Internship Report

Ylann Rouzaire

September 14, 2020

This internship can be separated in three distinct and independent projects, presented in chronological order :

1. The investigation of the dynamics of learning in a Teacher/Student Kernel Regression framework.
2. The investigation of the behaviour of a noiseless hard-margin SVM classifier in presence of a gap at the interface between the 2 classes.
3. The premises of an understanding of the learning of Boolean matrices.

Part [2] is the most consequent one, therefore, an abstract precedes the full description in order to sum up the main ideas and the result.

This project was supervised by Matthieu Wyart and Stefano Spigler.

Contents

1 Teacher/Student Kernel Regression Dynamics

1.1 Settings

Kernel : A kernel is a function from $\mathbf{R}^d$ x $\mathbf{R}^d$ to $\mathbf{R}$. We will only deal with isotropic translation-invariant kernel $k(x, x') = k(||x - x'||)$ of the Matérn family :

$$f_\nu(h) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{h}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{h}{\rho} \right), \tag{1}$$

where Γ is the Euler Gamma function, K_ν is the modified Bessel function of the second kind, $\rho = 1$ is the characteristic length scale and $\nu > 0$ is the smoothness parameter (the larger the smoother, $\nu = \infty \leftrightarrow$ Gaussian kernel).

Teacher/Student framework : The idea is the following : the Teacher generates data according to its own "law" K_T. A subset of size P of that data called the *training set* is passed to the Student. From it and thanks to its "law" K_S, the Student has to infer the underlying "law" K_T. Its performance is measured by the test error ϵ , defined as a quadratic loss between the prediction of the Student and true values at several *new/unseen* points generated by the Teacher.

$$\epsilon = \epsilon\left(P, \nu_T, \nu_S\right) = \mathbb{E}_T \; \frac{1}{P_{test}} \sum_{\mu=1}^{P_{test}} |\hat{Z}^S(x_\mu) - Z^T_{test}(x_\mu)|^2 \tag{2}$$

where $\mathbb{E}_T$ is the expectation over the Teacher random process.

In this project, Teacher kernels are chosen among the Matérn family (1) and the Student kernel will always be the Laplace kernel :

$$\text{Laplace}(x, x') = \text{Matérn}\,[\nu = 1/2](x, x') = \exp(-||x - x'||) \tag{3}$$

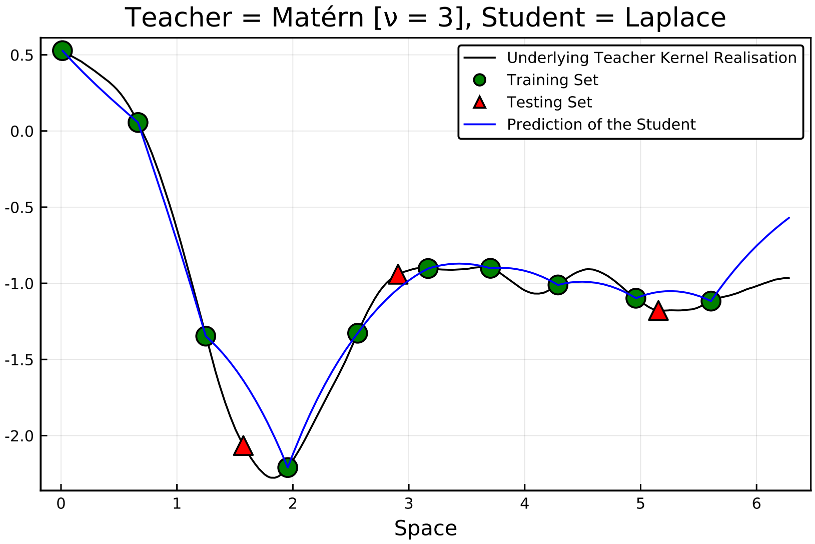


Figure 1:
Illustration of the Teacher/Student framework in the context of Matérn kernel Regression. The test error ϵ is the sum over all testing set of the square distance between the red triangles and value of the blue curve at the same x coordinate.

Generation of the data by the Teacher kernel :

$$Z^T \sim \mathcal{N}(0, K_T) \tag{4}$$

so that

$$\mathbb{E}\ Z^T(x) = 0 \text{ and } \mathbb{E}\ Z^T(x)Z^T(x') = K_T(x, x') \tag{5}$$

Prediction of the Student kernel :

$$\hat{Z}^S(x_0)(t) = \sum_{\mu=1}^{P} a_\mu(t) K_S(x_\mu, x_0) = a(t) \cdot k_S(x_0) \tag{6}$$

where $a(t)$ is a vector of weights of size P changing over time during the learning procedure. The different optimisation routines implemented in the code are described in the README.md file of the project.

1.2 Features of the code

The Julia code can be found here : https://github.com/Rouzaire/Kernel_Regression. The README.md file is self-explanatory and contains general information on the code and its structure. The code itself is commented and adds more precise details.

1.3 Numerical Results

The first checkup of the code was to recover the scaling for ϵ versus P found in [1] :

$$\epsilon \sim p^{-\beta}, \text{ with } \beta = \frac{1}{d} \min(\alpha_T - d, 2\alpha_S) \tag{7}$$

My main contribution at this stage is that my code is faster than what was previously coded.

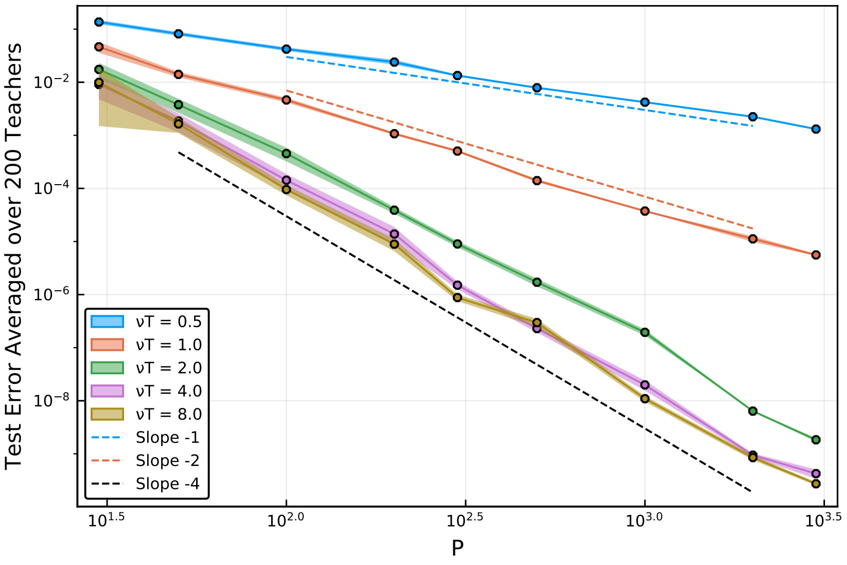
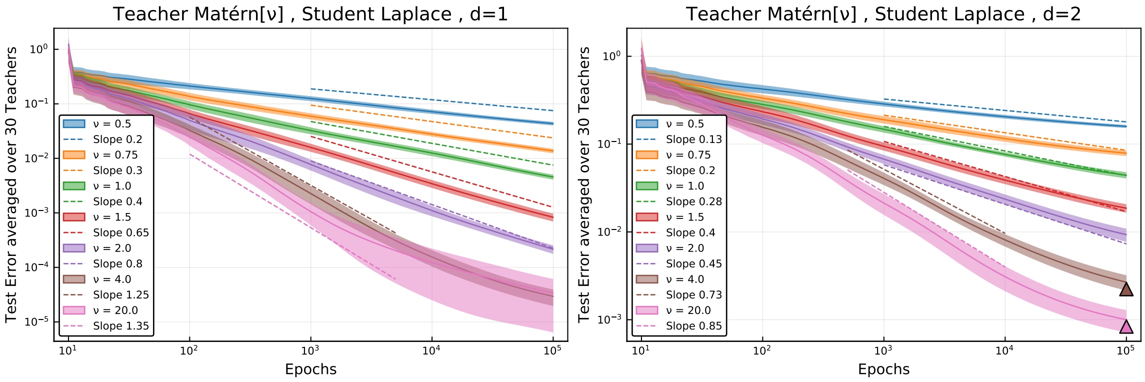


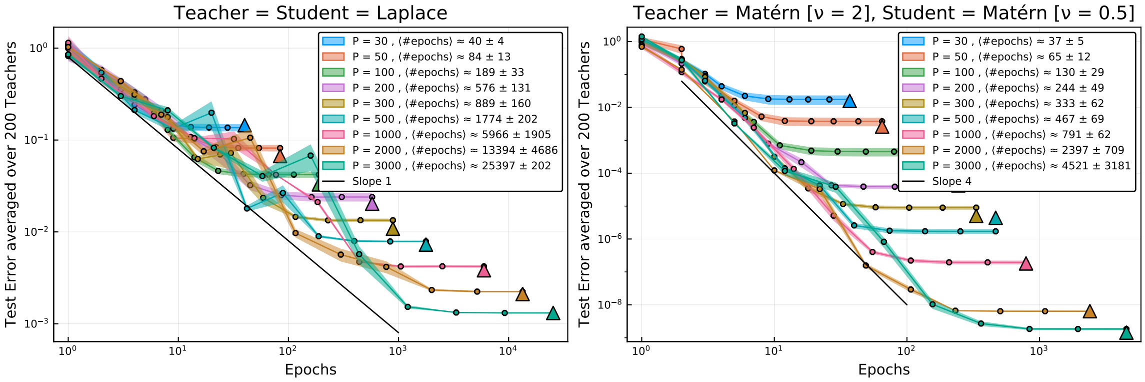
Figure 2: Learning curves for several Teacher kernels, here for $d = 1$. Student kernel is fixed to Laplace, cf. eq (3). One recovers the scaling predicted in [1].

Then I could investigate the dynamics of the test error during the optimization procedure. The conclusion is that for the Gradient Descent algorithm, the dynamics is very monotonic and exhibits no overfit : the test error is always decreasing and its limit is the exact test error plotted with triangles (only in Fig. (3b) for readability) :



(a) (b)

Figure 3: Optimization : Gradient Descent.



(a) (b)

Figure 4: Optimization : Conjugate Gradient Descent.

One can notice that for intermediate times, both for GD and CGD, the test error decays in a powerlaw. This is part of the analytic work in progress :

1.4 Analytical developments

The test error ϵ can be decomposed in two parts : the dynamical part, which will tend to zero as the number of epochs $t \to \infty$, and an infinite time error which represent the error due to the finite number of training points :

$$\epsilon(p,t) = \epsilon_{\text{dyn}}(p,t) + \epsilon_{\infty}(p) \qquad (8)$$

1.4.1 Infinite time error

This error arises from the fact that the trainset is of finite size p. It is therefore impossible to learn more than the first p "modes" (or wavelengths) of the Teacher kernel. It is possible to recover analytical scaling for ϵ_∞ as a function of p with different approaches : rigorous approaches are developed in [1] and [3]. The next paragraph is a side work recovering the same result from a more general framework.

In [2], the authors suggest that the test error can be decomposed in a sum over the modes ρ of the target function of error per mode. They provide with an expression for the error per mode : NB : $\Lambda \geq 0$ is the rigde parameter.

$$E = \sum_\rho E_\rho, \text{ where } E_\rho(p) = \frac{\langle \bar{\omega}_\rho^2 \rangle}{\lambda_\rho} \left(\frac{1}{\lambda_\rho} + \frac{p}{\Lambda + t(p)} \right)^{-2} \left(1 - \frac{p\,\gamma(p)}{(\Lambda + t(p))^2} \right)^{-1} \tag{9}$$

$$t(p) = \sum_\rho \left(\frac{1}{\lambda_\rho} + \frac{p}{\Lambda + t(p)} \right)^{-1} \tag{10}$$

$$\gamma(p) = \sum_\rho \left(1 - \frac{p\,\gamma(p)}{(\Lambda + t(p))^2} \right)^{-2} \tag{11}$$

Note that these equations are the exact errors, there is no notion of dynamics here. We wanted to check whether for large P, these equations boiled down to the scalings relations already found in [1] for the ridgeless case. The answer is yes and here are some intermediate results :

Without rigde : $\Lambda = 0$, for $p \gg 1$

$$t(p) \sim p^{-s} \quad s = \frac{2-\theta}{\theta - 1}$$

$$\gamma(p) \sim p^{-r} \quad r = \frac{3-\theta}{\theta - 1}$$

$$E(p) \sim p^{-\beta} \quad \beta = \frac{1}{d} \min(\alpha_T - d, 2\,\alpha_S) \tag{12}$$

With rigde : $\Lambda \sim \mathcal{O}(1)$, for $p \gg 1$

$$t(p) \sim p^{-s} \quad s = 2 - \theta$$

$$\gamma(p) \sim p^{-r} \quad r = 3 - \theta$$

$$E(p) \sim p^{-\beta} \quad \beta = \frac{1}{\alpha_S} \min(\alpha_T - d, 2\,\alpha_S) \tag{13}$$

NB : $\alpha = d + 2\,\nu > d$ so if the data is not noisy, a regression with ridge will never be optimal/better than a ridgeless regression.

Finally, in the ridgeless case, it is also possible to provide with a very simple argument that reproduces the main part of Eq.(12) :
Let's denote $\{\lambda_i\}_{i\in\mathbb{N}^*}$ the eigenvalues of the kernel. The density of small eigenvalues is defined as $\lambda^{-\theta}$. Thus, one has :

$$p \sim \int_0^{\lambda_p} d\lambda\ \lambda^{-\theta} = \lambda_p^{1-\theta} \iff \lambda_p \sim p^{-\frac{1}{\theta-1}} \tag{14}$$

Then comes the argument : the error generated by each mode i is proportional to λ_i, leading to :

$$\epsilon_\infty \sim \int_0^{\lambda_p} d\lambda\ \lambda^{-\theta}\cdot\lambda = \lambda_p^{2-\theta} \sim p^{-\frac{2-\theta}{\theta-1}} \equiv p^{-\tilde{\beta}} \tag{15}$$

where $\tilde{\beta} = \frac{\alpha-d}{d}$. This simple argument therefore recovers the desired result in the regime where the learning is dictated by the $\alpha_T - d$.

1.4.2 Dynamical test error

In this section, I will present the results obtained when using the same type of argument to explain the dynamical part of the test error.
WARNING : it does not match the experimental exponents of Fig. 3 and I could not find out why. I will explain possible issues after the argument.

As before, one has the $\lambda^{1-\theta}$ part coming from the error when trying to learn the different modes of the Teacher function, but due to Gradient Descent dynamics of the Student kernel, an exponential arises :

$$\epsilon_{dyn} \sim \int_{\lambda_p^T}^{\lambda_1^T\sim 1} d\lambda_T\ \lambda_T^{1-\theta_T} e^{-\eta t\lambda_S} \tag{16}$$

This exponential represents the learning process : for each eigenvalue, one need to learn for a long enough time t before the associated error becomes negligible. It is possible to carry out exact analytic computation using Gamma functions and its incomplete versions (upper and lower) but I will here present a more intuitive way to proceed. Note that they both deliver the same result. Let's crudely approximate the exponential by a step function :

$$e^{-\eta t\lambda_S} \equiv \begin{cases} 0 & \text{if } \eta t > \frac{1}{\lambda_S} \\ 1 & \text{otherwise} \end{cases} \tag{17}$$

The integral in Eq.(16) then becomes

$$\epsilon_{dyn} \sim \int_{\lambda_p^T}^{\lambda_S^{-1}} d\lambda_T\ \lambda_T^{1-\theta_T} \sim \lambda_T^{2-\theta_T}\Big|_{\lambda_p^T}^{\lambda_S^{-1}} \sim t^{-(2-\theta_T)\cdot\frac{\theta_T-1}{\theta_S-1}} \equiv t^{-s} \tag{18}$$

where I have used

$$\lambda_\rho^{1-\theta} \sim \rho \iff \lambda_T^{1-\theta_T} \sim \lambda_S^{1-\theta_S}$$

Unfortunately, even in the simplest case Teacher=Student=Laplace ($\nu = 1/2$), the exponent $s = 2-\theta = 2-(1+\frac{d}{d+2\nu}) = 1-\frac{d}{d+1} = \frac{1}{d+1}$ which does not match the exponents found in Fig. 3. It is approximately 2.5 times too large. Possible issues and comments :

- The numerics are incorrect for some reasons I cannot explain since they gave consistent and/or correct results for all the other topics.

- The assumption that the error on a mode ρ is proportional to λ_ρ is wrong but once again, the same hypothesis was made in the simple argument to recover the correct scaling for ϵ_∞.

- The approximation of the exponential is not controlled, but as I said, the Taylor expansion of the incomplete Gamma functions give the same exponent s for intermediate times.

NB : $s = (2 - \theta_T) \cdot \frac{\theta_T - 1}{\theta_S - 1}$ gives incorrect results but $s = \frac{2-\theta_T}{1+\theta_T} \cdot \frac{\theta_T - 1}{\theta_S - 1}$ give the correct scaling but I did not understand where did this factor $1/(1 + \theta_T)$ come from.

2 Kernel Classification with a gap at the interface

2.1 Settings

We investigate the learning curves of a supervised *2-class hard-margin kernel classification* task when the data present many invariant. In this project, the data labels will only depend on the first component x_1 of $\vec{x} \in \mathrm{R}^d$, all other dimensions are irrelevant for labelling : $y(\vec{x}) = \text{sgn } x_1$.

Generation of the data :
The data is generated uniformly on the unit hypersphere of dimension d [1] by normalizing (to 1) points from a multivariate random normal distribution $\mathcal{N}(0, I_{d+1})$ in d+1 dimensions.

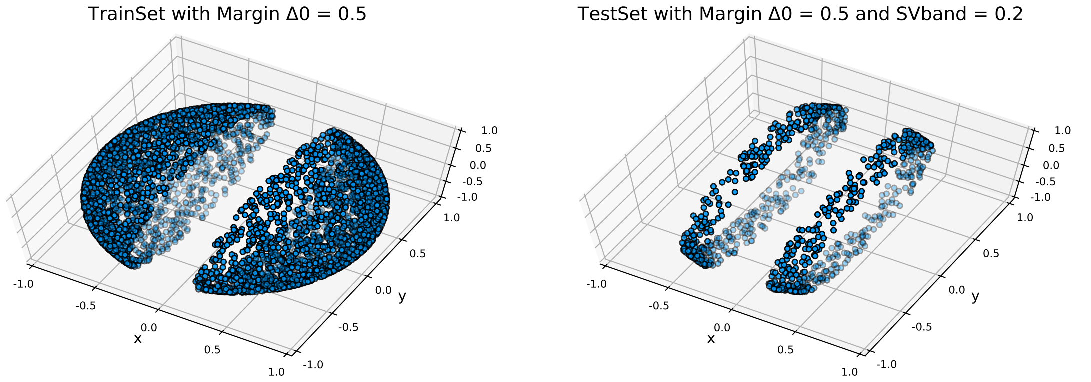
This project studies the consequences of a *gap* of size $\Delta_0 \geq 0$ at the interface between labels. There is no data (neither training nor testing) in the gap. Intuitively, since the more distant from the interface, the easier to classify, one could expect less misclassification errors. Indeed, one shall see that the learning curve decays as an exponential and not as a power law anymore when there is a gap between labels : see Fig. 7.

This very fact that the error $\epsilon(p)$ decays geometrically fast in presence of a gap $\Delta_0 > 0$ means that the testing set has to be at least bigger than $1/\epsilon$ to record at least one error. This quickly becomes numerically untractable in terms of memory and CPU resources. This is why the code implements and *Importance Sampling* (IS) algorithm, at least conceptually. The idea is that if a test point is far from the interface, it will for sure be correctly classified : it is therefore useless to compute that prediction. Thus, the testing set only contains data points in the vicinity of the gap. The resulting test error will then be weighted by the probability of falling into that area compared to all the surface of the unit hypersphere (minus the Δ_0 zone). This last step relies on the assumption that all the possibly misclassified points are contained in the testing set : the choice of the distance (called SVband) to the gap is therefore of capital importance. Train and Test sets can be vizualised in Fig. 5.

2.2 Features of the code

The Julia code can be found here : `https://github.com/Rouzaire/Kernel_Classification`. The README.md file is self-explanatory and contains general information on the code and its structure. The code itself is commented and adds more precise details.

[1] For clarity : $d = 1$ means the unit circle and $d = 2$ means the usual sphere embedded in the natural 3 dimensions.


TrainSet with Margin Δ0 = 0.5

TestSet with Margin Δ0 = 0.5 and SVband = 0.2

(a) Example of Training Set for $d = 2$.

(b) Example of Testing Set for $d = 2$.

Figure 5: Data is uniformly distributed around the unit hypersphere.

2.3 Description of Support Vector Classifier (SVC) with gap

2.3.1 Abstract

It has already been shown in [3] that the test error ϵ decays algebraically for a classification task. The present work extends this result for a setting with a gap at the interface and the resulting scaling of ϵ embraces the result for the gapless setup. We will show that the test error can be seen as a consequence of spatial fluctuations of the SVC's decision boundary $f(\vec{x}) = 0$, seen as a realisation of a Gaussian Process (GP). Identifying ϵ with the excursion area of this GP beyond the gap yields to (in perfect agreement with numerics) :

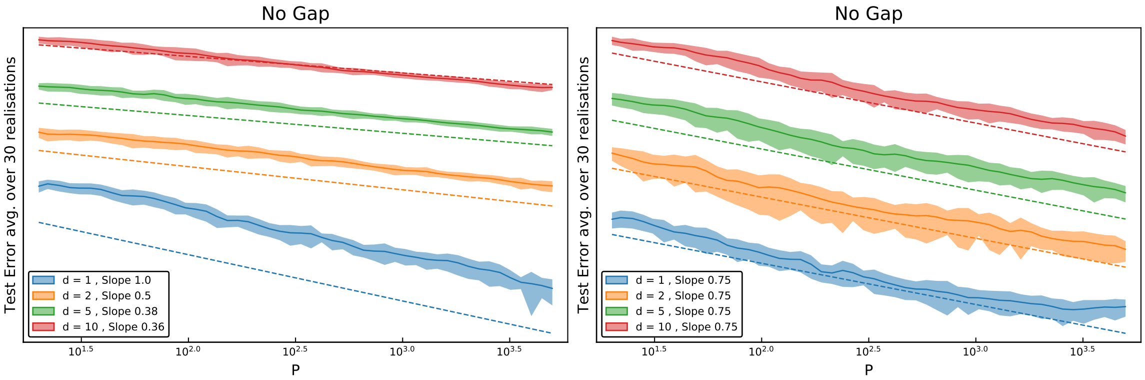
$$\epsilon(p, \Delta_0) \sim p^{-\tilde{\beta}} \left[1 - \mathrm{erf}(\frac{\Delta_0}{2} \sqrt{p}) \right] \quad \text{where } \tilde{\beta} = \frac{d - 1 + \xi}{2d - 2 + \xi}. \tag{19}$$

2.3.2 Recall : result for a gapless setup

The main reference for such a gapless setup is [3]. In this paper, authors work out analytic scaling relations leading to

$$\epsilon(p) \sim p^{-\beta} \quad \text{where } \beta = \frac{d + \xi - 1}{3d - 3 + \xi} \ . \tag{20}$$

This paper tackles kernel classification with Matérn kernels such that the exponent ξ that governs the cusp at the origin of the kernel satisfies $0 < \xi = \min(2, 2\nu) < 2$.
For Laplace kernel $k(h) = \exp(-h/\sigma)$, $\xi = 1$.
For Gaussian kernel $k(h) = \exp(-\frac{h^2}{2\,\sigma^2})$, $\xi = 2$.
Unless stated otherwise, $\sigma = 100 \gg 1$ so that the kernel is evaluated only close to the origin (simpler to work out analytically thanks to Taylor expansions).
WARNING : The theory developed in [3] thus does not apply to Gaussian kernels for whom $\xi = 2$. Thus, any fit for Gaussian kernel are "visual" fits and are not (yet) arising from any theoretical foundation.



(a) Laplace kernel

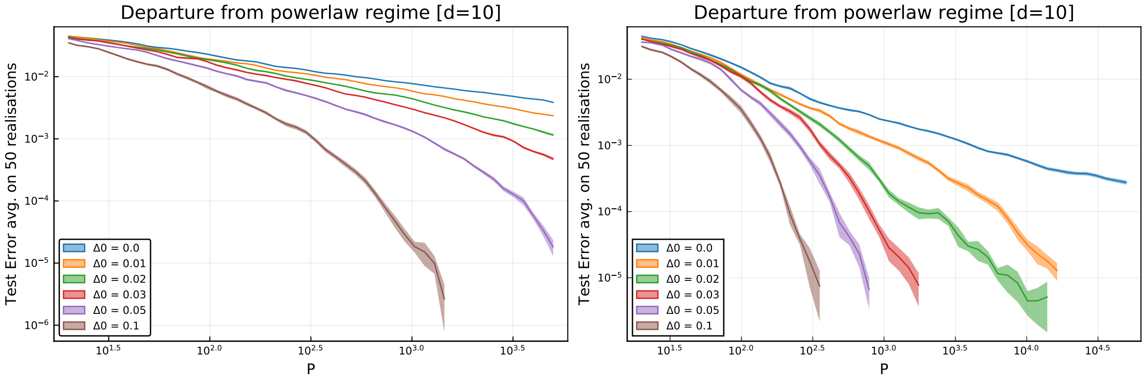
(b) Gaussian kernel

Figure 6: Learning curves in LogLog scale, No Gap Setup. NB : curves have artificially been shifted up for readability reasons so yaxis values become meaningless.

For Laplace kernel, the theory developed in [3] applies and is in perfect agreement : one recover the correct exponent. On the other hand, it appears that the Gaussian kernel exhibits a more or less constant exponent β that is numerically found to be close to $3/4$ for all dimensions. This could motivate further theoretical work.

2.3.3 Visualisation of the decision boundary for a 2D problem

Reference [4] suggests that the introduction of a gap between labels will add an geometrically decreasing component to the test error. This is indeed the case cf. Fig. 7, and we shall see that one way to tackle this problem is to focus on the decision function $f(\vec{x})$ and more particularly on the subspace of the decision boundary $f(\vec{x}) = 0$. Indeed, there exists a tight link between "extreme" fluctuations of the decision boundary and the test error.



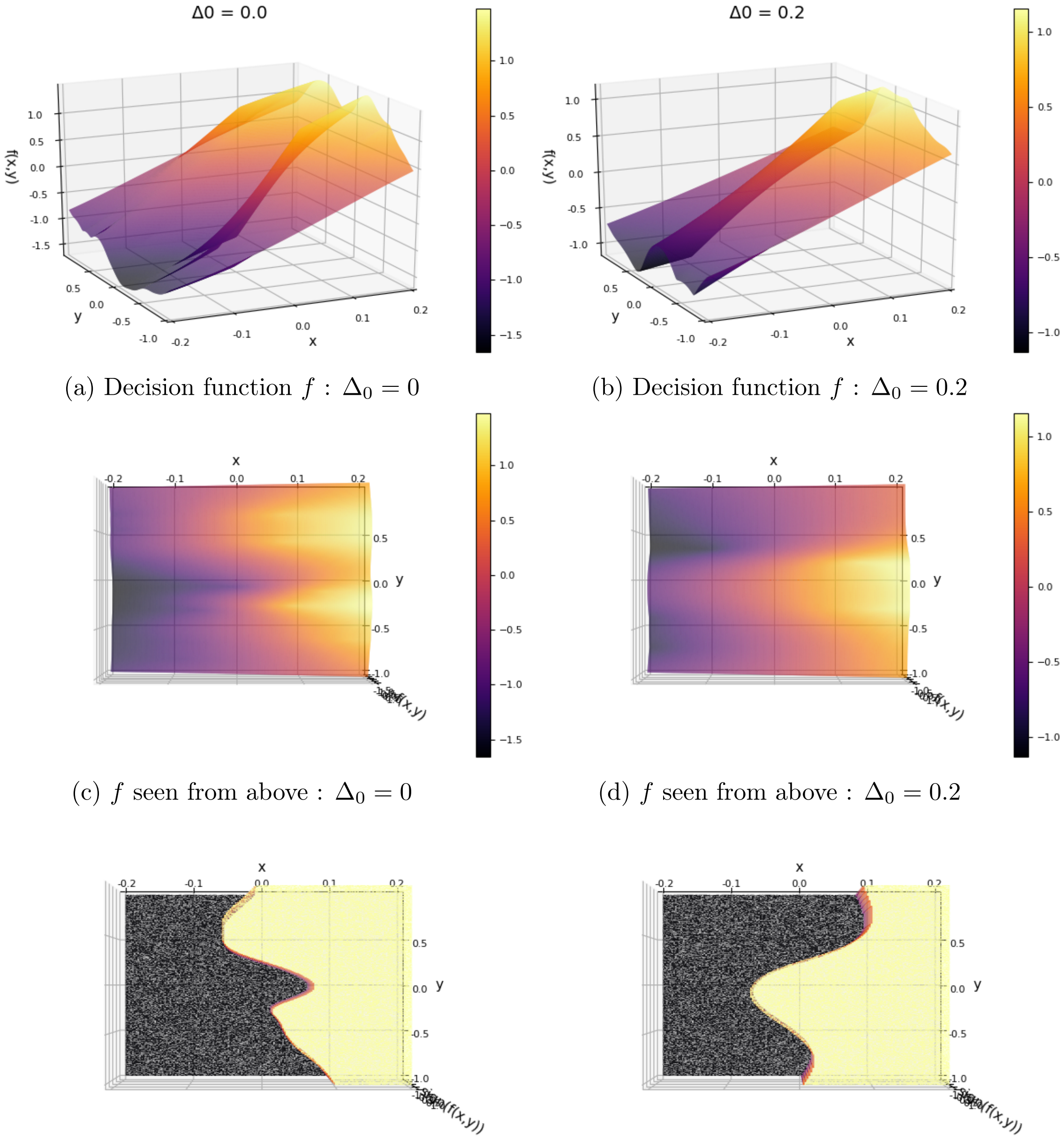
(a) Laplace kernel

(b) Gaussian kernel

Figure 7: Departure from powerlaw regime, here $d = 10$. The LogLog scale exhibits a mixture of powerlaw (straight lines) and exponential-like behaviour (departure from straight lines).

The SVM classifier algorithm fits the data creating a function f whose sign matches the labels of the provided training points. The prediction y_0 at a new point x_0 is given by $y_0 = \text{sgn}\ f(x_0)$. The understanding of the consequences on this function f of increasing the gap between labels is therefore capital. All the following figures were made using the *decision_boundary.jl* file.

Data lying on $[-1,1]^2$ **:**
The trainset (of size p) is sampled randomly uniformly while the testset is a simple fine meshgrid over $[-0.2, 0.2]^2$.



(a) Decision function f : $\Delta_0 = 0$

(b) Decision function f : $\Delta_0 = 0.2$

(c) f seen from above : $\Delta_0 = 0$

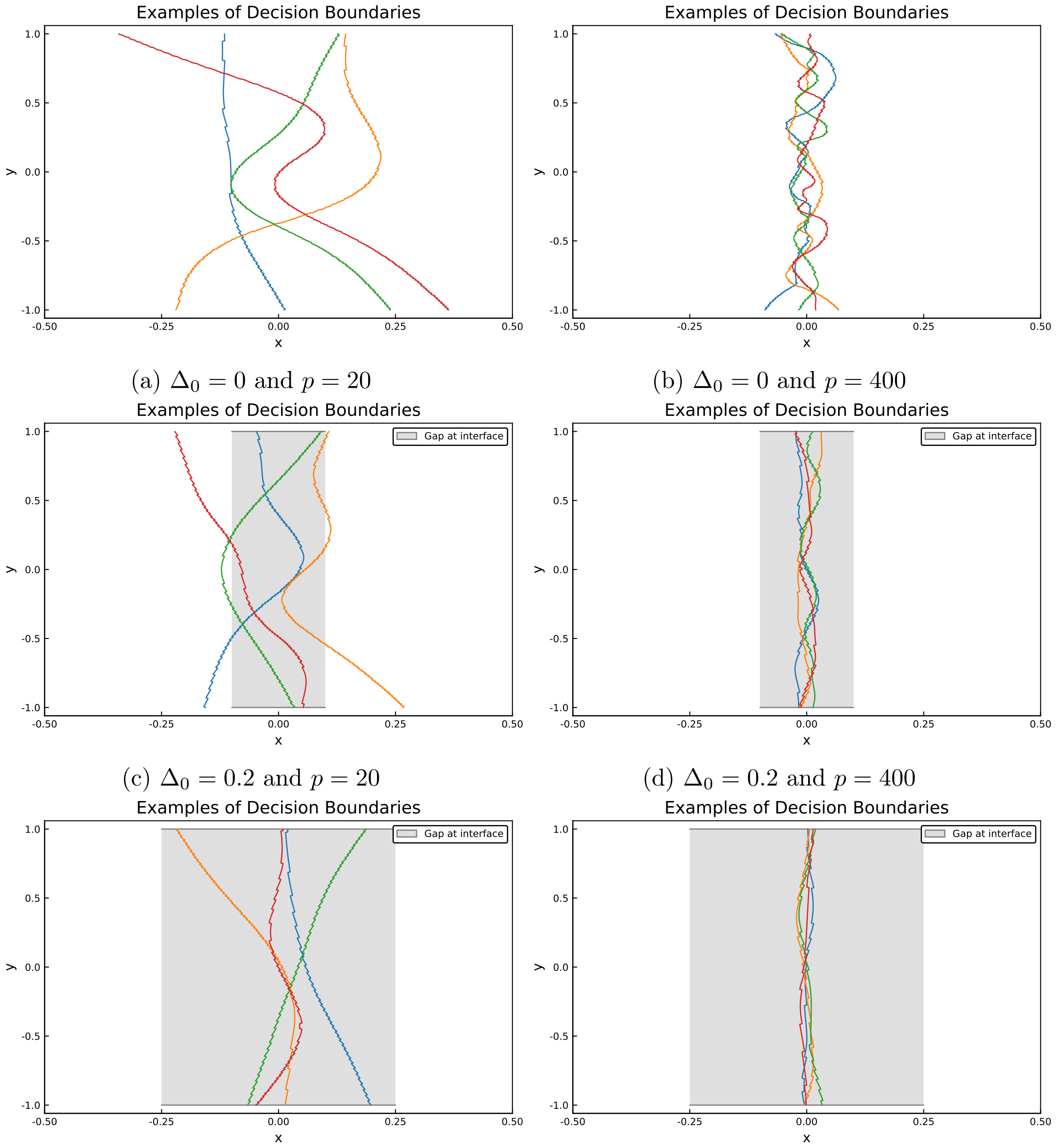
(d) f seen from above : $\Delta_0 = 0.2$

(e) Prediction seen from above : $\Delta_0 = 0$

(f) Prediction seen from above : $\Delta_0 = 0.2$

Figure 8: Decision function and prediction for $\Delta_0 = 0$ and $\Delta_0 = 0.2$. Here $p = 100$ and $d = 2$.

The pattern drawn in Fig. 8e and 8f are the so-called decision boundaries, corresponding to the subspace where $f(x) = 0$. In what follows, this subspace of dimension $d-1$ will be denoted by f_0. One can isolate them to study their properties :



(a) $\Delta_0 = 0$ and $p = 20$

(b) $\Delta_0 = 0$ and $p = 400$

(c) $\Delta_0 = 0.2$ and $p = 20$

(d) $\Delta_0 = 0.2$ and $p = 400$

(e) $\Delta_0 = 0.5$ and $p = 20$

(f) $\Delta_0 = 0.5$ and $p = 400$

Figure 9: Examples of boundary decisions for several p and several gap sizes. Left panels : $p = 20$, Right panels : $p = 400$. Top panels : $\Delta_0 = 0$, middle panels : $\Delta_0 = 0.2$, bottom panels : $\Delta_0 = 0.5$.

It clearly appears that the amplitude of the boundary decision decreases as p increases : that is the visualisation of the learning process. In the gapless setup, as $p \to \infty$, the boundary $f_0(y)$ will tend in distribution to the vertical line $x = 0$ to minimize the error made when departing from it.
The rate at which this amplitude $A(p)$ decreases is key in the argument for the scaling of the test error : cf Eq. (26).

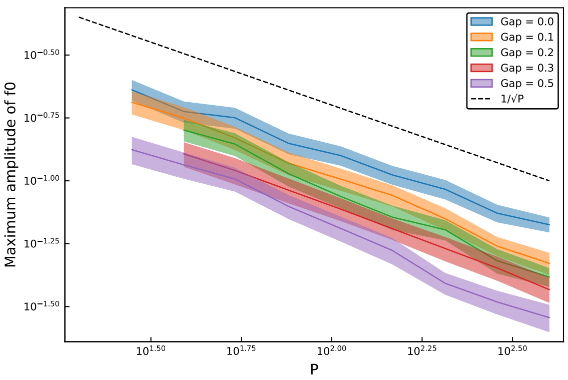


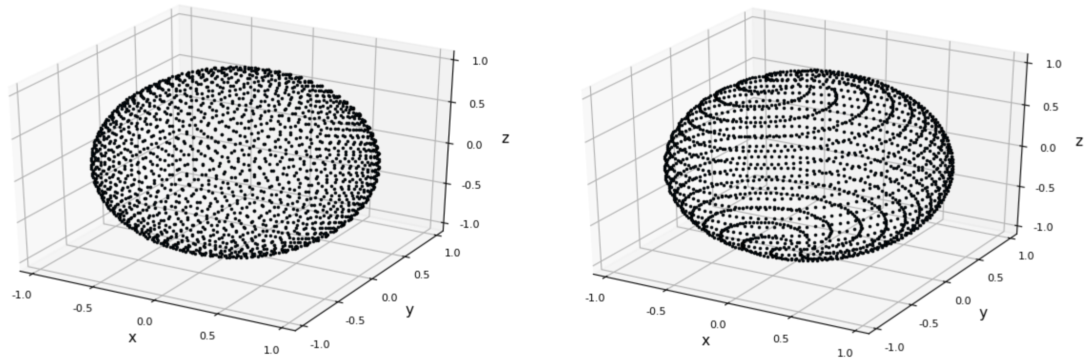
Figure 10: Amplitude of f_0 behaves like $p^{-1/2}$, independently of Δ_0.

Note that if one introduces a gap, an intuitive thought is that the decision is more free to fluctuate within the gap (grey area in Fig. 9). In fact, what happens is the complete opposite. One can visually attest that as the gap increases, the decision boundaries become smoother and smoother at fixed p. This is due to the fact that the function f is not constrained to match constantly changing labels on its path. Indeed, when $\Delta_0 = 0$ and as $p \to \infty$, it is possible to find points arbitrarily close to the interface. Since the algorithm is *hard-margin*, it means that it will try to fit each and every point, leading to these high-frequency fluctuations.

The distributions of distance to the interface for various values of Δ_0 and p can be found in the Appendix B because they might be of interest for a deeper understanding but are not important in the argument for the scaling of ϵ .

Data lying on the 2D unit sphere :
In this case, the trainset (of size p) is sampled randomly uniformly on the sphere with the algorithm explained in the paragraph "Generation of the data" of section 2.1. The choice of the testset was bit more complex than in the 2D plane case because it is in general impossible (except for special rare cases where N takes some special values) to evenly distribute N points on a sphere. This question has been tackled in [10]. There exist several methods (Latitude–longitude lattice, Fibonacci lattice, Dave Rusin's "disco ball", Saff & Kuijlaars method etc) but none is perfect. The choice is to be made as a tradeoff between ease of the implementation, degree of anisotropy, visual aspect, computational complexity and strong presence of poles. In what follows, I have chosen to construct the testset onto the Fibonacci lattice (aka Fibonacci Sphere) :

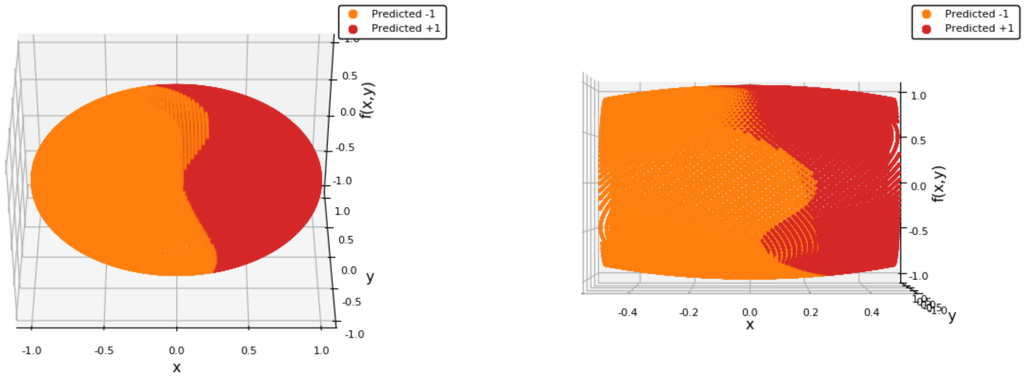
```
** Fibonacci Lattice Code **
    indices = 1:N
    seed = (1 + sqrt(5))/2
    theta = acos.(1 .- 2 indices/N)
    phi = 2pi * seed * indices
    x, y, z = cos.(phi) .* sin.(theta) , sin.(phi) .* sin.(theta) , cos.(theta)
```



(a) Using $\phi = \frac{1+\sqrt{5}}{2}$ as the "seed".

(b) Using π as the "seed".

Figure 11: Two Fibonacci spheres of $N = 2000$ points each. The left panel is the one actually used for simulations. *For information only*, the right panel is the same code but using π as "seed" instead of the golden ratio ϕ. The stripes appear because π is an easier to approximate/more regular irrational number, leading to periodic patterns. Indeed, the golden ratio is the slowest irrational number to converge under its continued fraction form : cf. this link.



(a) Prediction on the whole unit sphere.

(b) Zoom on the slice $-\frac{1}{2} < x < \frac{1}{2}$ of left panel.

Figure 12: Visualisation of the prediction and decision boundary on the sphere.

2.3.4 Scaling relations for the test error

The test error $\epsilon(p, \Delta_0)$ is closely related to the area between the decision boundary f_0 seen as a function of y and as a realisation of a $GP(0, k)$ [2]. For a fixed p, let's approximate this area as

$$\epsilon \sim \Delta \cdot \mathbb{E}\ \mu(\Gamma)\ , \tag{21}$$

where Δ is the characteristic excess amplitude (amplitude beyond the gap), which will turn out to be the band where support vectors are located, where $\mathbb{E}$ is the expectation value on the random process and where $\mu(\Gamma)$ is the Lebesgue measure of the so-called random

[2] Note that the profound reason why the decision boundary can be modelled by a Gaussian Process is not yet understood. It may be somehow related to the $1/\sqrt{p}$ decay of the amplitude (cf. Fig. 10) because both surely arise from the Central Limit Theorem.

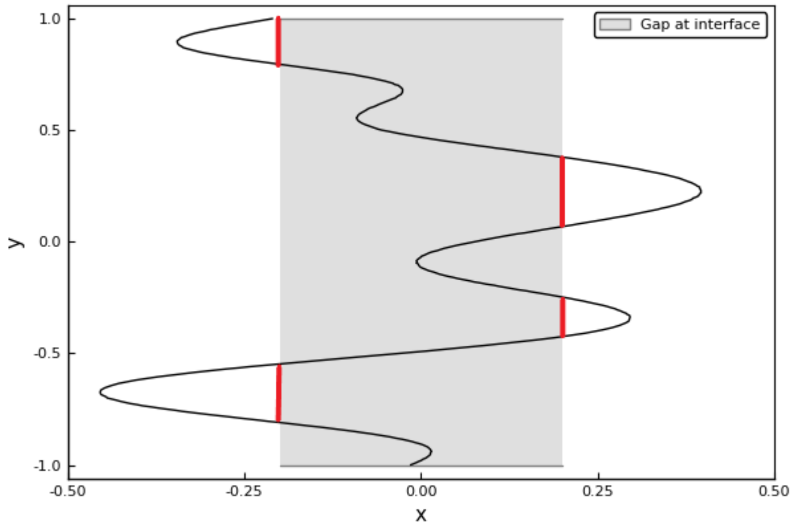


Figure 13: Visual definition of the excursion set.

excursion set Γ.

Random sets and excursion sets

Without entering technical definitions, a random set can be seen a random subset (here) of the real line. For our purposes, let us say it depends on an underlying realisation of some random process $(Z_x)_{x\in D}$, and define the random set

$$\Gamma \equiv \{x \in D : Z_x \in T\} \quad \text{, where } T \text{ is a closed set.} \tag{22}$$

An excursion set is a particular kind of random set when $T = [t \geq 0; +\infty[$. This is what interests us at this stage, with $t = \frac{\Delta_0}{2}$.
Note that

$$\mu(\Gamma) = \mathbb{P}\left(Z_x \in T\right) \quad \text{, with } x \in D \tag{23}$$

For clarity, the red segments in Figure 13 are the excursion set Γ and their cumulated lengths divided by 2 (2 = the Lebesgue measure of the interval $[-1, 1]$) represent its Lebesgue measure $\mu(\Gamma)$.

Finally, note that in the case where the random process Z_x comes from a Gaussian process $GP(m(x), k(x, y))$, their exists a closed form expression for $\mu(\Gamma)$ ([8], Ex. 8) :

$$\mu(\Gamma) = \mathbb{P}(x \in \Gamma) = \mathbb{P}(Z_x \geq t) = \Phi\left(\frac{m(x) - t}{\sqrt{k(x, x)}}\right) \quad , \tag{24}$$

where Φ is the cdf of a standard Gaussian distribution.

If $m(x) = 0$ and a translation invariant kernel such that $k(h = 0) = 1$, then by symmetry

$$\mathbb{P}(|Z_x| \geq t) = 2\,\Phi\left(-t\right) = \text{erfc}(t) = 1 - \text{erf}(t) \ , \tag{25}$$

where erf is the error function.

A important remark : in Eq. (24), the kernel only enter under the form $k(x, x)$: thus, there is no kernel dependency in the final result as long as the kernel is translation-invariant. Since the boundary decision is periodic by construction when the problem lies on a unit hypersphere, a simple symmetry argument implies that the underlying kernel (if the decision boundary indeed arises from a Gaussian Process) is invariant by translation along the interface.

Application to our problem for varying p

Observation of Figure 10 taught us that the amplitude of the decision function f_0 decays like $p^{-1/2}$. Therefore, the random set of interest is

$$\Gamma \equiv \left\{ y \in [-1,1] : \frac{|f_0(y)|}{\sqrt{p}} \geq \frac{\Delta_0}{2} \right\} \ , \tag{26}$$

where, and that precision is very important, f_0 is drawn from a GP that only depends on Δ_0 (but not on p anymore). Applying the result of eq. (25) gives a contribution to the error of $1 - \mathrm{erf}\big(\frac{\Delta_0}{2}\sqrt{p}\big)$.

Characteristic width of the support vector band Δ in presence of a gap Δ_0

The arguments presented hereafter are directly inspired from those of reference [3], and adapted to the case where one has a gap $\Delta_0 \gg \Delta \iff p \gg 1$. Note that this approximation is the only reason why results of eq. (20) [No Gap] and (19) [Gap] seem incompatible in the limit $\Delta_0 \to 0$.

The only change in the arguments developed in the reference is that the decision function changes on the scale of the gap, not anymore on the scale Δ : $\Delta_0 \nabla f \sim 1$, leading to the second scaling relation [3] :

$$p \, \Delta^2 \, \bar{\alpha} \sim 1 \tag{27}$$

Together with the unchanged scalings

$$p \, \Delta \, r_c^{d-1} \sim 1 \ \text{ for the distance to nearest support vector} \tag{28}$$

and

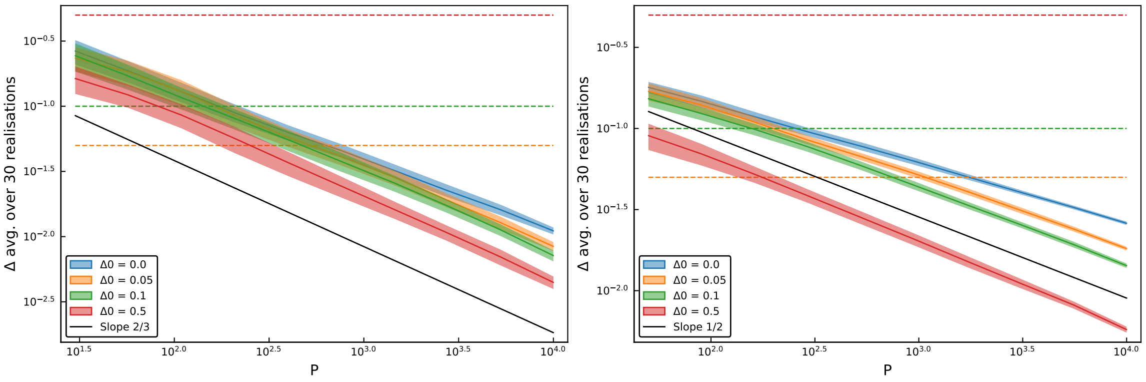
$$f(x + r_c \, \hat{e}_\perp) - f(x) \sim 1 \Rightarrow p \, \Delta \, \bar{\alpha}^2 \, r_c^{d-1+2\xi} \sim 1 \tag{29}$$

Solving the system of equations (27) to (29) for Δ yields

$$\Delta \sim p^{-\tilde{\beta}} \quad , \text{with } \tilde{\beta} = \frac{d-1+\xi}{2d-2+\xi} \tag{30}$$

This scaling is in good agreement with numerics :

[3] Let me consider only the scaling with p and throw away any dependency in Δ_0, γ or σ



(a) $d = 2 \Rightarrow \tilde{\beta} = 2/3$

(b) $d = 10 \Rightarrow \tilde{\beta} \approx 1/2$

Figure 14: Scaling of Δ with p, with a Laplace kernel : $\xi = 1$. The dashed horizontal lines represent (with the same color code) the values of Δ_0 : the theoretical prediction is only valid below these curves, where $\Delta \ll \Delta_0$. Note that the only curve that does not follow the black line is the blue one, for a gapless setup, where the prediction does not apply.

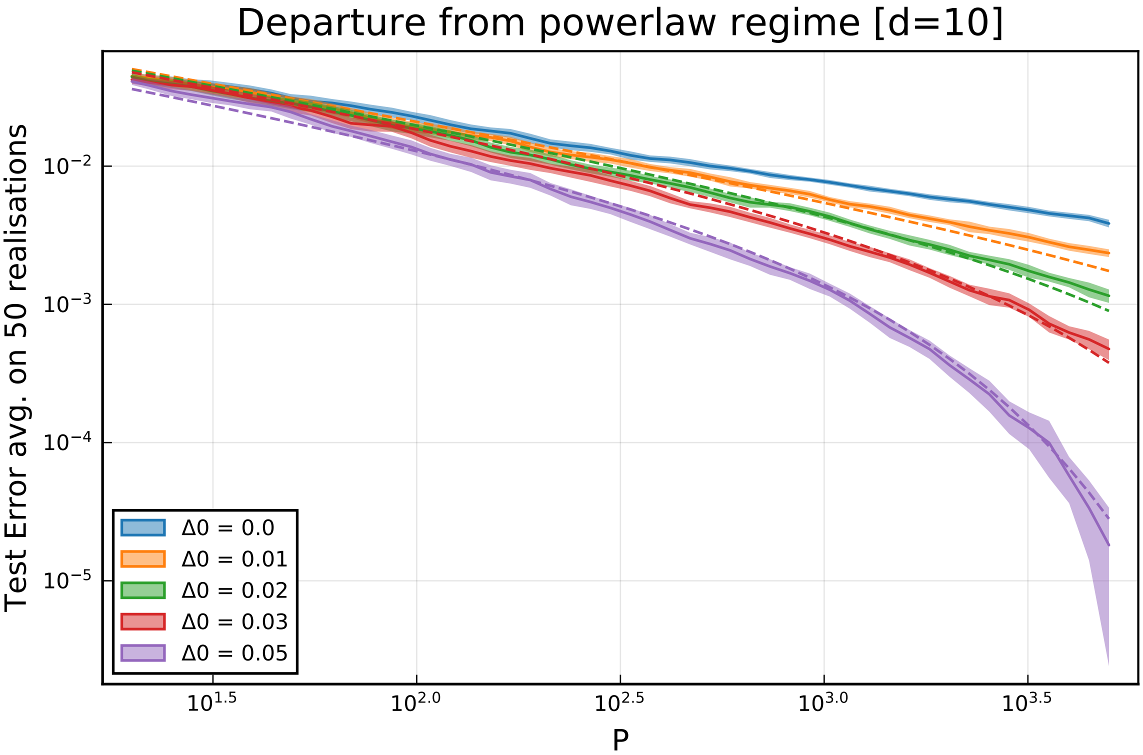


Figure 15: The dashed lines are the predictions $\epsilon \sim p^{-\tilde{\beta}} \operatorname{erfc}(\frac{\Delta_0}{2}\sqrt{p})$. [Laplace Kernel].

Note that the larger Δ_0, the better the fit ; indeed, for fixed p and as the gap size increases, one continuously change from the regime described in [3] and governed by the exponent β to the regime where $\Delta \ll \Delta_0$, governed by the exponent $\tilde{\beta}$. This is also why that the $\tilde{\beta}$-based fit for the blue curve had no sense and is therefore not included.

Finally, the same approach should work for a problem in which the interface is not 1D anymore. The maths are going to become more complex, but as long as the decision function lends itself to be modelled by a Gaussian Process, it should be (at least numerically) tractable to recover such a geometrically decreasing behaviour. For further mathematical details on excursion sets (in particular in higher dimensions), see the works of David Ginsbourger and Dario Azzimonti : [8] and [9] (especially the appendix).

3 Bounds on the test error for a Random Boolean classification task

This section's vocation is simply to set down on paper a few ideas, for which I have not had the time to conduct further investigations. The main idea is to explore the consequences of the following theorem [6], in particular when applying variational principle :

$$\epsilon \leq \mathbb{E}\ \phi\left[y\, f(x)\right] + \frac{B}{\gamma\, \sqrt{p}} + \left(\frac{8}{\gamma} + 1\right) \cdot \mathcal{O}(p^{-1/2})\ , \tag{31}$$

where ϵ is the classifier's test error, where $\mathbb{E}$ is the expectation value, where $\gamma \in \mathrm{R}$ (same γ as in the definition of ϕ, see below.), where B is the RKHS norm of the decision function $f(x)$ and p is the size of the trainset. ϕ is a cost function defined as follows :

$$\phi(z) \equiv \begin{cases} 1 & \text{if } z \leq 0 \\ 1 - z/\gamma & \text{if } 0 < z < \gamma \\ 0 & \text{if } z > \gamma \end{cases} \tag{32}$$

The argument of ϕ is negative if the point x is misclassified, in which case the cost is maximal (=1). If the point x is correctly classified but by less then a margin γ, one pays a cost of order 1. If the point x is correctly classified and beyond the margin γ, one pays no cost.

Also note that γ has to be of order 1, because the first term penalizes the case where $\gamma \gg 1$ while the last term penalizes the case where $\gamma \ll 1$.

RKHS Norm

The RKHS norm with reproducing kernel k of a function f is denoted by B and is defined as follows :

$$B^2 = ||f||_k^2 = \int\int dx\, dy\, f(x)\, k^{-1}(x,y)\, f(y) \tag{33}$$

which, for translation invariant kernels k can be rewritten as follows thanks to the Plancherel theorem :

$$B^2 = ||f||_k^2 = \int d\omega \frac{|\tilde{f}(\omega)|^2}{\tilde{k}(\omega)}\ . \tag{34}$$

Also recall that for Matérn kernels (that include the Laplace kernel for $\nu = 1/2$), one has

$$\tilde{k}(\omega) \sim \left(\frac{1}{\rho^2} + \omega^2\right)^{-(\nu+d/2)}\ , \tag{35}$$

where d is the dimension and $\rho = \rho(\nu)$ the characteristic length scale of the kernel.

3.1 Idea 1 : XOR should be easier to learn than AND

Before working with random boolean matrices attributing a random label from $\{\pm 1\}$ to each (hyper-)quarter of the hypercube in d dimensions, let's focus on the special case $d = 2$ and the logic gates XOR and AND :

$$\mathrm{AND}(\vec{x}) \equiv \begin{cases} +1 & \text{if } x_1 > 0 \text{ and } x_2 > 0 \\ -1 & \text{otherwise} \end{cases}\ , \quad \mathrm{XOR}(\vec{x}) \equiv \begin{cases} +1 & \text{if } x_1 \cdot x_2 > 0 \\ -1 & \text{otherwise} \end{cases} \tag{36}$$

Let's work in a setup without gap for the moment.

Let's try to find decisions functions that predict the correct signs for all $\vec{x} \in \mathrm{R}^2$. While XOR can be fitted by a simple polynomial $x_1\, x_2$, whose RKHS norm is finite, it is impossible to find a polynomial function that fits an AND gate : an easy choice would be the combination of 2 step functions but their RKHS norm would then be infinite, which would not help in bounding the test error. One can come up with

$$f(\vec{x}) \equiv x_2 - \beta\, e^{-\beta\, x_1} \;\text{ with } \beta > 0 \tag{37}$$

In the limit $\beta \longrightarrow \infty$, one recovers the AND decision function. What Thm (31) teaches us is that AND$(\vec{x})$ may not be the best decision function to fit an AND gate. The whole thing is about a tradeoff between the classification power of the decision function and its RKHS norm (corresponding to its smoothness/regularity). In what follows, I will use the variational principle to find optimal values for β and γ such that the right hand side (RHS) of Thm (31) is minimized.

3.1.1 AND Gate

Let's consider the decision function defined in Eq. (37). Since the Fourier transform of an exponential is not know, it will not be possible to compute its RKHS norm explicitly ; let's try to understand the scaling of B with β. Supported by [7], and backed up in Appendix C, let's suppose that for $\beta \gg 1/\sigma$ (where σ is the length scale of the kernel) :

$$||f||_k \sim \beta^{3/2}\, e^{\beta} \tag{38}$$

It exponentially diverges to ∞, conform to the visual intuition that f gets exponentially closer to an AND decision function as $\beta \longrightarrow \infty$.

Now comes the computation of the first term : since it is too complex to be handled analytically, one can understand its behaviour numerically. I performed the numerical integration of the expectation value with a simple Monte-Carlo procedure, with two different data distributions (called π) : $\mathcal{U}([-1,1]^2)$ and $\mathcal{N}(0, I_2)$: the results can be found in Fig. (16).

From this numerical clue, one can try to model the RHS of Thm (31). Let's have a try with

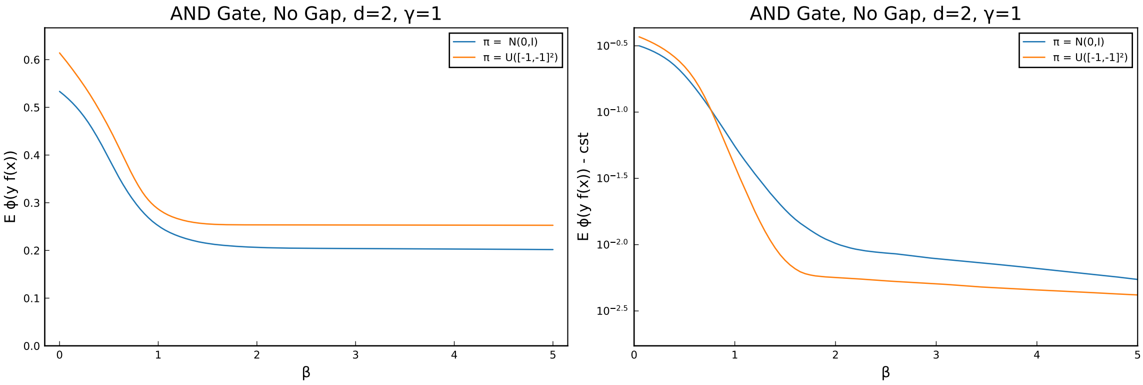
$$c_1\, e^{-\beta} + \mathrm{cst}(\gamma, \Delta_0, \pi) + \frac{\beta^{3/2}\, e^{\beta}}{\gamma\, \sqrt{p}} \;, \tag{39}$$

which, unfortunately does not have a tractable minimum for the variable β. It does have a solution, for p large enough, but obtaining a closed form seems impossible. This is due to the quite complex form of the Ansatz (38) under derivation. If one tries the simpler Ansatz

$$||f||_k \sim \beta\, e^{\beta} \tag{40}$$

then a closed form can be found and one can make use of the Lambert W function to express the minimum under variation of β, at γ fixed :

$$\beta_{opt} = \frac{1}{2}\, W(2\, c_1 \gamma\, \sqrt{p}\, e^2) - 1 \;, \tag{41}$$



(a) Linear scale. The expectation converges to a constant that depends on the data distribution.

(b) LogLin scale. The y axis is the function minus the constant found in left panel.

Figure 16: $\mathbb{E}\ \phi[y\, f(\vec{x})]$ decreases geometrically fast to a constant $\mathrm{cst}(\gamma, \Delta_0, \pi)$, where π is the data distribution.

which seems to grows as slowly as the numerical solution induced by Eq. (39).

In a future work, determining the behaviour of the constant with γ, Δ_0 and π would be a first step to be able to determine an upper bound to the test error.

3.1.2 XOR Gate

At first glance, one could think that the problem is easier because the XOR gate can be fitted by a function of intrinsically finite RKHS norm. Actually, the same approach applies. If one considers the decision function

$$f(\vec{x}) = \beta\, x_1\, x_2 \quad \text{, where } \beta > 0 \text{ ,} \tag{42}$$

one still has this tradeoff idea between the first term (the larger β, the smaller the expectation value) and the second term ($B \sim \beta$).

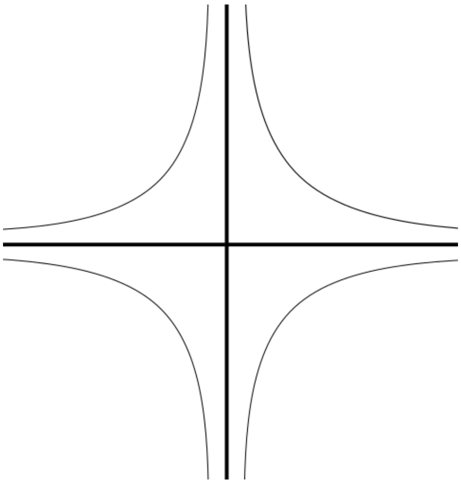


Figure 17: The data inside the star shape contributes to the expectation value. The top-right branch of the star is defined by $y = \gamma/(\beta x)$ and the others quarters similarly. Therefore, the larger β, the smaller the expectation value.

Here also, understanding the behaviour of the expectation value in this case would lead to an upper bound of the test error. I reckon it should be easier to solve that the AND case because there is no intertwining between γ and β.

3.1.3 Comparison of test errors

All in all, since Thm (31) is nothing but an upper bound of the error, one has to resort to actual numerics :

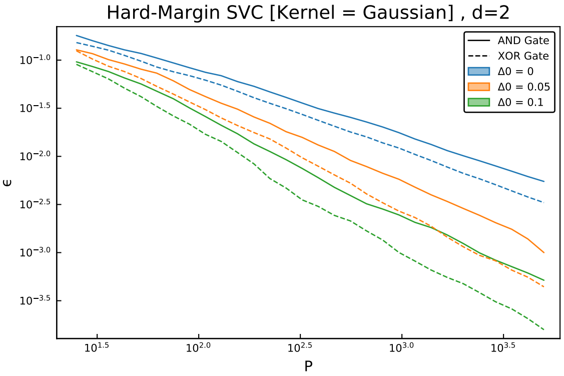


Figure 18: Learning curve on a LogLog scale : the XOR gate (dash) is indeed easier to fit that the AND gate (solid).

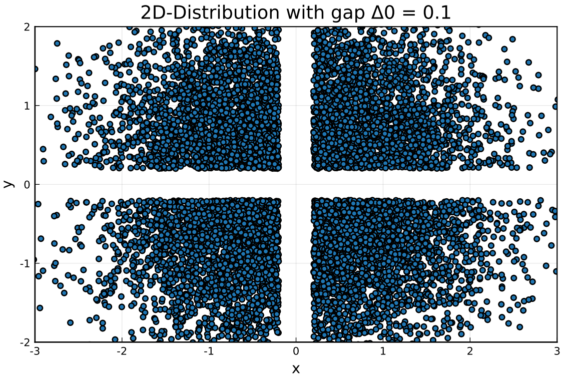


Figure 19: Illustration of the $\mathcal{N}(0, I_2)$ data distribution with gap.

Here, in contrast with what has been found in Section 2, the test error does not seem to decrease geometrically fast in presence of a gap. It would seem (at least in the range of p scanned in Fig. 18) that the change induced by the gap is a change in the slope in loglog $\rightarrow$ a bigger exponent of the power law. Note that Fig. 18 has been realised using a Gaussian kernel with the librairy's default length scale $\rho = 1$. Since for a 1D interface and labels $y(\vec{x}) = \text{sgn}(x)$, such a kernel did exhibit a geometrical decrease (cf. Fig. 12b), I do not think that it could be the reason why the geometrical behaviour disappeared. Solving the optimisation of both previous subsections would certainly provide an explanation to that phenomemon.

3.2 Idea 2 : A (loose) bound for ϵ in presence of a gap

Let's consider for simplicity the 1D classification, where the label $y(x) = \text{sgn}\ x$, with gap $2\,\Delta_0$ between labels. Any odd function like x or $\sin x$, sinh x, erf(x) would be a correct candidate for matching the labels. Let's go for $\sin x$ because of its simplicity in the Fourier space : We denote this decision function by $f^*(x) \equiv a\ \sin(\beta x)$, where $a > 0$ and $0 < \beta \leq \pi/(2L)$, L being the maximum spatial extension of the trainset . Since there is a gap, one pays no cost for $|x| < \Delta_0$. Therefore, let's construct a decision function such that $f^*(\Delta_0) > \gamma$, leading to the condition (after linearisation of the sine)

$$a\,\beta\Delta_0 \sim \gamma \tag{43}$$

This condition ensures that the first term of the Theorem is zero. Let's then focus on the RKHS norm of that function f^* :
Since $|\tilde{f}^*(\omega)|^2 \sim a^2\, \delta(\beta-\omega)$, one has

$$\begin{aligned} B^2 &= \int_0^\infty d\omega \; a^2\, \delta(\beta-\omega) \, \left(\frac{1}{\rho^2}+\omega^2\right)^{\nu+1/2} \\ &= a^2 \, \left(\frac{1}{\rho^2}+\beta^2\right)^{\nu+1/2} \\ &\sim \left(\frac{\gamma}{\Delta_0}\right)^2 \beta^{-2} \, \left(\frac{1}{\rho^2}+\beta^2\right)^{\nu+1/2} \quad . \end{aligned} \tag{44}$$

B^2 exhibits a minimum if and only if $\nu + 1/2 > 1$:

- Case $\nu > 1/2$ [4] and $\rho \geq L$: the minimum is reached for $\beta \sim 1/\rho$ and

$$\min_\beta B \sim \frac{\gamma}{\Delta_0} \rho^{\,\nu-1/2} \tag{45}$$

- Case $\nu \leq 1/2$ [5] or $\rho \leq L$: B is minimal ($B = 0$ or 1) for $\beta = \infty$, but since β is bounded by the spatial extension of the trainset (otherwise f^* does not even fit correctly the labels [6]), one has $\beta_{opt} = \pi/(2L)$, leading to

$$\min_\beta B \sim \frac{\gamma}{\Delta_0} \frac{1}{L} \left(\frac{1}{\rho^2}+\frac{1}{L^2}\right)^{(\nu-1/2)/2} \tag{46}$$

In any case, one has

$$\epsilon \leq \mathcal{O}(p^{-1/2}) \quad \text{independently of } \gamma \tag{47}$$

and therefore this approach fails at producing a tight bound. We saw in Section 2 that is presence of a gap, the test error could be expressed as a $p^{-\tilde{\beta}}\,\mathrm{erfc}(\Delta_0\,\sqrt{p})$. To recover such a scaling, one possibility is to choose f^* such that the first term (the expectation value) is zero and such that its Fourier transform satisfies [7]

$$\gamma^2\, p \left(\int_{\Delta_0\,\sqrt{p}}^\infty e^{-\omega^2}\, d\omega\right)^2 \sim \int_0^\infty |\tilde{f}^*(\omega)|^2 \left(\frac{1}{\rho^2}+\omega^2\right)^{\nu+1/2} d\omega \;. \tag{48}$$

This way, one would obtain $\epsilon \leq \mathrm{erfc}(\Delta_0\sqrt{p})$ which I guess is the tightest bound possibly reachable with this technique. However, I would be very surprised if such a f^* was found that way, there must be a more subtle approach.

[4] Includes Gaussian Kernel.

[5] Includes Laplace Kernel.

[6] Actually, if the data is rare for extreme values, it could be interesting to consider case where the sine changes sign before the end of a quarter but since it would requires the knowledge of the data's spatial distribution, I will put this possibility aside for now.

[7] Indeed, recall that $\mathrm{erfc}(x) \sim \int_x^\infty e^{-\omega^2}\, d\omega$.

3.3 Idea 3 : Symmetries and Polynomials

The following lines are nothing else but a blur idea. It might be provable that all the polynomials of the form

$$f(\vec{x}) = x_1\, x_2\, \cdots x_{d_{eff}} \quad \text{with } d_{eff} \text{ the number of relevant dimensions for labelling} \tag{49}$$

match a special symmetry in the labels. That symmetry could be the composition of a rotation followed by a reflection.

For instance :

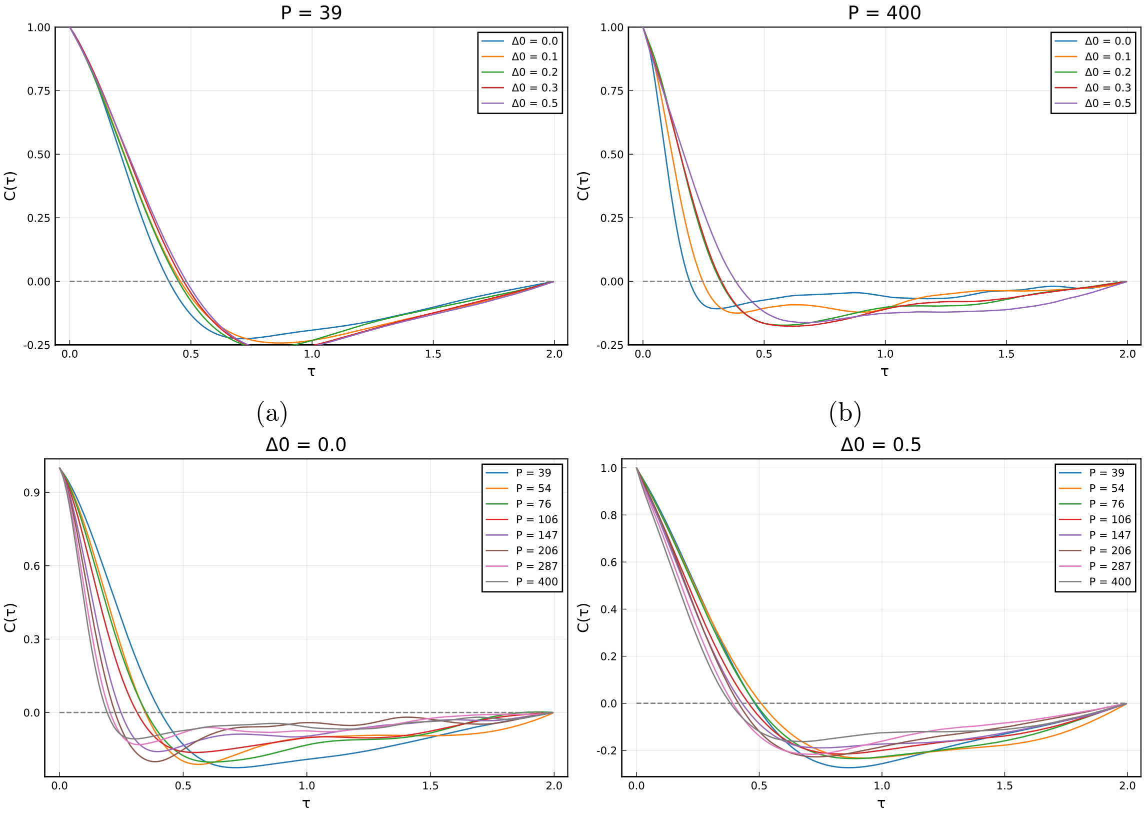
- In 1D, where the labels $y(x) = sgn(x)$, which can be matched by the decision function $f(x) = x$: the labels are left unchanged if one applies a rotation of π around the origin and then a reflection with respect to $x = 0$.
- In 2D, where the labels $y(x)$ are given by the XOR function, which can be matched by the decision function $f(\vec{x}) = x_1\, x_2$: the labels are left unchanged if one applies a rotation of $\pi/2$ around the origin and then a reflection with respect to any axis.

Since such polynomials of the form (49) have a finite RKHS norm, it could be a first step towards explaining how symmetries in data can be related to ease in learning/classifying.

More generally, it might be possible that symmetries in the data labels reflects in families of functions with more or less finite RKHS norms.

A Discussion on the underlying kernel of the Gaussian Process generating the decision boundaries

For this entire section, let's assume that the decision boundaries *do* arise from a Gaussian Process $GP(0, k)$ where $k(x, y) = k(|x - y|) \equiv k(h)$ is a positive definite, isotropic and translation invariant kernel.
From $M = 100$ independent realisations of the decision boundaries for multiple values of Δ_0 and several p, one is able to compute the ensemble average of their autocorrelation, leading to the underlying kernel k. Let's first present the raw data to then comment on different choices of fit :



(a)

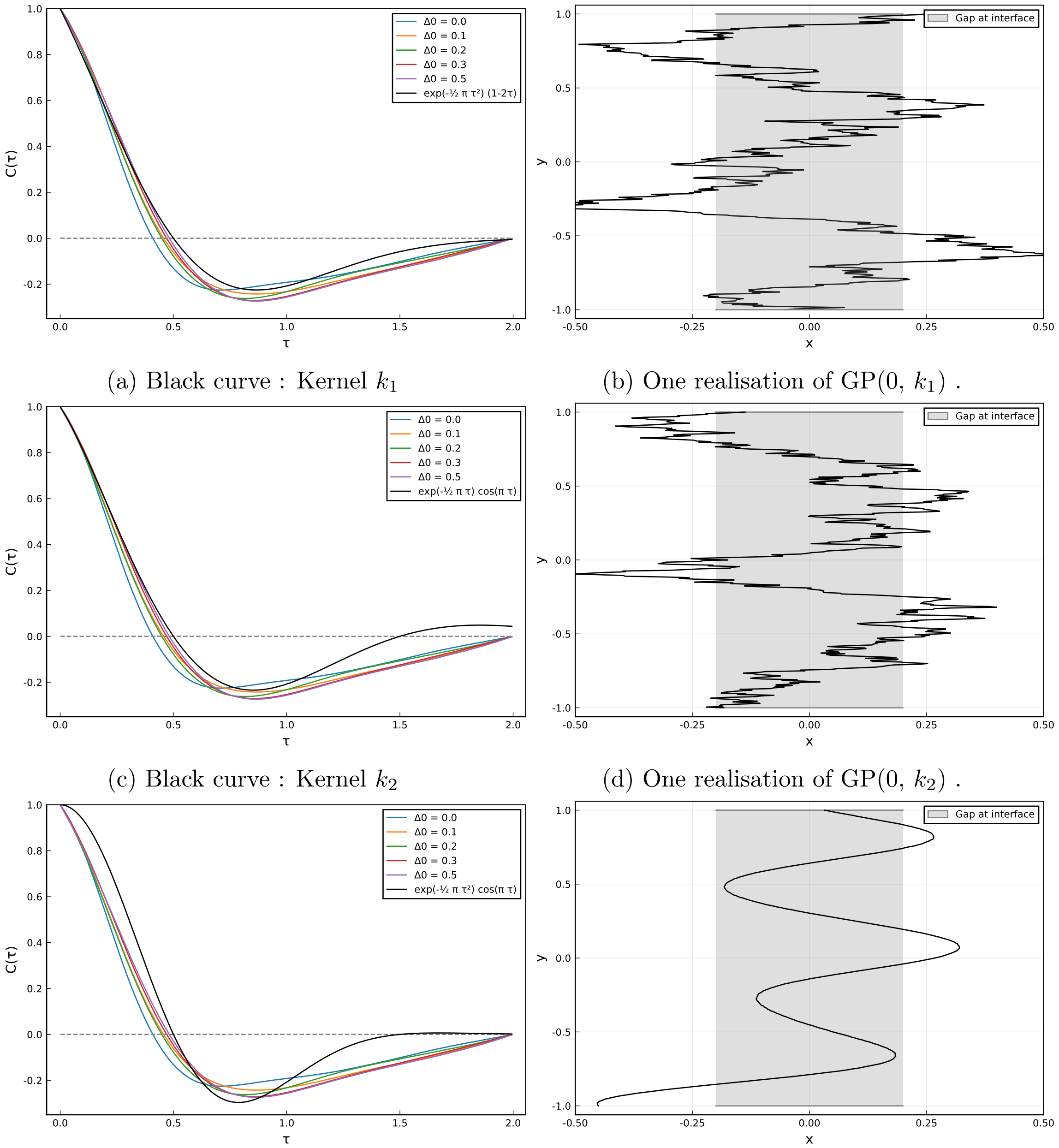
(b)

(c)

(d)

Figure 20: Autocorrelation functions (averaged over realisations) of decision boundaries from uniform data on $[-1, 1]^2$.

Let's try to find a positive definite, isotropic and translation invariant kernel that fits our data : one can came up with 3 different examples, but none is perfect ; see Fig. (21) :



(a) Black curve : Kernel k_1

(b) One realisation of GP(0, k_1) .

(c) Black curve : Kernel k_2

(d) One realisation of GP(0, k_2) .

(e) Black curve : Kernel k_3

(f) One realisation of GP(0, k_3) .

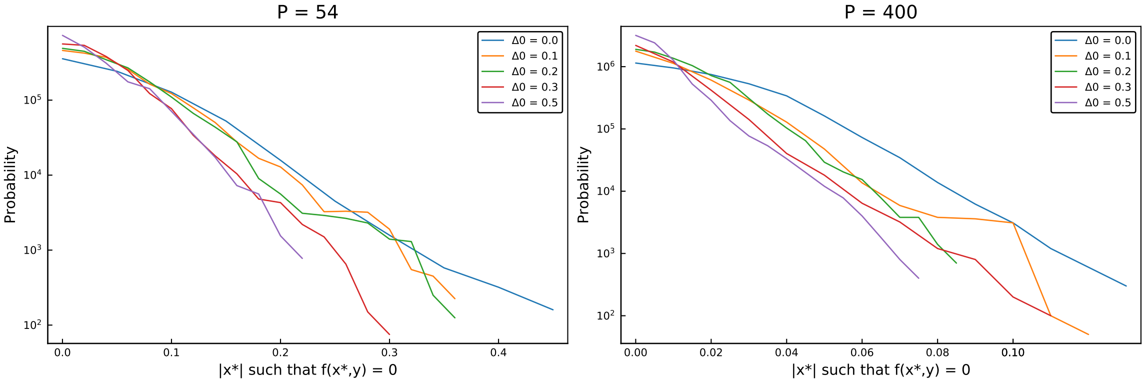
Figure 21: Left Panels : Visualisation of the kernels with data from Fig. (20a) for comparison. Right Panels : One random realisation of a GP with the kernel plotted on its left. Top panels : $k_1(h) = \exp\left(-\frac{\pi}{2}\,|h|\right)\,(1-2h)$. Middle panels : $k_2(h) = \exp\left(-\frac{\pi}{2}\,|h|\right)\,\cos(\pi\,h)$. Bottom panels : $k_3(h) = \exp\left(-\frac{\pi}{2}\,h^2\right)\,\cos(\pi\,h)$.

The issue is obvious : the kernels that best match the data produce very rough realisations. Yet, the numerical experiments show that the decision boundaries are very smooth (up to the grid discretisation precision).

All in all, as discussed after Eq. (25), the actual nature of the kernel is not relevant as long as it is translation invariant (hence the fact that this discussion was postponed in Appendix) but the topic is of interest for a full understanding.

B Amplitude Histograms for decision functions

The following figures represent statistics of $M = 100$ independent realisations of f_0 when Data $\sim$ Uniform$\big([-1,1]^2\big)$. All 4 figures are in LogLin scale, the y axes are more or less identical but the x axes vary a lot from one plot to another.

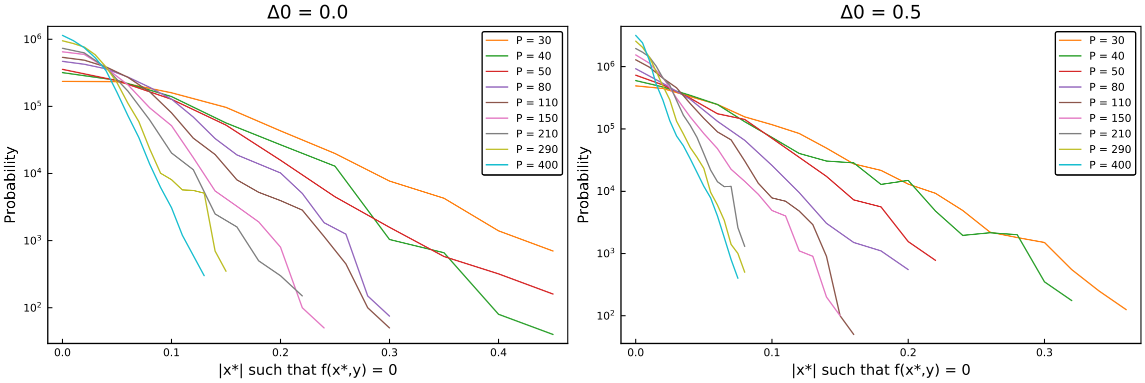


(a) LogLin pdf

(b) LogLin pdf

Figure 22: Histograms at p fixed for several Δ_0.

It would appear that the slope of these exponential distributions is independent of the gap Δ_0.



(a) LogLin pdf

(b) LogLin pdf

Figure 23: Histograms at Δ_0 fixed for several p.

C Justification of the Ansatz $||\beta\, e^{-\beta x}||_k \sim \beta^{3/2}\, e^{\beta}$

Based on Chapter 7.4 ; Example 17 of [7].
For a Sobolev Space $H^1(L,-L)$ with reproducing kernel $k(x,x') = \exp(-|x-x'|/\sigma)$ the norm of a function u is given by

$$||u||^2_{\mathcal{H}} = \frac{1}{2}\left(u(-L)^2 + u(L)^2\right) + \frac{\sigma}{2}\int_{-L}^{L} u'(t)^2 + \frac{1}{\sigma^2}\,u(t)^2\;dt \tag{50}$$

Now if $u = \exp(-\beta x)$, one has

$$||u||^2_{\mathcal{H}} = \cosh(2L\beta) + \frac{\sigma}{4\beta}\left(\frac{1}{\sigma^2} + \beta^2\right)\sinh(2L\beta) \tag{51}$$

which, for $\beta \gg 1/\sigma$ (which is insured if one takes $\sigma \gg 1$), boils down to

$$||u||^2_{\mathcal{H}} \sim (1+\sigma\,\beta)\,e^{2L\beta} \iff ||u||_{\mathcal{H}} \sim \sqrt{\beta}\,e^{\beta} \quad \text{for } L \text{ fixed.} \tag{52}$$

Thus, scaling the entire result by β leads to the Ansatz used in Eq. (38).

References

[1] Spigler, Geiger and Wyart (2019)
Learning Curves of Kernel Methods, empirical data vs. Teacher/Student paradigm

[2] Bordelon, Canatar and Pehlevan (2020)
Spectrum dependent learning curves in kernel regression and wide neural networks

[3] Paccolat, Spigler and Wyart (2020)
How isotropic kernels learn simple invariants

[4] Pillaud-Vivien, Rudi and Bach (2018)
Exponential Convergence of Testing Error for Stochastic Gradient Methods

[5] Caponnetto and De Vito (2007)
Optimal Rates for the Regularized Least-Squares Algorithm

[6] Bartlett and Mendelson (2002)
Rademacher and Gaussian Complexities

[7] Berlinet and Thomas-Agnan (2004)
Reproducing Kernel Hilbert Spaces in Probability and Statistics

[8] Azzimonti (2016)
Contributions to Bayesian set estimation relying on random field priors

[9] Fossum et al. (2020)
Learning excursion sets of vector-valued Gaussian random fields for autonomous ocean sampling

[10] Gonzalez (2009)
Measurement of areas on a sphere using Fibonacci and latitude–longitude lattices