_{\text{Deg}})$ 
4  for  $1 \leq p \leq P$  do
5     $f'_p = \text{Basis}_{\text{Deg}}[p] - \sum_{i=1}^{p-1} [\int_C \text{Basis\_orthoDeg}[i] \cdot \text{Basis}_{\text{Deg}}[i]] \times \text{Basis\_orthoDeg}[i]$ 
6     $\text{Basis\_orthoDeg}[p] = \frac{f'_p}{\|f'_p\|_2}$ 
7  return  $\text{Basis\_orthoDeg}$ 

```

Algorithm 2: Dataset initialisation

```

1 function dataset_initialisation( $f, C, M, \text{ratio\_data}$ )
  Input  $y, C, M, \text{ratio\_data}$ :  $y : \mathbb{R}^D \rightarrow \mathbb{R}, C = [[a_d, b_d], 1 \leq d \leq D], \text{ratio\_data} \in [0, 1]$ 
2   $U = \mathcal{U}([0, 1]^{(M,D)})$ 
3   $X = \text{Diag}(a_d) + U \text{Diag}(b_d - a_d) \# X \hookrightarrow \mathcal{U}(C^M)$ 
4   $Y = y(X)$ 
5   $X_{\text{train}}, X_{\text{test}}, Y_{\text{train}}, Y_{\text{test}} = \text{test\_split}(X, Y, \text{ratio\_data})$ 
6  return  $X_{\text{train}}, X_{\text{test}}, Y_{\text{train}}, Y_{\text{test}}$ 

```

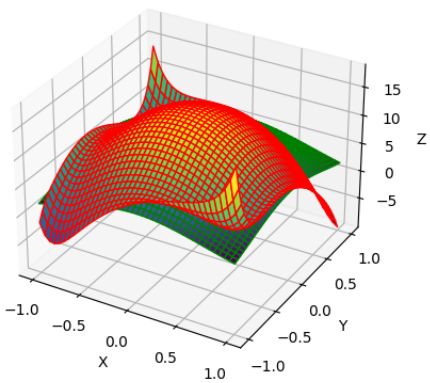
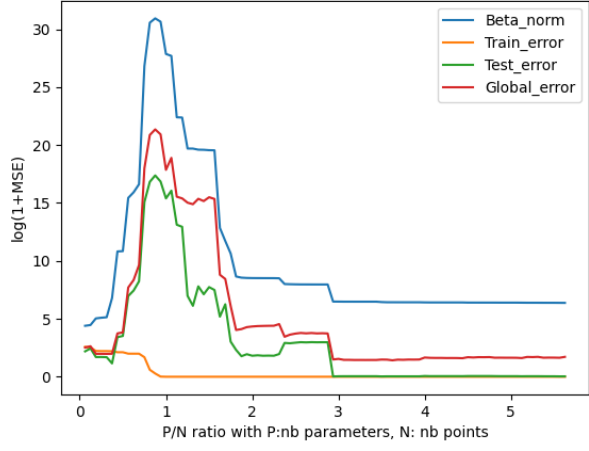
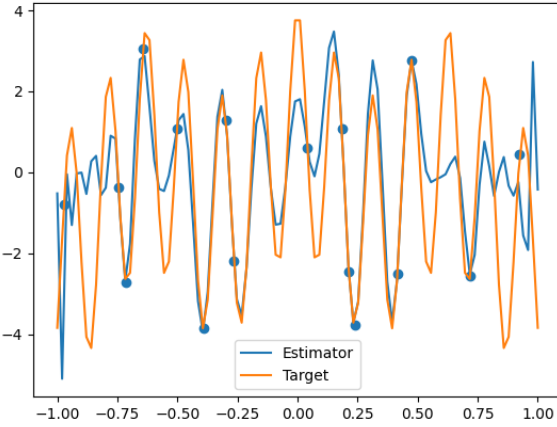
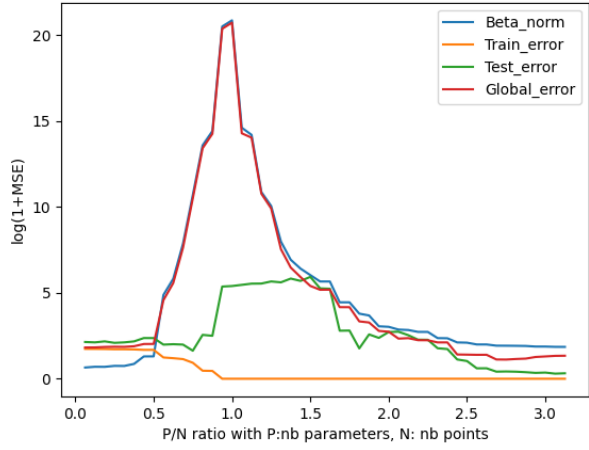
Algorithm 3: Ridge Regression

```

1 function ridge_regression( $P_{\min}, P_{\max}, \text{Features}, \lambda, \text{Data}, \text{Data\_global}$ )
2   $X_{\text{train}}, X_{\text{test}}, Y_{\text{train}}, Y_{\text{test}} = \text{Data}$ 
3   $X_{\text{global}}, Y_{\text{global}} = \text{Data\_global}$ 
4  for  $P_{\min} \leq p \leq P_{\max}$  do
5     $\Phi_p = [f_p, 1 \leq p \leq P]^T \# (f_p) \text{ an orthonormal basis for p.s. on } C$ 
6     $Z_p = \Phi_p^T(X_{\text{train}}) \# Z_p \in \mathcal{M}_{N,P}(\mathbb{R})$ 
7     $\hat{\beta}_p = \begin{bmatrix} Z_p \\ \sqrt{N\lambda}I_p \end{bmatrix}^+ \begin{bmatrix} Y_{\text{train}} \\ O_p \end{bmatrix}$ 
8     $\text{Train\_error}[p] = \log(1 + \text{MSE}(Y_{\text{train}}, Z_p \hat{\beta}_p))$ 
9     $\text{Test\_error}[p] = \log(1 + \text{MSE}(Y_{\text{test}}, \Phi_p^T(X_{\text{test}}) \hat{\beta}_p))$ 
10    $\text{Global\_error}[p] = \log(1 + \text{MSE}(Y_{\text{global}}, \Phi_p^T(X_{\text{global}}) \hat{\beta}_p))$ 
11    $\text{Beta\_norm}[p] = \log(1 + \|\hat{\beta}_p\|_2^2)$ 
12   $\hat{y} = \Phi_p \cdot \hat{\beta}_{P_{\max}}$ 
13  return  $\text{Train\_error}, \text{Test\_error}, \text{Beta\_norm}, \text{Global\_error}, \hat{y}$ 

```

3 Examples (random seed : 23334)

Function plot	Graph error
<p>Estimator: red and target: green, $\lambda = 0.0$</p>  <p>$y(x) = 2e^{-(x_1^2+x_2^2)}$, Basis : <i>cano</i></p>	 <p>$M = 20, r = 0.2, n = 40, a = 1, Deg = 12, D = 2, \lambda = 0$</p>
 <p>$y(x) = 3\cos(40x) + \cos(10x) - x^2$, Basis : <i>ortho</i></p>	 <p>$M = 20, r = 0.2, n = 100, a = 1, Deg = 50, D = 1, \lambda = 0$</p>

References

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