PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure – Inria

Optimization, Generalization and Non-convex Optimization with Kernel Methods

Soutenue par
Gaspard Beugnot
Le 5 avril 2024

Ecole doctorale n° 386
Sciences mathématiques de Paris Centre

Spécialité
Mathématiques

Composition du jury :

Alain Célisse Sorbonne Panthéon	*Rapporteur*
Julien Mairal Inria	*Directeur de thèse*
Nicole Mücke TU Braunschweig	*Examinatrice*
Edouard Pauwels Toulouse School of Economics	*Président - rapporteur*
Lorenzo Rosasco Universita' di Genova	*Examinateur*
Alessandro Rudi Inria	*Co-encadrant de thèse*

Inria

Optimization, Generalization and Non-convex Optimization With Kernel Methods

Gaspard Beugnot

March 2024

PhD Thesis

Remerciements

Je suis toujours curieux de voir les Remerciements d'un manuscrit de thèse. On y vient en général pour trouver une équation, un théorème ou une revue de littérature, en espérant avec cet énième point de vue gagner une compréhension un peu moins vague du concept abstrait qui nous taraude sur le moment. Et pourtant, cette partie-là offre une escale amusante, car on y voit pendant quelques lignes un peu de la personne qui a fourni ce travail, avec des tuteurs ou tutrices de thèses, des amis et de la famille, qui y apparaissent tour à tour comme des puits de science, de formidables rocs dans les tempêtes ou des compagnons infaillibles d'une traversée de quelques années dans le monde académique. Alors, rassuré de ce trait d'humanité, je survole le reste du document pour aller au chiffre, la valeur ou le résultat qui m'intéresse vraiment.

Je souhaiterais remercier d'abord mes tuteurs, Alessandro et Julien, pour leur aide et conseils tout au long de ma thèse. Vous m'avez chacun solidement épaulés et redonné la motivation qui me manquait quand il fallait.

Merci bien sûr aux membres du jury : aux examinateurs Nicole Mücke et Lorenzo Rosasco, ainsi qu'aux rapporteurs qui ont bien voulu me relire, Alain Célisse et Edouard Pauwels.

Merci à tous les chercheurs qui ont pu me donner des conseils à différents moments : Francis, Justin, Stéphane et Adrien notamment. Un merci tout particulier pour Stéphanie Allassonnière, qui m'a pris sous son aile pour mon projet de troisième année à polytechnique, et qui a véritablement éveillé chez moi le goût pour le machine learning, sans lequel je ne serai pas arrivé ici.

Merci bien sûr aux autres membres du labo, à Paris et Grenoble. L'ambiance amicale qui y régnait était propice aux discussions quand la motivation manquait. Un merci tout particulier à Bruno, Céline, Fabian et Lawrence, et bien entendu, au Super Bureau : Ziad, Antoine, Théophile et Bertille. Vous étiez sans aucun doute les raisons de mon assiduité pour venir, car j'étais sûr d'y trouver un sourire, de gentilles moqueries ou de grands débats inutiles sur les manoirs en Normandie ou la façon de ranger son bureau.

Merci du fond du cœur à mes amis d'avant pour m'avoir accompagné de près ou de loin pendant ces trois ans. Et bien sûr, merci à ma famille, grands-parents, beaux-parents, parents, frères et sœurs et beaux-frères et belle-sœurs. Votre soutien, admiration et amour rendent tous mes autres problèmes bien secondaires en comparaison. Merci à tous mes petits neveux et nièces pour me laisser jouer aux kaplas de temps en temps, me donnant l'occasion de transmettre subtilement quelques principes de mathématiques.

Merci enfin à Célia, pour ton soutien constant, tes conseils et tes rires. Rien ne saurait être trop difficile avec toi à mes côtés.

Résumé

Cette thèse étudie le lien entre généralisation et optimisation dans les algorithmes d'apprentissage machine, par le biais des méthodes à noyaux reproduisant. D'abord, dans un cadre de théorie de l'apprentissage sur des fonctions convexes, nous présentons des bornes d'excès de risques améliorées pour les fonctions auto-concordantes, qui incluent l'erreur logistique. Ce résultat étend la théorie du filtrage spectral, traditionnellement utilisée pour obtenir des bornes d'excès de risque pour plusieurs types de régularisation avec les moindres carrés, à cette plus grande famille de fonctions. Cela permet d'avoir des estimateurs avec des taux d'apprentissages plus rapides et optimaux, là où la régularisation de Tikhonov ne l'est pas. Ensuite, nous étudions un curieux phénomène qui a lieu lors de l'entraînement de réseaux de neurones par descente de gradient, où prendre de grands pas de gradients permet de mieux généraliser, au détriment de l'optimisation du risque empirique. Nous développons un modèle intuitif avec des fonctions convexes dans un espace de Hilbert, et nos résultats démontrent que de grands pas de gradients permettent effectivement de mieux généraliser, notamment dans des tâches de classification. Enfin, nous proposons un algorithme pour donner des certificats d'optimalité sur des minima de fonctions non-convexes définies sur le tore ou l'hypercube. Notre algorithme utilise le pouvoir de représentation des fonctions positives par les sommes de noyaux reproduisant pour obtenir des certificats à des problèmes qui ne peuvent pas être traités par les méthodes existantes.

Abstract

This thesis studies the interplay between generalization and optimization in machine learning algorithms, through the lens of kernel methods. We first cover learning theory in classical convex settings, presenting improved excess risk bounds for generalized self-concordant functions, which include the logistic loss. This effort extends the utility of spectral filter theory—-traditionally used to provide risk bounds for various regularization methods with square loss—-to a broader range of loss functions. Our approach yields estimators with fast and optimal rates for a wider array of problems compared to traditional Tikhonov regularization. The next section investigates a notable phenomenon in neural network training with square loss: the advantageous role of high learning rates in generalization, despite their adverse effect on training loss optimization. We develop an intuitive model using convex functions in a Hilbert space to explain this. Our results indicate that high learning rates can improve generalization performance, applicable not only when there's a disparity between training and testing objectives but also in typical classification tasks. Finally, we design an algorithm for providing certificates of optimality for the minimization of non-convex functions on the torus or the hypercubes. Our algorithm harnesses the strength of kernel sum-of-squares to effectively model positive functions, offering certificates for challenges unaddressed by existing methods.

Outline of the Thesis

We briefly summarize the structure of the thesis, which is divided into four chapters.

- In the Introduction (p. 6), we establish the broader context of this thesis, detailing the motivations driving this research. We then highlight our main contributions, before detailed discussions in the following chapters.
- In Part 1 (p. 18), we review the spectral filter theory, an elegant tool to compare regularization scheme in regression task with kernels. We then broaden this theory to include general loss functions in the context of the proximal point algorithm. This part is grounded in the work *Beyond Tikhonov: Faster Learning with Self-Concordant Losses via Iterative Regularization*, spotlighted at NeurIPS 2021.
- Part 2 (p. 57) explores the connection between optimization and generalization in machine learning algorithms. Specifically, we give a model for explaining the effectiveness of large learning rate for generalization when training neural network with gradient descent. This builds on *On the Benefits of Large Learning Rates for Kernel Methods*, published at the Conference on Learning Theory (2022).
- Finally, part 3 (p. 81) tackles non-convex optimization with certificates. We first review existing approaches, provide new research directions which would be fruitful to explore, before introducing the *GloptiNets* algorithm. This last bit is based on the paper *GloptiNets: Scalable Non-Convex Optimization with Certificates*, published at NeurIPS (2023) as a spotlight.

Specific attention is devoted to highlighting the open research problems we did not have time to tackle.

Contents

Introduction

1 A high overview on learning theory

Learning theory, one of the central theme of this thesis, focuses on the fundamental question, "To what extent can we derive general knowledge from a few observations?" This question embodies the essence of *inductive* reasoning. Thus, learning theory is concerned with the process of generalization - extrapolating new information from existing observations. An illustrative example of this concept is the task of completing sequences based on observed patterns, such as deducing that the next number in "1 -> 2, 2 -> 4, 3 -> 6, 4 -> ..." is "8," recognizing the pattern of doubling each number. Similarly, identifying "Berlin" as the capital of Germany from the sequence "France -> Paris, Spain -> Madrid, Germany -> ..." involves discerning the pattern of "Country -> Capital".

In its simplest form, machine learning involves designing computer programs that process training pairs (like "1 -> 2" or "France -> Paris") and output a function that captures the underlying pattern in the training data. This process, known as *training*, aims to develop algorithms that, once exposed to available data, can generalize effectively. Generalization, in this context, means providing reliable insights into unseen data.

In practical scenarios, we typically have access to a dataset that we divide into two parts: a training set and a test set. We develop a function based on the training set - this is the training phase of the model - and then assess its generalization capabilities on the test set, which contains previously unseen data. This approach is largely *empirical*. The algorithm's ability to generalize, or its effectiveness in learning, is gauged at the end of this process, before the model is deployed in real-world applications.

Learning theory, however, takes a different approach. Instead of starting with data, it begins with assumptions about the nature of the learning problem. The theory posits that training examples follow certain well-defined patterns. This assumption-driven approach offers several advantages. *(i)* It allows us to gauge the speed of learning for different machine learning algorithms. We can predict how well a model will generalize based on the number of samples it has been trained on. *(ii)* We can mathematically quantify the difficulty of a learning task. For example, a challenging task will slow down the learning process regardless of the algorithm used, necessitating a large number of samples for effective generalization. Conversely, an easy task might require only a few examples to discern the underlying pattern, like in the "1 -> 2, 2 -> 4, 3 -> 6" example. Finally, *(iii)* This framework provides a basis for comparing different machine learning algorithms. An effective algorithm is one that generalizes better than others given the same number of training samples.

1.1 Typical results

In the realm of learning theory, our primary focus is on supervised learning, specifically considering regression tasks as a key example. However, learning theory extends beyond this scope, encompassing a broad range of predictive modeling challenges.

At the core of this exploration is the assumption that data pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are generated from an unknown distribution ρ. In our framework, we typically define $\mathcal{X}$ as a space like $\mathbb{R}^d$, and $\mathcal{Y}$ as $\mathbb{R}$. We work with a set of n training examples $(x_1, y_1), \ldots, (x_n, y_n)$, assumed to be i.i.d samples drawn from ρ. Note that this is already a restrictive hypothesis, as real-world data may be correlated or with missing information.

The central objective in this context is to discover a function $\theta : \mathcal{X} \to \mathcal{Y}$ that can accurately predict y for a given x from the distribution ρ. This function, termed as an estimator or predictor, is the crux of learning algorithms. Formally, a machine learning algorithm is a function

$$\begin{aligned} \mathcal{A} : \qquad & (\mathcal{X} \times \mathcal{Y})^n \to (\mathcal{X} \to \mathcal{Y}) \\ & (x_i, y_i)_{1 \leqslant i \leqslant n} \mapsto \theta \end{aligned}$$

Note that since the training data is random, so is our estimator θ.

We thus hope that $\theta(x) \approx y$ for $(x, y) \sim \rho$. To measure the accuracy of our predictor, we employ a loss function, typically represented as $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. A common example is the square

loss, defined as $\ell(\theta(x), y) = 1/2\,(\theta(x) - y)^2$. Our ultimate aim is risk minimization: reducing the expected loss of our estimator θ. This is quantified as $\mathcal{R}(\theta) = \mathbb{E}_{x,y\sim\rho}[\ell(\theta(x), y)]$, representing the average accuracy of θ over all possible data instances. In pursuit of the optimal predictor, we target approximating the minimizer of $\mathcal{R}$, denoted as $\theta^\star$. In the case of the square loss, this ideal predictor is known as the "Bayes predictor" and is defined as $\theta^\star(x) = \mathbb{E}[y|x]$. Learning theory is all about measuring the excess risk, $\mathcal{R}(\theta) - \mathcal{R}(\theta^\star)$, which represents the difference in risk between the chosen estimator and the optimal one. This metric essentially gauges how far our model is from achieving theoretical perfection.

To derive meaningful conclusions in learning theory, we first make assumptions about the nature of the optimal model $\theta^\star$ and the resources available to the machine learning algorithm. These assumptions are mandated by the no-free-lunch theorem, which states that without them, no concrete lower or upper bounds on excess risk can be established.

The lower bounds on excess risk are of the form

$$\forall \mathcal{A}, \exists \theta^\star,\ \ \mathcal{R}(\mathcal{A}((x_i, y_i)_i)) - \mathcal{R}(\theta^\star) \geqslant C n^{-\alpha} \ \text{ with proba. } 1 - \delta$$

where C and δ are some constant. The exponent α symbolizes a baseline learning rate, akin to a universal constant. It serves as an indicator of a fundamental limit on learning efficiency.

Conversely, upper bounds on excess risk demonstrate the capacity of certain algorithms to utilize additional data effectively. For a given ML algorithm $\mathcal{A}$, they indicate that

$$\forall \theta^\star,\ \ \mathcal{R}(\mathcal{A}((x_i, y_i)_i)) - \mathcal{R}(\theta^\star) \leqslant C n^{-\gamma} \ \text{ with proba. } 1 - \delta.$$

The parameter γ in this context represents the learning rate of the algorithm, reflecting its efficiency in leveraging additional information from the data to improve generalization.

An algorithm reaches optimality when $\gamma = \alpha$, signifying that its practical performance matches the theoretical lower limit. In such cases, the algorithm has maximized its learning efficiency given the available data, and no further improvements are possible within the confines of the assumed model and data constraints. This threshold of optimality is a cornerstone in assessing the efficacy of machine learning algorithms in the realm of supervised learning.

1.2 Why do we need learning theory?

Learning theory's relevance in machine learning, particularly in practical scenarios, might initially seem questionable due to the inherent limitations in testing the assumptions about the learning problem. In real-world applications, the target knowledge or patterns we aim to learn (*i.e.* the assumptions on $\theta^\star$) are unknown, making it impossible to validate these foundational assumptions directly. This might lead to the misconception that learning theory has limited practical utility.

Understanding generalization, designing efficient algorithms, enabling objective benchmarking and comparisons, and managing realistic expectations all derive their foundation from theoretical principles. Learning theory elucidates how algorithms can adapt from training scenarios to novel data, a key predictor of real-world performance. It guides the creation of algorithms that balance effectiveness with computational efficiency, addressing scalability challenges in processing complex datasets. Additionally, it provides a framework for comparing various algorithms, assessing their relative strengths and effectiveness. This approach helps in setting pragmatic boundaries for algorithmic capabilities, considering data and resource limitations.

In the realm of deep learning, the lack of comprehensive explanation through learning theory highlights its necessity. The struggle to fully understand deep learning's generalization capabilities results in reliance on empirical methods and extensive parameter tuning. This scenario underscores the importance of learning theory in developing a solid theoretical foundation, crucial for advancing machine learning algorithms and practices.

2 Generalization, regularization and optimization

Machine learning fundamentally revolves around the ability to generalize knowledge from a set of observations. This central challenge can be modeled through the relationship between two functions.

Definition 1 (Machine Learning problem)**.** *We introduce two functions* R *and* F *which characterize a machine learning problem.*

- *The* risk *or* generalization error *is a function* R *that measures how well we generalize to unknown samples but which we do not have access to;*
- *the* optimization objective *is a surrogate function* F *related to* R*, which we can optimize.*

In supervised learning, R typically represents the expected risk, formulated as:

$$R(\theta) = \mathbb{E}_{x,y\sim\rho}\left[\ell_{\text{test}}(\theta(x), y)\right],$$

where ℓ_{test} is the loss function measuring generalization abilities. In practice, R is approximated by an empirical average over a held-out test set

$$R(\theta) \approx \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \ell_{\text{test}}(f(x_{\text{test},i}), y_{\text{test},i}).$$

Conversely, F relates to the empirical risk, commonly involving the minimization of a regularized version of this risk:

$$F(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{train}}(\theta(x_i), y_i) + \lambda\Omega(\theta),$$

where $\Omega(\theta)$ measures the complexity of the estimator θ, and λ is a regularization parameter. This formulation embodies the classic tradeoff in machine learning: fitting the training data (gaining knowledge from observations) while avoiding to overinterpret the noisy data, a balance between underfitting and overfitting.

Example 1 (Ordinary Least Squares)**.** *The interconnected nature of these concepts can be highlighted through the lens of ordinary least squares, a simple yet illustrative example. In this context,* $\theta(x) = \omega^\top x$*, with* $R(\theta)$ *and* $F(\theta)$ *defined as follows*

$$R(\theta) = \mathbb{E}[1/2(\omega^\top x - y)^2],$$
$$F(\theta) = \frac{1}{n} \sum_{i=1}^{n} 1/2(\omega^\top x_i - y_i)^2 + \lambda/2 \left\|\omega\right\|^2.$$

The optimization problem's solution in F *is given with*

$$\widehat{\theta}_\lambda = \frac{1}{n}\left(\frac{1}{n}X^\top X + \lambda\mathbb{I}\right)^{-1} X^\top y$$

Computing $\widehat{\theta}_\lambda$ *involves solving an inverse problem. The difficulty of solving this problem is related to the condition number of the linear operator* $(1/n X^\top X + \lambda\mathbb{I})$*. This condition number is dependent on the regularization parameter* λ*: if the spectral decomposition of* $1/n X^\top X$ *is* $\sigma_1 > \cdots > \sigma_n$*, then the condition number takes the form* $\frac{\sigma_1+\lambda}{\sigma_n+\lambda}$*.*

As such, the larger the value of λ*, the smaller the condition number becomes, simplifying the inverse problem. However, there is a critical tradeoff here: while a specific* λ *might minimize* $\lambda \mapsto R(\widehat{\theta}_\lambda)$*, if it is too small, it can render the optimization problem challenging to solve with desired precision. In such cases, alternative forms of regularization might be more appropriate. This highlights the delicate balance between choosing a* λ *that aids in generalization while ensuring the tractability of the optimization process.*

In summary, modern machine learning involves simultaneously tackling the dual problems of optimization and generalization, with the choice of regularization playing a pivotal role in influencing both aspects. The intricacies of these interrelationships are crucial for understanding the development and performance of machine learning algorithms.

3 Contribution of the thesis

Based on the formulation introduced in definition 1 in section 2, we now introduce the contribution of this thesis:

1. In *Beyond Tikhonov* [2], we extend the spectral filter theory to a more general class of loss function for a given regularization. That is, we want to minimize R which is the expected risk for a *generalized self-concordant* loss ℓ; we have access to F which is the empirical risk with the same loss ℓ; we optimize with the *proximal point algorithm* with a regularization parameter λ. This encompass the classical settings of logistic regression, highlights the limitations of the usual Tikhonov regularization and shows how the proximal point algorithm circumvents those limitations.

2. In *The Benefits of Large Learning Rate* [3], we offer an explanation for the good generalization induced by large learning rate in gradient descent when training neural networks. On a convex model, we prove that specific discrepancies between F and R generates this benefits for large learning rate. We explain why it already occurs in classification tasks without assuming any particular mismatch between train and test data distributions.

3. In *GloptiNets* [4], we leave aside the statistical problem (*i.e.* optimizing R) and we focus on optimizing a function F with certificates, *without a convexity assumption*. Building on previous work on kernel Sum-of-Squares [11], we propose a new algorithm dubbed *GloptiNets* which solve problems which are untractable for the competitors.

3.1 Beyond Tikhonov: Extending spectral filter theory to more general loss functions

This work is covered extensively in section 4 of part I.

In this work, we are interested in providing *probabilistic upper bounds on the excess risk*. We define R to be the expected risk, and F to be the empirical risk, as in

$$R(\theta) = \mathbb{E}_{x,y\sim\rho}\left[\ell(y,\theta(x))\right], \qquad F(\theta) = \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\theta(x_i)). \tag{1}$$

We assume that θ and $\theta^\star$ – the minimizer of the risk R, *i.e.* the best estimator we can hope for – are both elements of some Hilbert space $\mathcal{H}$. The aim is to derive an estimator $\widehat{\theta}$ from n training samples s.t.

$$R(\widehat{\theta}) - R(\theta^\star) \leqslant C_1 n^{-\gamma}\log\tfrac{2}{\delta} \quad \text{with confidence } 1-\delta. \tag{2}$$

As stated earlier, γ is the rate of convergence of the estimator. The quality of this rate of convergence is measured by comparing it to lower bounds obtained on family of distribution ρ. An algorithm is deemed optimal when this rate matches the lower bounds.

When the loss ℓ is set to least-squares, extensive literature offers various optimal algorithms differing in the applied regularization [5]. A surprising outcome of this analysis is that not all regularization methods are equally effective: some excel in "easy" learning tasks (to be defined), while others fail to leverage the additional regularity to enhance the learning rate. Such holistic analysis is made possible by the fact that for the square loss, optimizing F amounts to solving an ill-posed inverse problem, with a solution in closed-form.

When considering other loss functions ℓ, the estimator $\widehat{\theta}$ is only available as the solution of an optimization problem involving the empirical risk F. Yet, Marteau-Ferey et al. [10] managed to extend the results from least squares to generalized self-concordant loss functions in the specific case of Tikhonov regularization. These loss functions are three-times differentiable function whose third derivative bounded by the second derivative. Their analysis is motivated by the fact that they contain logistic regression, a loss widely used for classification.

While a positive step towards understanding the generalization abilities of logistic regression, Tikhonov regularization is known to *saturate* on *easy* learning task. There is thus a need to provide an optimal algorithms for these settings too. A natural candidate is the "**Iterated**

Tikhonov (IT) estimator", which is known to overcome the limitations of Tikhonov estimator for least square. This estimator consists in iteratively fitting the residual. This definition is specific to the square loss, but we generalize it by mean of the proximal operator. An unexpected side-effect is that the resulting estimator is the proximal point algorithm, which is widely known in the optimization community.

Definition 2 (Definition 1 from [2])**.** *Given* $\lambda > 0$ *and* $\widehat{\theta}_\lambda^0 = 0$*, define the iterated Tikhonov estimator (a.k.a the proximal point algorithm) with*

$$\widehat{\theta}_\lambda^{t+1} = \mathsf{prox}_{F/\lambda}(\widehat{\theta}_\lambda^t) \stackrel{\text{def.}}{=} \operatorname*{arg\,min}_{\theta \in \mathcal{H}} \left\{ F(\theta) + \frac{\lambda}{2} \left\| \theta - \widehat{\theta}_\lambda^t \right\|^2 \right\}, \tag{3}$$

where $\mathsf{prox}_{F/\lambda}$ *denotes the proximal operator of the empirical risk* F *rescaled by* $1/\lambda$*.*

Note that with $t = 1$, we recover Tikhonov estimator studied in [10].

3.1.1 Difficulty of a learning task with the source/capacity conditions

To gauge a learning task's difficulty, we use the *source* and *capacity* condition, with respective parameters $r \geqslant 0$ and $\alpha \geqslant 1$, with larger parameters indicating an easier task. A formal definition is given in our paper (in section 4). Introduced in inverse problems, these conditions are particularly illustrative with the square loss, examining the spectrum of the *covariance operator* $T = \int_x \phi(x) \otimes \phi(x) d\rho_x(x)$, a p.d. operator of $\mathcal{H}$.

- the capacity condition is an assumption on the decay of the spectrum of the covariance operator. The bigger α, the faster the decay. In other words, a large α in the capacity condition implies that few eigenvectors are needed in expectation to approximate an element $\phi(x) \in \mathcal{H}$ of some $x \sim \rho_x$;
- the source condition is an assumption on how well does the optimum $\theta^\star$ aligns on the eigenvectors of the covariance matrix.

These source and capacity condition were generalized to self-concordant loss functions in [10], and form our core assumptions (see assumptions 3 and 4). Intuitively, they amount to replacing the covariance operator T with the Hessian of the risk at $\theta^\star$, *i.e.* considering the operator

$$\mathbf{H}(\theta^\star) = \mathbb{E}_{x,y \sim \rho} \left[\nabla^2 \ell(\theta^\star(x), y) \right].$$

3.1.2 Contributions

With those notions introduced, we can state our main result:

Theorem (Informal version of theorem 1 in section 4)**.** *Assume that the source condition holds with parameter* $r \geqslant 0$ *(assumption 3), and that the capacity condition holds with parameter* $\alpha \geqslant 1$ *(assumption 4). Let* $\delta \in (0, 1)$ *and* λ *be an appropriate regularization parameter. Under various additional technical assumptions detailed in section 4, the following bound on the excess risk for the Iterated Tikhonov estimator with* t *steps holds with confidence* $1 - \delta$

$$R(\widehat{\theta}_\lambda^t) - R(\theta^\star) \leqslant 2C_{\text{risk}}\, n^{-\frac{2\alpha s}{1+2\alpha s}}, \quad s = \min\{r + 1/2, t\}. \tag{4}$$

The constants C_{risk} *is detailed in theorem 4; it is explicit and depend only on* r, α, t, δ *and the distribution* ρ*.*

Saturation. The value of s illustrates the saturating effect we mentioned earlier. Indeed, assume that $r \geqslant t - 1/2$. In that case, the learning rate is $2\alpha t/1+2\alpha t$ which is smaller than $\alpha(2r+1)/1+\alpha(2r+1)$. The latter, obtained when $t \geqslant r + 1/2$, turns out to be optimal for the prior considered on ρ, *via* the source and capacity condition, as it matches the lower bounds known for the square loss. Said differently, not making sufficiently many steps with IT estimator on *easy* learning tasks results in a suboptimal estimator.

Comparison with [10]. Recalling that setting $t = 1$ amounts to using Tikhonov estimator, we see that we fully generalize the results from [10]. Specifically, if the source condition satisfies $r \geqslant 1/2$, then Tikhonov estimator will suffer from a suboptimal rate, which can be overcome by IT estimator with $t \geqslant r + 1/2$.

For instance, if the capacity condition does not hold (which amounts to setting $\alpha = 1$) and we deal with a difficult learning task (*e.g.* $r = 0$), then both Tikhonov and IT estimator with t steps will have a learning rate of $n^{-1/2}$. On the other hand, on an easy learning task (*e.g.* $r = 9.5$), Tikhonov estimator will saturate with a learning rate of $n^{-2/3}$ whereas IT with 10 steps will have a learning rate of $n^{-20/21}$!

Numerical experiments. In our paper, we design a synthetic classification task with known source and capacity condition, which enables us to test our upper bound numerically. The learning rate we measure empirically match remarkably well the theoretical rates we expect (table 1 in section 4).

Inexact solvers. Computing the IT estimator with t steps involves solving t successive optimization problems. Optimization can never be carried without error on a computer, so it could be argued that the error accumulates and result in a useless estimator. Hopefully, we cover this case with proposition 2 and give condition to ensure that the computational error is of the same order as the statistical error.

3.2 Modeling the influence of the learning rate on generalization in neural nets

This work is covered extensively in section 5 of part II.

The optimization of the empirical risk F in logistic regression, though differing in approach due to various regularization methods, always converges to a definable minimum. This is a consequence of using a linear predictor (e.g., $\theta(x) = \omega \cdot \varphi(x)$ in kernel methods) and a convex loss function, resulting in a convex optimization problem with a well-defined minimum. However, the introduction of non-linearities in neural networks leads to non-convexity in the objective function, complicating the analysis of the optimization path.

This work is motivated by the following empirical observation: neural networks are usually trained by minimizing the empirical risk F with gradient descent. Here, the step-size η and early-stopping criteria α are key hyperparameters. Specifically, we perform

$$\theta_0 \in \mathbb{R}^p, \;\; \theta_{t+1} = \theta_t - \eta \nabla F(\theta_t) \;\; \text{until} \;\; F(\theta_T) \leqslant \alpha. \tag{5}$$

For a stopping criteria α and initialization θ_0, a wide range of learning rates η can achieve the desired optimization threshold. Even though indistinguishable in term of optimization, as they all satisfy $F(\theta_{T_\eta, \eta}) \approx \alpha$, only estimators trained with larger learning rates consistently yield low generalization error R, hence the common wisdom to use "the biggest learning rate before the algorithm diverges" [6, 7, 9].

In [3], we hypothesize that this phenomenon is due to a discrepancy between the loss we optimize F (*e.g.* the empirical risk) and the desired minimization target R (*e.g.* the generalization error). To support our claim, we consider a convex model in a Hilbert space $\mathcal{H}$ where both F and R are quadratic functions:

$$\forall \theta \in \mathcal{H}, \;\; F(\theta) = \frac{1}{2} \|\theta - \theta^\star\|_T^2 + \text{cst}, \;\; \text{and} \;\; R(\theta) = \frac{1}{2} \|\theta - \nu^\star\|_U^2 . \tag{6}$$

Here, $\theta^\star$ and $\nu^\star$ are the minimizer of F and R respectively, and T and U are p.d. operators in $\mathcal{H}$.

3.2.1 Relevance of the model in eq. (6)

One can check that performing kernel ridge regression amounts to solving a quadratic objective in $\mathcal{H}$, and thus is encompassed by the definition of F in eq. (6). The operator T is then the regularized empirical covariance operator. In scenarios aiming to minimize excess risk under a standard Gaussian noise model (i.e., $y = \nu_\star \cdot \varphi(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma \mathbf{I})$), the risk function R

is also expressed as a quadratic form. The operator U in this context is the covariance operator $\mathbb{E}[\varphi(x) \otimes \varphi(x)]$.

The model becomes particularly relevant when the objective is to minimize classification error, as discussed in example 2 of section 5. The aim here is to minimize $\mathbb{E}[\mathbf{1}_{y \neq \mathrm{Sign},\theta(x)}]$. Building on the work of [13], provided that *(i)* the low-noise condition holds (Assumption A1 in [13]) and *(ii)* that the classes are well separated (Assumption A4 in [13]), then any estimator sufficiently close to $\nu^\star$ in Hilbert norm will inherently achieve optimal classification error (Lemma 1 in [13]). Thus, we are in a typical case where the loss we minimize F is a quadratic form with p.d. operator the covariance matrix, but the ideal loss to minimize is the Hilbert norm $R(\theta) = 1/2 \, \|\theta - \nu^\star\|^2_{\mathcal{H}}$.

3.2.2 Intuition

The essence of our result is captured in a two-dimensional toy problem, reproduced here in fig. 1. The key observation is that although the minimum of F is unique, its α level-sets are not. Moreover, taking big step size induce a bias toward optimizing the smallest eigenvectors of T first. Upon reaching the α level-set, the resulting estimator aligns more closely with the largest eigenvector of T. This result in an estimator closer to $\theta^\star$ in Hilbert norm, hence closer to $\nu^\star$ in Hilbert norm, which results in smaller classification error.

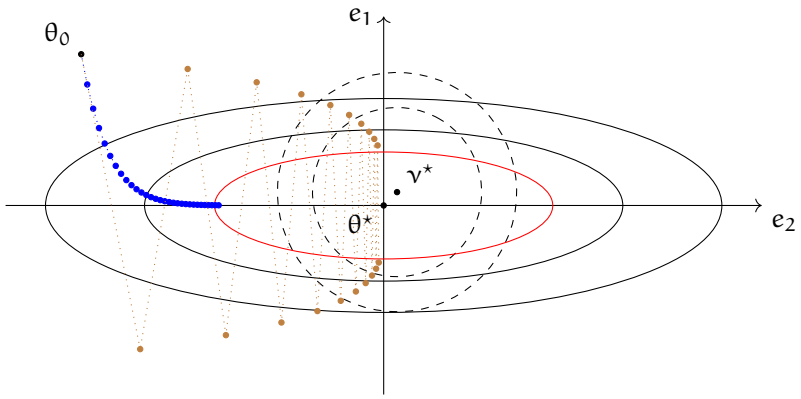


Figure 1: We optimize the quadratic F *(level sets are filled lines, centered in* $\theta^\star$*)* with gradient descent, starting from θ_0 until we reach the level sets α *(filled line, red)*. However, we evaluate the quality of the estimator through R *(level sets are dashed lines, centered in* $\nu^\star$*)*. Doing small step size *(blue dots)* optimizes the direction e_1 first, and yields an estimate which is far from $\nu^\star$ in U norm; doing big step size *(brown dots)* oscillates in the direction e_1, but ultimately yields an estimator which is close to $\nu^\star$ in U norm.

3.2.3 Contributions

Our main result formalizes this intuition.

Theorem (Informal version of theorem 2 in section 5)**.** *Under assumptions on (1) the operators* T *and* U*, (2) the learning rate, (3) the initialization and (4) the target training loss* α*. Perform gradient descent, with either small learning rate* η_s *or big learning rate* η_b*, and stop as soon as* $F(\theta_t) \leqslant \alpha$*. Denote* θ_s *and* θ_b *the resulting estimator. Then*

$$R(\theta_b) - R(\nu^\star) \;\leqslant\; 34 \frac{\kappa_U}{\kappa_T} (R(\theta_s) - R(\nu^\star)).$$

Interpretation of the result. The result bounds the excess risk of the estimator obtained with large learning rate with the one of the estimator obtained with small learning rate. An informal read is "large learning rates provide better generalization as soon as the optimization objective F is ill-conditioned". We can add that the worse the conditioning, the better the improvement.

Most importantly, in the case of classification with a Hilbert norm surrogates, $\kappa_U = 1$. This explains why in our experiments the benefits of large learning rate is especially strong with classification.

Assumptions. Assumptions (1-3) are loose and can be easily satisfied. On the other hand, assumption (4) is more restrictive. It is necessary as the optimum of F is unique: taking $\alpha \to 0$ results in $\theta_s, \theta_b \to \theta_\star$ and $R(\theta_s), R(\theta_b) \to R(\theta_\star)$, the risk of the unregularized estimator.

Numerical experiments. Experiments conducted on MNIST with the Gaussian kernel validate our main theorem (fig. 3 in section 5): taking large step size results in an estimator mostly projected on the first eigenvector of the kernel matrix, which results in lower Hilbert norm, hence better classification accuracy. Furthermore, we check that taking $\alpha \to 0$ annihilates the benefits of large learning rate, as both estimators tend to the unregularized minimizer of F. Finally, we check that a more ill-conditioned loss (which can be obtained by increasing the scale of the Gaussian kernel) results indeed in bigger improvement for large step sizes (fig. 4 in section 5).

3.3 GloptiNets: an algorithm for non-convex optimization with certificate

This work is covered extensively in section 9 of part III.

This contribution shifts the focus from statistical problems to the optimization of a non-convex function f, aiming to create an algorithm that provides a certificate of optimality for the minimum of f.

3.3.1 Motivation

Convex optimization enjoys widespread use in industry primarily because of the assurance it provides regarding the success of an algorithm. This assurance comes in the form of certificates of optimality, which are a result of duality theory. These certificates enable a clear understanding of when an algorithm has successfully achieved its objective. We aim to establish a method that offers similar certainties for the minima of non-convex functions f.

Alternatives. In those settings, the main alternative are polynomial hierarchies, introduced by Lasserre in the 2000s (a thorough introduction is given in part III) [8]. It consists in expressing the polynomial f as $c + g$, where $g = \sum_i g_i^2$ is a sum of squares. Maximizing w.r.t. c provides a lower bound on f. The key observation is that checking whether a polynomial is a sum-of-square of degree at most $2r$ can be verified by solving a semidefinite program. Such program is costly to solve, and recent works focused on leveraging algebraic properties of f, such as sparsity of the coefficient, to optimize polynomial of higher degree in higher dimension. Yet, the exponential complexity in the dimension and in the degree remains for other polynomials which lacks such sparsity.

Another line of work related to ours is [16], which writes f as $c + g$ with $g = \sum g_i^2$ is a *kernel* Sum-of-Squares. If f and the g_i are periodic functions, truncating their Fourier spectrum yields a convex semidefinite program. Unfortunately, its complexity scales exponentially with the dimension as all the frequencies $\{\omega \in \mathbb{Z}^d; \|\omega\|_\infty \leqslant N\}$ must be computed.

Kernel SoS. This work was motivated in part by the advances in kernel Sum-of-Squares (k-SoS), for which a thorough overview is given in part III. Introduced in [11], they provide an elegant way to model positive functions. In a nutshell, it replaces the linear operation of kernel methods $\theta(x) = \theta \cdot \varphi(x)$ ($\varphi(x)$ is the RKHS embedding) with a linear operation in operator space, *i.e.* $g(x) = \varphi(x) \cdot A\varphi(x)$, and $A \succeq 0$ an operator of the RKHS. Given their good theoretical properties – they benefit from a representer theorem, they are universal approximator and linear in their parameters – they have been introduced in Optimal Transport [12], Optimal Control [1], Density Modeling [14] and Global Optimization [15].

3.3.2 Contributions

We provide an algorithm, *GloptiNets*, designed to certify potential global optima of either periodic functions f or smooth functions h defined on the hypercube. For trigonometric polynomials, the algorithm address problems with a high or even infinite number of coefficients,

which are typically beyond the scope of polynomial hierarchy methods. For smooth function, our algorithm is the only solution we are aware of. Typically, it can globally optimize kernel mixtures $h(x) = \sum_i \alpha_i K(z_i, x)$, a form frequently encountered across various machine learning applications. The following provides a succinct overview of how the algorithm operates.

Recipe for certificates. We consider a periodic function f on the torus $\mathbb{T}^d$ – the same discussion applies to smooth function on the hypercube. The minimum $f_\star$ of f is the solution of this convex problem with dense constraints,

$$f_\star = \sup_c \{c \ \text{ s.t. } \ f \geqslant c\} = \sup_{c, g \geqslant 0} \{c \ \text{ s.t. } \ f - c = g\}. \tag{7}$$

Following [16, Lemma 1], this can be rewritten in a penalized form with the L_∞ norm. In passing, this highlights the relation between global optimization and uniform approximation (both problems are equivalent),

$$f_\star = \sup_{c, g \geqslant 0} c - \|f - c - g\|_{L_\infty(\mathbb{T}^d)}. \tag{8}$$

We now perform two modifications on eq. (8): *(i)* we replace the L_∞ norm with a tractable surrogate norm, and *(ii)* the search for positive function is restricted to a subspace $\mathcal{G}$.

For *(i)*, we follow [16] and use the ℓ_1 norm of the Fourier coefficients, dubbed the F-norm, which is the tightest upper bound for a set of norms (see Lemma 3 in [16])

$$\|u\|_F = \sum_{\omega \in \mathbb{Z}^d} |\widehat{u}_\omega| \geqslant \|u\|_\infty. \tag{9}$$

For *(ii)*, we introduce **extended k-SoS models**, defined with

$$\forall x \in \mathbb{T}^d, \ \ g(x) = \|RK_{\mathbf{z}}(x)\|_2^2, \ \ K_{\mathbf{z}}(x) = (K(\mathbf{z}_1, x), \ldots, K(\mathbf{z}_m, x)) \in \mathbb{R}^m. \tag{10}$$

Compared with k-SoS models of [11], *(a)* we remove the positivity constraint of the operator as $R^\top R \succeq 0$, *(b)* the resulting operator is low-rank by design, and *(c)* we gain additional expressivity by learning the anchor points $\mathbf{z}$. This formulation is at the cost of losing the convexity of the model.

Our first observation is that by combining eq. (9) and eq. (10) into eq. (8), *any* extended k-SoS model g and candidate $\widehat{x}$ provides a certificate on f.

Theorem (Theorem 2 in section 9)**.** *Given a point $\widehat{x} \in \mathbb{T}^d$ and a non-negative and periodic function $g_0 : \mathbb{T}^d \to \mathbb{R}_+$, we have*

$$f(x_\star) \geqslant f(\widehat{x}) - \|f - f(\widehat{x}) - g_0\|_F \tag{11}$$

Probabilistic estimate. To evaluate the F-norm, an infinite sum, we adopt a probabilistic estimation approach. This involves applying importance sampling using a suitably chosen probability distribution $\widehat{\lambda}$ over $\mathbb{Z}^d$. For u a periodic function with Fourier coefficients $(\widehat{u}_\omega)_{\omega \in \mathbb{Z}^d}$, we have

$$\|u\|_F = \sum_{\omega \in \mathbb{Z}^d} |\widehat{u}_\omega| = \sum_{\omega \in \mathbb{Z}^d} \frac{|\widehat{u}_\omega|}{\widehat{\lambda}_\omega} \cdot \widehat{\lambda}_\omega = \mathbb{E}_{\omega \sim \widehat{\lambda}_\omega} \left[\frac{|\widehat{u}_\omega|}{\widehat{\lambda}_\omega} \right]. \tag{12}$$

Thus, $|\widehat{u}_\omega| / \widehat{\lambda}_\omega$ is an unbiased estimate of the F-norm. Next, we assess the estimator's variance, which, fortunately, equates to an RKHS norm, as

$$\mathsf{Var} \left[\frac{|\widehat{u}_\omega|}{\widehat{\lambda}_\omega} \right] \leqslant \mathbb{E}_{\omega \sim \widehat{\lambda}_\omega} \left[\left(\frac{|\widehat{u}_\omega|}{\widehat{\lambda}_\omega} \right)^2 \right] = \sum_{\omega \in \mathbb{Z}^d} \frac{|\widehat{u}_\omega|^2}{\widehat{\lambda}_\omega} = \|u\|_{\mathcal{H}_\lambda}^2, \tag{13}$$

with $\mathcal{H}_\lambda$ the RKHS induced by the reproducing kernel

$$\forall \mathbf{x}, \mathbf{z} \in \mathbb{T}^d, \ \ K(\mathbf{x}, \mathbf{z}) = \sum_{\omega \in \mathbb{Z}^d} \widehat{\lambda}_\omega e^{2\pi i \omega \cdot (\mathbf{x} - \mathbf{z})}.$$

Combining the probabilistic estimate of eq. (12) and the bound on the variance in eq. (13) with *e.g.* the Chebychev bound, we obtain a probabilistic estimator on the F-norm which, when combined with eq. (11), gives a certificate on $f_\star$.

Theorem (Theorem 3 in section 9). *Let $(\widehat{\lambda}_\omega)_\omega$ be a probability distribution on $\mathbb{Z}^d$, $\delta \in (0,1)$ and g a positive function. Let $N > 0$ and $\widehat{S}$ be the empirical mean of $|\widehat{f}_\omega - c - \widehat{g}_\omega|/\widehat{\lambda}_\omega$ obtained with N samples $\omega_i \sim \widehat{\lambda}_\omega$. Then, a certificate with probability $1-\delta$ is given with*

$$f_\star \geqslant c - \widehat{S} - \frac{\|f - c - g\|_{\mathcal{H}_\lambda}}{\sqrt{N\delta}}, \quad \widehat{S} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|\widehat{f}_{\omega_i} - c\mathbf{1}_{\omega_i=0} - \widehat{g}_{\omega_i}\right|}{\widehat{\lambda}_{\omega_i}}. \tag{14}$$

The key hindsight is that eq. (14) holds *no matter the choice of* g. This flexibility allows for optimizing the lower bound by any means we wish, with the positive model g arbitrarily over-parametrized. This strategy aims to capitalize on the strengths of neural networks in optimizing non-convex objectives, thereby providing a reliable certificate for the non-convex function f.

GloptiNets in practice. In practice, given a candidate c for $f_\star$, we first interpolate $f - c$ with g with stochastic gradient descent. This step, unlike the certificate computation, is amenable to automatic differentiation. Certificates are computed at regular interval and the interpolation proceeds until the certificates are sufficiently tight. The main hindsight from the numerical simulations are

- By using Chebychev polynomials instead of complex exponentials as a basis, GloptiNets can effectively certify mixtures of kernels defined on the hypercube.
- When optimizing trigonometric polynomials, GloptiNets does not match the performances of Lasserre hierarchies when the polynomial has algebraic structure. However, the running time of GloptiNets does not depend on the dimension nor the number of coefficients of the polynomials, making it the only alternative for certifying polynomials under these conditions. To our knowledge, there are no alternative to GloptiNets in that case.
- Intuitively, while not sensible to the representation of $f - c$, GloptiNets is sensible to the RKHS norm of $f - c$. No matter the number of coefficients in $f - c$, GloptiNets will interpolate well function for which $\|f - c\|_{\mathcal{H}}$ is small, and will struggle to offer tight certificates for those with high norm.
- A crucial insight is that **the more parameters** the positive model g contains, the more accurately it can fit the function $f - c$. This accuracy, in turn, results in a smaller F-norm as per eq. (14), leading to **tighter certificates**.

References for the Introduction

[1] Eloïse Berthier, Justin Carpentier, Alessandro Rudi, and Francis Bach. Infinite-Dimensional Sums-of-Squares for Optimal Control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 577–582, December 2022. doi: 10.1109/CDC51059.2022.9992396.

[2] Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. Beyond Tikhonov: Faster Learning with Self-Concordant Losses via Iterative Regularization. In *Advances in Neural Information Processing Systems*, October 2021.

[3] Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. On the Benefits of Large Learning Rates for Kernel Methods. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 254–282. PMLR, June 2022.

[4] Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. GloptiNets: Scalable Non-Convex Optimization with Certificates. In *Advances in Neural Information Processing Systems*, November 2023.

[5] G. Blanchard and N. Mücke. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18, 2016. doi: 10.1007/s10208-017-9359-7.

[6] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[7] Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic Fisher explosion: Early phase Fisher matrix impacts generalization. In *International Conference on Machine Learning (ICML)*, 2021.

[8] Jean B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, January 2001. doi: 10.1137/S1052623400366802.

[9] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[10] U. Marteau-Ferey, D. Ostrovskii, F. Bach, and A. Rudi. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. In *Conference on Learning Theory (COLT)*, 2019. URL `http://proceedings.mlr.press/v99/marteau-ferey19a.html`.

[11] Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Non-parametric Models for Non-negative Functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12816–12826. Curran Associates, Inc., 2020.

[12] Boris Muzellec, Adrien Vacher, Francis Bach, François-Xavier Vialard, and Alessandro Rudi. Near-optimal estimation of smooth transport maps with kernel sums-of-squares. *arXiv:2112.01907*, December 2021.

[13] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Exponential convergence of testing error for stochastic gradient methods. In *Conference On Learning Theory (COLT)*, 2018.

[14] Alessandro Rudi and Carlo Ciliberto. PSD Representations for Effective Probability Models. In *Advances in Neural Information Processing Systems*, volume 34, pages 19411–19422. Curran Associates, Inc., 2021.

[15] Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding Global Minima via Kernel Approximations. *arXiv:2012.11978*, December 2020.

[16] Blake Woodworth, Francis Bach, and Alessandro Rudi. Non-Convex Optimization with Certificates and Fast Rates Through Kernel Sums of Squares. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 4620–4642. PMLR, June 2022.

Part I
Extending Spectral Filters Theory to More General Loss Functions

The theory of spectral filtering is a remarkable tool to understand the statistical properties of learning with kernels. For least squares, it allows to derive various regularization schemes that yield faster convergence rates of the excess risk than with Tikhonov regularization. This is typically achieved by leveraging classical assumptions called source and capacity conditions, which characterize the difficulty of the learning task. In order to understand estimators derived from other loss functions, Marteau-Ferey et al. [21] have extended the theory of Tikhonov regularization to generalized self concordant loss functions (GSC), which contain, *e.g.*, the logistic loss. In section 4, we go a step further and show that fast and optimal rates can be achieved for GSC by using the iterated Tikhonov regularization scheme, which is intrinsically related to the proximal point method in optimization, and overcomes the limitation of the classical Tikhonov regularization.

Section 4 in this part is based on our first article [4],

> Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. Beyond Tikhonov: Faster Learning with Self-Concordant Losses via Iterative Regularization. In *Advances in Neural Information Processing Systems*, October 2021.

4 Beyond Tikhonov: Faster Learning with Self-Concordant Losses via Iterative Regularization

4.1 Introduction

We consider the problem of supervised learning where we want to find a prediction function θ mapping an input point x living in a set $\mathcal{X}$ to a label y in $\mathcal{Y}$. In this section, we assume that θ lives in a separable Hilbert space $\mathcal{H}$ and is learned from a set of observations $(x_i, y_i)_{i=1,\dots,n}$ that are i.i.d. samples drawn from an unknown probability distribution ρ on $\mathcal{X} \times \mathcal{Y}$. The goal is to find θ that minimizes the expected risk L, which is defined below along with the empirical risk $\widehat{L}$:

$$L(\theta) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, \theta(x)) d\rho(x, y), \qquad \widehat{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta(x_i)), \tag{1}$$

where ℓ is a suitable loss function comparing true labels with predictions. This section aims for upper bounds on the excess risk for a specific estimator $\widehat{\theta}$. That is, we assume that the minimum of the expected risk is attained for some $\theta^\star$ in $\mathcal{H}$, and we want to derive *probabilistic upper bounds on the excess risk*:

$$\mathbb{P}\left[L(\widehat{\theta}) - L(\theta^\star) > C_1 n^{-\gamma} \log \tfrac{2}{\delta}\right] \leqslant \delta, \tag{2}$$

given some value δ in $(0, 1)$, where C_1 is a positive constant, and $\widehat{\theta}$ is an estimator built from the n observations. The quantity $O(n^{-\gamma})$ denotes the rate of convergence of the estimator $\widehat{\theta}$. A classical "slow" rate with $\gamma = 1/2$ is typically achieved by many estimators and is in fact optimal if only mild assumptions are made about the data distribution ρ. Even though optimal, this rate is nevertheless a worst case and faster rates with $\gamma > 1/2$ can be achieved both in theory and in practice, by making additional assumptions about the difficulty of the learning task. Originally introduced in the literature of inverse problems, the so-called *source* and *capacity* conditions have been shown to be appropriate for this purpose, leading to statistical analysis with fast rates of convergence [21, 13, 8]. The optimality of results of the form (2) is characterized by comparing them with lower bounds that are available for various sets of data distributions ρ [8]. Matching upper bounds with lower bounds ensures that the estimator $\widehat{\theta}$ is *optimal*, in the sense that no information is lost in the process of exploiting the data samples to compute $\widehat{\theta}$, for the given set of distributions.

In this search for optimal estimators, most of the attention has been devoted to minimizers of some function of the empirical risk $\widehat{L}$, which is defined in eq. (1). Then, the key challenge is to *regularize* $\widehat{L}$ in order to achieve better generalization properties. The most widely used scheme is probably Tikhonov regularization; other examples when $\mathcal{H}$ is a RKHS include truncated regression [27], or early stopping in gradient descent algorithms [31, 1]. When the loss ℓ is set to least squares, it can be shown that minimizing the excess risk amounts to solving an ill-posed inverse problem [29], which led to the remarkable theory of *spectral filtering*. A large class of regularization schemes can indeed be seen as a filtering process applied to the training labels y_i after regularizing the spectrum of the kernel matrix [13, 3]. Interestingly, this theory has highlighted the fact that not all regularization schemes are equal: some of them obtain fast learning rates in (2) on "easy" problem (a thorough definition is given in section 4.2) while others cannot leverage this additional regularity to improve the learning rate.

Such a general analysis for least squares is made possible by the fact that a closed-form expression of the estimator is available. When considering different loss function ℓ, the estimator $\widehat{\theta}$ is unfortunately only implicitly available as the solution of an optimization problem involving $\widehat{L}$. A step to extend least squares results to more general loss functions has been achieved by Marteau-Ferey et al. [21], who provide bounds on the form (2) for Tikhonov estimator on generalized self concordant (GSC) functions. GSC functions are three-times-differentiable functions whose third derivative is bounded by the second-derivative. In practice, they were introduced to conduct a general analysis of the Newton method in optimization [7, 22], and

adapted in [2] to encompass a larger class of loss function. It includes notably the logistic regression loss, which is widely used for classification.

While Tikhonov yields fast rates of convergence in several data regimes, it is known to be unable to adapt to the whole range of learning task difficulties. More precisely, it suffers from a "saturation" effect [13], meaning that when the learning task becomes simpler, the learning rate stops improving and is suboptimal. Our section addresses this limitation for GSC functions by considering instead the iterated Tikhonov regularization (IT) scheme. In the context of least squares, this approach consists of successively fitting the residuals. For more general loss functions, it is equivalent to performing a few steps of the proximal point method in optimization [23]. Our main result is a probabilistic upper bound on the excess risk, which is optimal given usual source and capacity conditions assumptions on the learning task, thus addressing the limitations of the classical Tikhonov regularization.

4.2 Background and Preliminaries

4.2.1 Definitions: Estimator and Loss Function

Let $\mathcal{X}$ be a Borel input space, $\mathcal{Y}$ be a vector-valued output spaces, and ρ a probability distribution on $\mathcal{X} \times \mathcal{Y}$. We consider $\mathcal{H}$ to be a separable Hilbert space of functions from $\mathcal{X}$ to $\mathcal{Y}$. Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we aim at minimizing the expected loss, while we only have access to the empirical loss – both are defined in eq. (1). Our work provides an upper bound on the excess risk of the iterated Tikhonov estimator. For the basic case of least squares with $\mathcal{Y} = \mathbb{R}$, it is usually defined as a procedure that refits the residuals, see, e.g., §5.4 in [13]. Starting with $\widehat{\theta}_\lambda^0 = 0$, it consists of the sequence

$$\widehat{\theta}_\lambda^t = \widehat{\theta}_\lambda^{t-1} + \operatorname*{arg\,min}_{\theta \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \left(y_i - \widehat{\theta}_\lambda^{t-1}(x_i) - \theta(x_i) \right)^2 + \frac{\lambda}{2} \|\theta\|^2 \right\}. \tag{3}$$

To extend this regularization to other loss function, we make the change of variable $\theta' = \widehat{\theta}_\lambda^{t-1} + \theta$ in the equation above, which yields the proximal point algorithm [23].

Definition 1 (Iterated Tikhonov estimator a.k.a. proximal point algorithm)**.** *We define the iterated Tikhonov estimator with the following sequence. Given $\lambda > 0$ and $\widehat{\theta}_\lambda^0 = 0$,*

$$\widehat{\theta}_\lambda^{t+1} = \mathsf{prox}_{\widehat{L}/\lambda}(\widehat{\theta}_\lambda^t) \overset{\text{def.}}{=} \operatorname*{arg\,min}_{\theta \in \mathcal{H}} \left\{ \widehat{L}(\theta) + \frac{\lambda}{2} \left\| \theta - \widehat{\theta}_\lambda^t \right\|^2 \right\}, \tag{4}$$

where $\mathsf{prox}_{\widehat{L}/\lambda}$ denotes the proximal operator of the empirical risk $\widehat{L}$ rescaled by $1/\lambda$.

Remark 1. *In practice, the proximal operator is only computed approximately by using an optimization algorithm. Nevertheless, the benefits in terms of statistical accuracy of the iterated Tikhonov scheme are robust to inexact solutions, as long as the accuracy for solving the sub-problems eq.* (4) *is high enough. We discuss this point in section 4.3.2.*

Remark 2. *It is easy to show that the sequence of the proximal point algorithm always converges to a minimizer of the unregularized empirical risk, which is of course not what we are interested in. Instead, we consider and analyze the procedure with a fixed small number of steps t and show later that optimal learning rates can be obtained by choosing an appropriate parameter λ.*

Remark 3. *When the loss is a function of a residual $y - \theta(x)$—assuming $\mathcal{Y}$ to be a vector space—as in the least square case, we recover the classical definition consisting of refitting the residual, and with $t = 1$, we recover Tikhonov.*

Interestingly, our definition makes the estimator compatible with other loss functions, such as the logistic loss. More precisely, the main assumption we make on the loss is to be *generalized self concordant*. We follow the definition of [21], which is a special case of 2-self concordance introduced in [28]:

Definition 2 (Generalized self-concordance)**.** *For any $z = x, y \in \mathcal{X} \times \mathcal{Y}$, the function $\ell_z : \mathcal{H} \to \mathbb{R}$ defined as $\ell_z(\theta) = \ell(y, \theta(x))$ is convex and three times differentiable. Besides, there exists a set $\phi(z) \subseteq \mathcal{H}$ s.t:*

$$\forall \theta, h, k \in \mathcal{H}, \quad \left| \nabla^3 \ell_z(\theta)\,[h, k, k] \right| \leqslant \sup_{g \in \phi(z)} |k \cdot g|\, \nabla^2 \ell_z(\theta)\,[k, k]\,. \tag{5}$$

The brackets indicate that the vectors h, k and k are applied to the 3-dimensional tensor $\nabla^3 \ell_z(\theta)$. The definition seems technical at first sight, but intuitively, this assumption allows to upper bound the deviation between the objective function and its local quadratic approximation. This enables a simple analysis of the Newton method for optimization, making it easy to quantify the basin of quadratic convergence [20]. On top of this, it has the benefit of encompassing a large class of loss functions, such as the logistic loss: see Example 1 in [21] for values of $\phi(z)$ with usual losses. We provide some intuition on GSC loss functions in remark 6 in section A.3.1.

In order to ensure the existence of the loss and its derivatives everywhere, we also need the following technical assumptions also introduced in [21], which are reasonable in practice. This ensures that both L and $\widehat{L}$ are generalized self concordant too.

Assumption 1 (Technical assumptions)**.** *There exists* R *s.t* $\sup_{g \in \phi(z)} \|g\| \leqslant R$ *almost surely for* z *drawn from the distribution* ρ *and* $|\ell_z(0)|, \|\nabla \ell_z(0)\|, \operatorname{Tr} \nabla^2 \ell_z(0)$ *are almost surely bounded.*

The following assumption is usual in excess risk analysis [21, 25]. In our proof strategy, all the quantities are vectors and operators in $\mathcal{H}$, which makes the analysis simpler. Weakening this assumption (*e.g.* assuming that $\theta^\star \in \mathcal{L}_2(\mathcal{X})$) would require finding an equivalent of the covariance operator for GSC loss function, which constitute an interesting future direction.

Assumption 2 (Existence of a minimizer)**.** *There exists* $\theta^\star$ *in* $\mathcal{H}$ *s.t* $L(\theta^\star) = \inf_{\theta \in \mathcal{H}} L(\theta)$.

Finally, following [21] we also define the *expected Hessian* and the *regularized expected Hessian* as

$$\forall \theta \in \mathcal{H},\ \lambda > 0, \quad \mathbf{H}(\theta) = \mathbb{E}_{z \sim \rho}\left[\nabla^2 \ell_z(\theta)\right], \quad \mathbf{H}_\lambda(\theta) = \mathbf{H}(\theta) + \lambda \mathbf{I},$$

and we introduce the degrees of freedom, also known as the effective dimension of the problem:

Definition 3 (Degrees of freedom)**.** *The degrees of freedom is defined as:*

$$\forall \lambda, \quad \mathsf{df}_\lambda = \mathbb{E}_{z \sim \rho}\left[\|\nabla \ell_z(\theta^\star)\|^2_{\mathbf{H}_\lambda^{-1}(\theta^\star)}\right].$$

where we denote by $\|\theta\|_A = \left\|A^{1/2}\theta\right\|$*, with* $\theta \in \mathcal{H}$*, the norm induced by a positive definite operator* A *on* $\mathcal{H}$*.*

Remark 4. *The intuition about this definition is not straightforward. To better understand why this quantity is a key to characterize the amount of regularization in a learning problem, it is useful to consider the specific case of the square loss with kernels. In such a case,* $\mathcal{H}$ *is a reproducing kernel Hilbert space (RKHS) and* $\theta(x) = \theta^\top \Phi(x)$*, where* $\Phi : \mathcal{X} \to \mathcal{H}$ *is the kernel mapping. Then, the Hessian is constant everywhere and equal to the covariance operator* $T = \mathbb{E}_{x \sim \rho_x}\left[\Phi(x) \otimes \Phi(x)\right]$ *where* ρ_x *is the marginal of* ρ*. Consequently, the degrees of freedom (also known as* effective dimension*) is a spectral function of* T *which may be written as* $\mathsf{df}_\lambda = \operatorname{Tr} T T_\lambda^{-1}$*. This is the classical quantity which appears on the bias/variance decomposition of the excess risk, with a variance part decaying in* $\operatorname{Tr} T T_\lambda^{-1}/n$*, see [8].*

4.2.2 Source and Capacity Conditions

We now introduce the hypotheses we make on the learning task, which will allow us to derive fast rates of convergence. They measure the difficulty of the problem and are classical in the context of learning with kernels, see *e.g.* [3, 24, 6]. It is indeed established that given an algorithm which outputs an estimator $\widehat{\theta}$, one can find a probability measure ρ s.t the learning rate of the estimator is arbitrarily low, a result known as the "no-free lunch theorem" [14]. Inspired by the literature of inverse problems, two assumptions were introduced to restrict the space of considered distributions.

Assumption 3 (Source condition)**.** *There exists* $r > 0$ *and* v *in* $\mathcal{H}$ *s. t:* $\theta^\star = \mathbf{H}^r(\theta^\star)v$.

$A \mapsto A^r$ is the usual power for positive definite operators. The source condition should be seen as a smoothness assumption on $\theta^\star$, and for least square, we recover the usual definition of the source condition, that is $\theta^\star = T^r v$, with T the covariance operator we previously defined. Bigger r implies that the optimum can be well approximated by a few eigenvectors. Assuming $r = 0$ simplifies to $\theta^\star \in \mathcal{H}$.

The second assumption characterizes the ill-posedness of the problem:

Assumption 4 (Capacity condition). *There exists* $\alpha > 1$, $s, S > 0$ *s.t* $s\lambda^{-1/\alpha} \leqslant \mathrm{df}_\lambda \leqslant S\lambda^{-1/\alpha}$.

Again, for the square loss, it turns to a bound on the eigenvalue decay of the covariance operator. If σ_j, e_j is an eigenbasis of T, then $\sigma_j = O(j^{-\alpha})$. Said differently, the bigger α, the fewer directions are needed to approximate well a sample $x \sim \rho_x$ in expectation, and the easier is the learning task. This is an assumption on the input space $\mathcal{X}$ and does not imply anything on the labels $\mathcal{Y}$.

4.2.3 Previous Results

Our main result considers iterated Tikhonov *with* GSC loss functions. While iterated Tikhonov has been previously analyzed for squared loss by leveraging the theory of spectral filtering (see below), extensions to other loss functions raise several difficulties, which will be detailed in section 4.3.

Spectral filters and least squares. As we mentioned earlier, the key insight on regularization with the square loss is that a closed-form expression of the estimator is available. By using the same notation as in remark 4, the kernel ridge regression estimator can be for instance written

$$\widehat{\theta}_\lambda = \sum_{i=1}^{n} \beta_i \Phi(x_i) \quad \text{with} \quad \beta = \frac{1}{n} g_\lambda\left(\frac{K}{n}\right) y, \tag{6}$$

where K is the $n \times n$ kernel matrix, $y = (y_i)_{1 \leqslant i \leqslant n}$ is the vector of training labels and $g_\lambda(K/n) = (K/n + \lambda I)^{-1}$. Note that g_λ is a function acting on the spectrum of K, which makes it a special case of regularization by *spectral filtering*, which may be analyzed for more general functions g_λ. In particular, a key quantity for understanding the regularization effect of a filter g_λ is the so-called *qualification*. Following [13, 3], this quantity is defined below.

Definition 4 (Qualification of a spectral filter). *For any* $\lambda > 0$, *define* $g_\lambda : [0, 1] \to \mathbb{R}$ *a filter function. Its qualification is the highest* q *such that*

$$\forall \nu \leqslant q, \quad \sup_{\sigma} |1 - \sigma g_\lambda(\sigma)|\, \sigma^\nu \leqslant \omega_\nu \lambda^\nu, \tag{7}$$

with ω_ν *a constant independent of* λ.

Under the source and capacity conditions, it is possible to show that the resulting estimator would enjoy an optimal rate in $n^{-\frac{\alpha(1+2r)}{1+\alpha(1+2r)}}$ if $r + 1/2 \leqslant q$ (where r comes from the source condition). When $r + 1/2 > q$, the rate is instead of order $n^{-\frac{\alpha(1+2q)}{1+\alpha(1+2q)}}$, which is suboptimal, see *e.g.* Thm. 3.4 [6] (set the parameter s to $1/2$). This illustrates the *saturation effect* of some regularization schemes. For example, Tikhonov regularization amounts to filtering with $g_\lambda : \sigma \mapsto (\sigma + \lambda)^{-1}$ and has *qualification* 1, so the parameter r *saturates* at $r = 1/2$. Thus, even if $r \gg 1/2$, the excess risk of $\widehat{\theta}_\lambda$ will decay in $n^{-\frac{\alpha}{1+\alpha}}$, which is suboptimal. Designing estimators with high qualification is key to obtaining fast rates that can adapt to both hard and easy learning tasks.

Iterated Tikhonov with the Square Loss. We can compute the spectral filter function g_λ^t corresponding to t iterations of IT, which yields

$$g_\lambda^t : \sigma \mapsto (\sigma + \lambda)^{-1} \sum_{i=0}^{t-1} \left(\frac{\lambda}{\sigma + \lambda}\right)^i = \sigma^{-1}\left(1 - \left(\frac{\lambda}{\sigma + \lambda}\right)^t\right). \tag{8}$$

Choosing a fixed t and computing the supremum of $\sigma \mapsto |1 - \sigma g_\lambda(\sigma)|\, \sigma^\nu$, we find that IT estimator has qualification t, which is thus better than Tikhonov. IT has been thoroughly studied in the community of inverse problems, dating back to the work of [15]. It was naturally transferred to learning with kernels thanks to the aforementioned connection with inverse problems.

The link we make with the proximal point algorithm has never been studied from a statistical perspective, to the best of our knowledge, even though it has attracted a lot of attention in the

optimization literature, notably with accelerated algorithms [17, 16], or variants of the proximal operator on a class of self-concordant loss functions [9]. More attention was devoted to *boosting*, where the penalty λ is fixed but the number of iterations t may go to infinity, necessitating an appropriate stopping rule [18]. Nevertheless, such a work focuses on the least square loss, where the theory of spectral filter can be applied. Finally, the proximal sequence in eq. (4) can be cast as a constrained optimization problem related to sequential greedy approximation [32].

Tikhonov and Generalized Self Concordant losses. Extending the results obtained with the square loss to more general losses is challenging since there is no closed form available for the resulting estimator, and the theory of spectral filtering does not apply. Nevertheless, the case of Tikhonov regularization for GSC loss functions was treated in [21]. It is shown that the resulting estimator enjoys optimal rate as long as $r \leqslant 1/2$, meaning that the saturation of Tikhonov regularization is recovered in those settings. We will extend these results to the IT regularization, showing that an improved qualification can be achieved, leading to fast rates for a larger class of learning tasks.

4.3 Main Result

Our main result establishes an optimal non-asymptotic bias variance decomposition of the excess risk. It is optimal in the sense that choosing an appropriate regularization parameter λ enables to achieve the optimal lower rates of convergence established for least squares.

Theorem 1 (Optimal rates of IT estimator)**.** *Let $\delta \in (0, 1]$, and set $\lambda \in (0, L_0)$, $n \geqslant N$. The following bound on the excess risk holds with probability greater than $1 - 2\delta$:*

$$L(\widehat{\theta}^t_\lambda) - L(\theta^\star) \leqslant C_{\text{bias}} \lambda^{2s} + C_{\text{var}} \frac{\mathrm{df}_\lambda}{n}, \text{ with } s = \min\{r + 1/2, t\}. \tag{9}$$

If we further assume that the capacity condition holds and that the estimator does not saturate, that is $t \geqslant r + 1/2$, then setting

$$\lambda = C_{\text{risk}}\, n^{-\frac{\alpha}{1+\alpha(2r+1)}}, \tag{10}$$

makes the following holds with probability greater than $1 - 2\delta$:

$$L(\widehat{\theta}^t_\lambda) - L(\theta^\star) \leqslant 2C_{\text{risk}}\, n^{-\frac{\alpha(2r+1)}{1+\alpha(2r+1)}}. \tag{11}$$

The constants $L_0, N, C_{\text{bias}}, C_{\text{var}}, C_{\text{risk}}$ are detailed in theorem 4 in the appendix; they are explicit and depend only on $r, \alpha, S, R, t, \delta$ and the distribution ρ.

Optimal rates. First, we note that the decay rate of the excess risk is optimal provided $t \geqslant r + 1/2$. It means that, up to constant factors, no estimators trained on n observations can benefit from a better learning rate (in the worse case sense) with the prior considered on ρ, that is source and capacity conditions of parameters r, α. This leads to the second point: we see that IT has qualification $q = t$. When $t = 1$, this is Tikhonov estimator and we recover the result of [21]. This qualification shows in the bound on the bias: if $r \leqslant t - 1/2$, the bias is optimal in λ^{2r+1}; otherwise, it is suboptimal and decays only in λ^{2t}, which leads to higher excess risk, hence generalization error.

Influence of t. The leading multiplicative constant of the rate C_{var} in eq. (9) depends linearly on the number of steps t, as shown in eq. (56) in section A.2.5. Thus, the rate in eq. (11) is optimal in n when $t = O(r)$. Letting t go to infinity amounts to minimizing the empirical risk, which yield the unregularized estimator: this agrees with our bound on the excess risk, as the constant C_{var} would go to infinity in that case.

Source and capacity condition. The source and capacity conditions enable precise bounds on the bias and the variance, respectively. If they do not hold, the bias can only be bounded by $O(\lambda)$, while we can upper bound the degrees of freedom with $O(1/\lambda)$, leading to slow learning rates. If the source condition holds but the capacity condition does not, we then obtain learning rates in $n^{-2s/(2s+1)}$, $s = \min\{r + 1/2, t\}$, which are also optimal in these settings.

Example: a very easy learning task. Suppose the source condition satisfies $r = 10$ and that the capacity condition does not hold. Then, using Tikhonov estimator [21] amounts to setting $t = 1$. The generalization error would then decay as $n^{-2/3}$. On the other hand, using Iterated Tikhonov estimator with $t = 10$ would make the generalization error decay in $n^{-20/21}$, which is much better.

4.3.1 Sketch of the proof

The proof, which is fully detailed in the appendix, has the following outline:

- First, we give technical results on generalized self concordant functions;
- Then, we define the intermediate quantity in our bias-variance decomposition;
- Finally, we proceed to bounding the bias and the variance separately, which plugged together give our bound on the excess risk.

To prove the theorem above we build upon the tools from [21] on generalized self concordant functions. The resulting proof covers and simplifies the case of Tikhonov regularization (one step of iterated Tikhonov) and generalizes the rates to $r > 1/2$. We provide also a fine control of the constants, that takes into account the sequential nature of the IT estimator.

Properties of generalized self concordant loss functions Here, we report key properties of GSC loss functions, which are covered in depth in section A.1. GSC loss functions are convenient to study as they come with a set of bounds on the Hessian, the gradients and the function values. Intuitively, by integrating multiple times the relation between the third and second derivative in the definition from eq. (5), one can obtain bounds on function values. To introduce them, we first define the following function:

$$\forall \theta \in \mathcal{H}, \quad t(\theta) = \sup_{z \in \operatorname{Supp} \rho} \sup_{g \in \phi(z)} |g \cdot \theta| \,. \tag{12}$$

By integrating three times the bound of the definition, one can show that:

$$L(\widehat{\theta}_\lambda^t) - L(\theta^\star) \leqslant \Psi\left(t(\widehat{\theta}_\lambda^t - \theta^\star)\right) \left\|\widehat{\theta}_\lambda^t - \theta^\star\right\|^2_{\mathbf{H}(\theta^\star)}, \quad \Psi : t \mapsto (e^t - t - 1)/t^2. \tag{13}$$

This type of bound first appeared in [2] and was given in this form in [21]. We report it in proposition 3 in the appendix. For instance, when ℓ is the square loss, $t = 0$ everywhere and the r.h.s turns to $1/2\|\widehat{\theta}_\lambda^t - \theta^\star\|^2_{\widehat{T}}$, see [6, 11]. On top of this, we generalize a lower bound on the gradient:

Lemma 1 (Stacking operator on gradient bounds)**.** *Let $\theta, \nu, \xi \in \mathcal{H}, \lambda > 0$. If $A : \mathcal{H} \to \mathcal{H}$ commutes with $\mathbf{H}(\xi)$, the following holds:*

$$e^{-t(\theta - \xi)} \underline{\phi}\left(t(\nu - \theta)\right) \|A(\nu - \theta)\|_{\mathbf{H}_\lambda(\xi)} \leqslant \|A(\nabla L_\lambda(\nu) - \nabla L_\lambda(\theta))\|_{\mathbf{H}_\lambda^{-1}(\xi)}, \tag{14}$$

where $\underline{\phi} : t \mapsto (1 - e^{-t})/t$.

Together with eq. (13), this result is the workhorse of our proof for the upper bound on the excess risk. It is detailed and proven in section A.4.

Bias-variance decomposition. Thanks to eq. (13), we can relate the excess risk with the distance between estimates. This is why bounding the excess risk amounts to finding a good bias-variance decomposition. Most of the proof we find for the square loss rely on the quantity

$$\vartheta_\lambda^t = g_\lambda^t(\widehat{T})\widehat{T}\theta^\star, \tag{15}$$

with $\widehat{T} = 1/n \sum_i \Phi(x_i) \otimes \Phi(x_i)$ the empirical covariance operator, obtained by replacing ρ with the empirical distribution in remark 4. This is basically the estimator trained on *noiseless empirical data* (*i.e.* using $\theta^\star(x_i)$ instead of y_i) [6, 10, 18]. Unfortunately, working with GSC function makes the spectral filtering point of view inapplicable. We need to translate a closed-form expression

of the intermediate quantity with filters into the solution of an optimization problem. In our case, we can achieve the optimal bias-variance decomposition with the following quantity:

$$\begin{aligned} \vartheta_\lambda^0 &= \theta^\star, \\ \vartheta_\lambda^{k+1} &= \mathsf{prox}_{\widehat{L}/\lambda}(\vartheta_\lambda^k), \quad k \geqslant 0. \end{aligned} \tag{16}$$

Consequently, we write

$$\|\widehat{\theta}_\lambda^t - \theta^\star\|_{\mathbf{H}(\theta^\star)} \leqslant \|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\|_{\mathbf{H}(\theta^\star)} + \|\vartheta_\lambda^t - \theta^\star\|_{\mathbf{H}(\theta^\star)}. \tag{17}$$

We recover eq. (15) with the square loss. In [21], a different decomposition is used; we found eq. (16) to greatly simplify the proof.

Bounding the bias and the variance. The first term in eq. (17) is the *bias* of the estimator, as it goes to 0 when the regularization λ goes to 0. By applying the lower bound on gradient values – eq. (14) – with the definition of the proximal operator, one can express $\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\|$ function of $\|\widehat{\theta}_\lambda^{t-1} - \vartheta_\lambda^{t-1}\|$. Unfolding the recursion, we obtain theorem 2 in the appendix. It shows that the bias decreases in $O\left(\lambda^{r+1/2}\right)$ if the qualification is sufficient, *i.e.* $t \geqslant r + 1/2$. Otherwise, we recover the saturation experienced with least squares: the bias only decreases in $O\left(\lambda^t\right)$. Specific attention is devoted to bounding the prefactor, which is otherwise difficult to manage.

The second term in eq. (17) is the *variance*, as it goes to 0 when the number of samples n increases. theorem 3 shows that it decays in $O(\sqrt{\mathsf{df}_\lambda/n})$. It follows closely the work of [6]. However, we cannot use the convenient fact that $\mathsf{df}_\lambda = \mathrm{Tr}\ \mathbf{H}(\theta^\star)\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)$, which is valid for least squares but not in general. Thus, we took specific care in adapting our bounds to the different regimes so as not to impact the learning rate.

Plugging these results together, we obtain the upper bound on the excess risk.

4.3.2 Optimization

The aim of this section is to extend the result of theorem 1 to a practical case, where we only have access to an inexact solver for computing the proximal operator. Specifically, let $\epsilon > 0$ be the error (to be defined precisely in proposition 1) made when approximating $\widehat{\theta}_\lambda^t$ with $\overline{\theta}_\lambda^t$, the quantity we compute numerically. We aim for a bound of the type:

$$L(\overline{\theta}_\lambda^t) - L(\theta^\star) \leqslant C_{\text{risk}}\, n^{-\frac{\alpha(2r+1)}{1+\alpha(2r+1)}} + \epsilon.$$

The first term in the right hand side is the *statistical error*, and is optimal following the discussion of theorem 1. The second term is the *optimization error*, which is the price to pay for approximating $\widehat{\theta}_\lambda^k$ by $\overline{\theta}_\lambda^k$ with tolerance ϵ. The goal is to give a simple optimization rule on the sub-problems to ensure that ε is of the same order as the upper-bound for the noiseless case.

Assuming that we cannot compute the proximal operator in eq. (4) exactly, we need to evaluate how the error in approximating $\widehat{\theta}_\lambda^1$ propagates to the evaluation of $\widehat{\theta}_\lambda^2$, and so on. As generalized self-concordant functions are well suited to (approximate) second-order optimization scheme, we assume we use a solver with guarantees on a quantity called *Newton decrement*, such as the one developed in [20]. Starting from $\widehat{\theta}_\lambda^0 = \overline{\theta}_\lambda^0 = 0$, define the following for $k > 0$:

$$\widehat{\theta}_\lambda^k = \arg\min_{\theta \in \mathcal{H}} \widehat{L}_\lambda^{k-1}(\theta) \overset{\text{def.}}{=} \widehat{L}(\theta) + \frac{\lambda}{2}\left\|\theta - \widehat{\theta}_\lambda^{k-1}\right\|^2, \qquad \nu_\lambda^k(\theta) \overset{\text{def.}}{=} \left\|\nabla \widehat{L}_\lambda^{k-1}(\theta)\right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta)}, \tag{18}$$

$$\overline{\theta}_\lambda^k \approx \arg\min_{\theta \in \mathcal{H}} \overline{L}_\lambda^{k-1}(\theta) \overset{\text{def.}}{=} \widehat{L}(\theta) + \frac{\lambda}{2}\left\|\theta - \overline{\theta}_\lambda k - 1\right\|^2, \quad \overline{\nu}_\lambda^k(\theta) \overset{\text{def.}}{=} \left\|\nabla \overline{L}_\lambda^{k-1}(\theta)\right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta)}. \tag{19}$$

$\overline{\theta}_\lambda^k$ approximates the proximal operator evaluated on $\overline{\theta}_\lambda k - 1$, and $\overline{L}_\lambda^{t-1}$ is the function we manipulate at step t. If the optimization was carried without error in eq. (19), we would have $\overline{L}^{t-1} = \widehat{L}^{t-1}$. The quality of the approximation is measured with the Newton decrement of eq. (19), see *e.g*, Lemma 6 of [20]. We need to enforce a bound on the true Newton decrement in eq. (18) when we only have access to $\overline{L}_\lambda^{t-1}$. The next proposition gives a simple rule to achieve this.

Proposition 1 (Error propagation with proximal sequence). *Let $\epsilon > 0$ the target precision. Assume that we can solve each sub-problem with precision $\overline{\epsilon}_k$:*

$$\forall k \in \{1, \dots, t\}, \quad \overline{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) \leqslant \overline{\epsilon}_k = \epsilon \frac{1.4^{k-t}}{t},$$

and that $\epsilon \leqslant \sqrt{\lambda}/(2R)$. This suffice to achieve an error ϵ on the target function:

$$\nu_\lambda^{t-1}\left(\overline{\theta}_\lambda^t\right) \leqslant \epsilon.$$

This is a specialized version of proposition 7, whose proof is detailed in the appendix. Intuitively, this means that enforcing a geometrically higher precision on the first steps is sufficient to obtain high precision on the final estimate. To compute IT's estimator in practice, one would need to solve t optimization problem with decreasing precision. As second order schemes have double logarithmic complexity w.r.t the precision ϵ, the complexity of computing the proximal sequence of IT with tolerance ϵ would be only (up to logarithm term) t times bigger than estimating Tikhonov estimator with tolerance ϵ. In practice, when learning with kernels, one would use the representer theorem and aim at estimating β in $\mathbb{R}^n$ as in eq. (6) [26]. This results in an optimization problem with n observations in dimension n, with complexity $O(n^3)$. A practical implementation could use Nyström projection to avoid this cubic computational burden in the number of samples. The statistical effects of such projection are well studied with Tikhonov regularization [20, 25]; their effect on other regularization scheme is an interesting future research direction.

This proposition can be used directly to bound the excess risk with inexact solvers.

Proposition 2 (Upper bound on the excess risk with inexact solvers). *Let $\delta \in (0, 1)$ and assume that the* statistical *assumptions of theorem 1 hold as well as the* optimization *assumptions of proposition 1. Then, the following bound on the excess risk holds with probability greater than $1 - 2\delta$:*

$$L(\overline{\theta}_\lambda^t) - L(\theta^\star) \leqslant 2C_{\text{risk}}\, n^{-\frac{\alpha(2r+1)}{1+\alpha(2r+1)}} + E_{1/2}\,\epsilon, \quad s = \min\{r + 1/2, t\} \tag{20}$$

with C_{risk} as in theorem 1 and $E_{1/2} \leqslant 4.3 \cdot 10^3$.

This is a specialized version of proposition 8 proved in the appendix. The first term is the statistical excess risk, whereas the second term in ϵ is the price we pay for inexact approximation. For the sake of clarity, crude upper bounds were used (notably $\widehat{\mathbf{H}}_\lambda^{-1/2}(\cdot) \leqslant B_2^\star/\sqrt{\lambda}$) at the expanse of big constants. They can be expected to be an order of magnitude lower in practice.

Setting t in real application. In classical machine learning settings, we do not have access to the source condition parameter r. The number of proximal steps t can be seen as an hyperparameter, which is chosen by cross-validation. One would run the algorithm and test the resulting error on a validation set for each iteration, and keep doing proximal steps as long as the validation loss improves.

4.4 Experiments

The purpose of the experiments is to illustrate the saturation effect of the Tikhonov estimator when $r \gg 1/2$, and see how the saturation is overcome by iterated Tikhonov IT. We also show that the statistical rates we derive are achieved both in theory and in practice on synthetic data with well-controlled source and capacity conditions.

Settings. To that end, we use a synthetic binary classification data set for which we know the source and capacity condition parameters r and α by design. Then, we study the performance of $\text{IT}(t)$, $t \in \{1, \dots, 8\}$, trained with the logistic loss, which satisfies definition 2 about generalized self-concordant functions. Related experiments were conducted in the context of kernel ridge regression with synthetic data in [24], which we follow here. Specifically, we use splines of order α to define a kernel matrix:

$$K(x, z) = \Lambda_\alpha(x, z) = \sum_{k \in \mathbb{Z}} \frac{e^{2i\pi k(x-z)}}{|k|^\alpha},$$

for which a closed form expression is available as soon as α is a positive even integer (see for instance Eq (2.1.7) in [30]). We then use $\mathcal{X} = [0,1]$, ρ_X is the uniform distribution, and $\theta^\star(x) = \Lambda_{(r+1/2)\alpha+1/2}(0,\cdot)$, which may be shown to live in the RKHS $\mathcal{H}$ of K. Then, it is possible to show that the source and capacity assumption are satisfied with value r, α, see [24].

Finally, we design the distribution $\rho_{Y|X}$ of the labels such that $\theta^\star$ is indeed the minimizer of the risk over $\mathcal{H}$. This may be ensured if $\theta^\star$ coincides with the minimizer of the risk over the set of measurable functions, which has the following form under mild assumptions (see Eq. (3) in [10]):

$$\theta^\star(x) = \arg\min_z \mathbb{E}_{Y|X}\left[\ell(Y,z)\right]. \tag{21}$$

The previous relation can be satisfied by choosing $\rho_{Y|X}$ accordingly. More precisely, we need

$$\mathcal{Y} = \{-1,1\}, \quad \mathbb{P}(Y=1 \mid X) = \left(1+e^{-\theta^\star(x)}\right)^{-1}, \quad \mathbb{P}(Y=-1 \mid X) = \left(1+e^{\theta^\star(x)}\right)^{-1},$$

which ensures that eq. (21) holds – see details in section A.5.3. To our knowledge, this is the first synthetic dataset with given source and capacity condition for classification tasks. For each λ, t, we sample n points uniformly on $[0,1]$, evaluate $\theta^\star$, the observed labels y_i, and $\widehat{\theta}^t_\lambda$. We evaluate the excess risk $L(\widehat{\theta}^t_\lambda) - L(\theta^\star)$ with Monte Carlo sampling. We then report the *lowest excess risk* achieved across the regularization λ, and the *optimal regularization* used to achieve this loss. We plot lines of slope $^{2s\alpha}/_{1+2s\alpha}$ and $^{\alpha}/_{1+2s\alpha}$ respectively, with $s = (r+1/2) \wedge t$ in order to compare the statistical rates achieved in practice and in theory.

Results. Results for the logistic loss are available in fig. 1 and we also present results with least squares where the noise is Gaussian in section A.5.2. We set $\alpha = 2$, $r \in \{1/4, 41/4\}$, and we study the performance of Iterated Tikhonov estimators with $t \in \{1,3,8\}$. $t = 1$ corresponds to Tikhonov estimator and saturates at $r = 1/2$. $\mathsf{IT}(3)$ and $\mathsf{IT}(8)$ saturates at $r = 5/2$ and $r = 15/2$ respectively. Consequently, all estimators have optimal rates on the difficult task with $r = 1/4$; however, only IT exploits the additional regularity of the easy task, with $r = 41/4$. This experimentally shows that better sample complexity can be achieved when the learning task is easier and t is high, matching the rates predicted in theorem 4, which are $n^{-\alpha(1+2s)/1+\alpha(1+2s)}$, with $s = \min\{r, t-1/2\}$. Learning rates were estimated with an ordinary least square regression in log-log scale, and are given in table 1, where they are compared with the theoretical values. To conclude, we observe a slight improvement in absolute value of the excess risk in the range $r \ll t$, suggesting that IT is useful even when the learning task is hard. This could be because of lower constants for high t: *e.g.* we show that C_{bias} decays in $1/t^r$ when $t \geqslant r+1/2$, see theorem 2 in the appendix. We report in the appendix additional experimental results such as plots with the chosen regularization λ as a function of n, and plots on the ratio between the excess risk of $\mathsf{IT}(t)$ and Tikhonov, to show that the former is consistently better than the latter on easy tasks.

Table 1: Learning rate coefficients for capacity condition $\alpha = 2$ and various source condition assumption r. We estimate γ with ordinary least square with the model $L(\widehat{\theta}^t_\lambda) - L(\theta^\star) \propto n^{-\gamma}$. We display the coefficient we expect in theory, and the one we estimate.

	r	0.25	3.25	10.25
$t=1$	Theory	0.75	0.80	0.80
	Estimation	0.71	0.73	0.72
$t=3$	Theory	0.75	0.92	0.92
	Estimation	0.75	0.83	0.87
$t=8$	Theory	0.75	0.94	0.97
	Estimation	0.79	0.95	0.98

4.5 Conclusion

This section studies a well-known regularization scheme for least square, and extend it for the first time to other loss functions, which notably contain the logistic loss used for classification. We

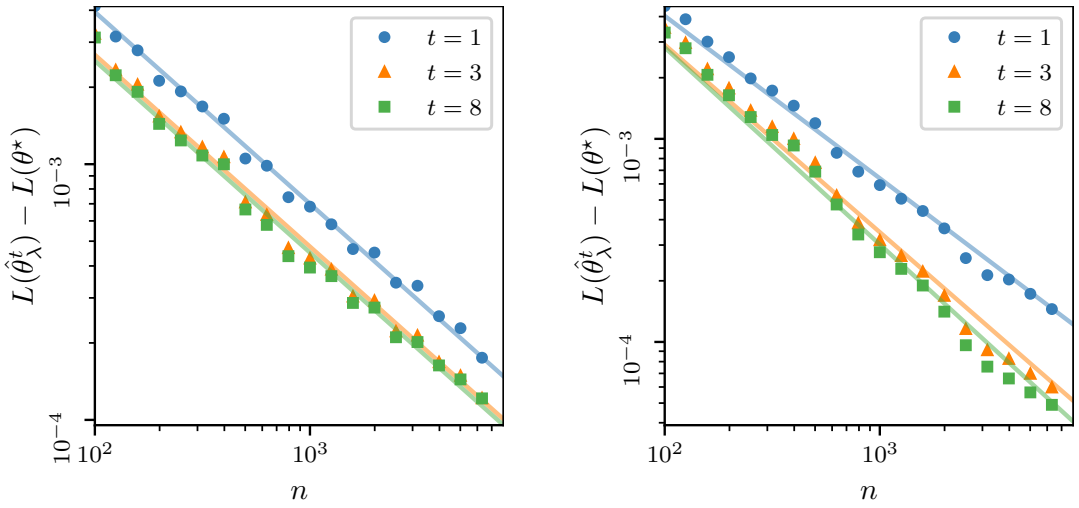


Figure 1: Excess risk for various Iterated Tikhonov estimators as a function of n. **Colors**: $t = 1$ (Tikhonov) estimator is shown in blue; $t = 3, 8$ in green, orange. **Left**: from a difficult problem, $r = 1/4, \alpha = 2$. **Right**: easy problem, $r = 41/4, \alpha = 2$. Plain lines are predicted by theory, with slope $-\alpha(1+2s)/1+\alpha(1+2s)$, $s = \min\{r, t - 1/2\}$ (see main text). All plots are averaged over 100 runs of the optimization procedure with different initialization.

prove that Iterated Tikhonov, corresponding to proximal point iterations, has optimal learning rates and higher qualification than Tikhonov, and as such could outperform it on easy tasks. We extend the scope of the theory of learning with generalized self concordant loss functions beyond standard Tikhonov regularization, which fills a gap in the previous theory, showing that it is possible to be fully adaptive to the regularity of the learning problem, without saturation effects. On top of this, we gave sufficient conditions to compute the estimator in practice, which is nontrivial by its sequential nature. Interesting research directions include related regularization schemes, such as boosting, but also implementations of the iterated Tikhonov procedure with sketching techniques as Nyström projections. The goal is to derive algorithms that are both optimal, in terms of statistical guarantees, and with reduced computational complexity, which is an aspect we will address in future work.

A Appendix to Beyond Tikhonov: Faster Learning with Self-Concordant Losses via Iterative Regularization

A.1 Settings, notations and assumptions

Given a separable Hilbert space $\mathcal{H}$, $\|\cdot\|$ denotes the norm in $\mathcal{H}$. For any operator A on $\mathcal{H}$, $\|A\|$ denotes its operator norm, and $\operatorname{Tr} A$ its trace norm. If A is a p.d operator, we denote by $\|\cdot\|_A = \left\|A^{1/2}\cdot\right\|$ the norm induced by A. We denote $\|A\|_{HS}$ the Hilbert Schmidt norm of A. We use the short-hand notation

$$A_\lambda = A + \lambda \mathbf{I},$$

where $\mathbf{I}$ is the identity. We denote by $a \wedge b$ the minimum of $\{a, b\}$, and $a \vee b$ its maximum.

A.1.1 Settings and technical assumptions

The settings in this subsection are the same as in [21]. We report them for completeness.

Let $\mathcal{X}$ a Borel input space, $\mathcal{Y}$ be a vector-valued output spaces, and ρ a probability distribution on $\mathcal{X} \times \mathcal{Y}$. We consider $\mathcal{H}$ to be a separable Hilbert space of functions from $\mathcal{X}$ to $\mathcal{Y}$. We consider a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ for measuring the fit between predictions and true labels. Given n observations $(x_1, y_1), \ldots, (x_n, y_n)$ i.i.d according to ρ, the goal is to build a measurable function $\widehat{\theta}$, which minimizes the expected loss

$$L(\widehat{\theta}) = \mathbb{E}_{x,y\sim\rho}\left[\ell(y, \widehat{\theta}(x))\right].$$

In this paper, we evaluate the quality of the estimator with probabilistic upper bounds on the *excess risk*

$$L(\widehat{\theta}) - \inf_{\theta\in\mathcal{H}} L(\theta) \leqslant K n^{-\gamma},$$

with probability greater than $1-\delta$. The rate of decay γ is referred to as the *learning rate* of the estimator. Our main assumption on the loss function is to be generalized self-concordant (GSC).

Assumption 5 (Generalized Self-Concordance)**.** *For any $z = x, y \in \mathcal{X} \times \mathcal{Y}$, the function $\ell_z : \mathcal{H} \to \mathbb{R}$ defined as $\ell_z(\theta) = \ell(y, \theta(x))$ for $\theta \in \mathcal{H}$ is convex and three times differentiable. Besides, there exists a set $\phi(z) \subset \mathcal{H}$ s.t*

$$\forall \theta \in \mathcal{H},\ \forall h, k \in \mathcal{H}, \quad \left|\nabla^3 \ell_z(\theta)\,[h,k,k]\right| \leqslant \sup_{g\in\phi(z)} |k\cdot g|\, \nabla^2 \ell_z(\theta)\,[k,k].$$

Next, we introduce the following quantities.

Definition 5 (Useful quantities)**.** *Let $\theta \in \mathcal{H}$. The following quantities are independant of the random variable $z \sim \rho$, either by taking the supremum over the support of ρ or by considering the expectation. Define:*

- *uniform bounds on the derivatives:*

$$B_1(\theta) = \sup_{z\in\operatorname{Supp}\rho} \|\nabla \ell_z(\theta)\|, \quad B_2(\theta) = \sup_{z\in\operatorname{Supp}\rho} \operatorname{Tr} \nabla^2 \ell_z(\theta);$$

- *the Hessian of the expected and empirical loss:*

$$\mathbf{H}(\theta) = \nabla^2\, \mathbb{E}\left[\ell_z(\theta)\right], \quad \widehat{\mathbf{H}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \nabla^2 \ell_{z_i}(\theta);$$

- *the function t, s.t:*

$$t(\theta) = \sup_{z\in\operatorname{Supp}\rho}\ \sup_{g\in\phi(z)} |\theta\cdot g|.$$

We make technical assumption to ensure that the loss function and its derivatives are well defined everywhere and that we can exchange expectation and derivative.

Assumption 6 (Technical assumptions)**.** *There exists* R *s.t* $\sup_{g \in \Phi(z)} \|g\| \leqslant R$ *almost surely;* $|\ell_z(0)|, \|\nabla \ell_z(0)\|, \operatorname{Tr} \mathbf{H}(\nabla^2 \ell_z(0))$ *are almost surely bounded.*

Using Prop. 2 of [21], we have that $B_1(\theta), B_2(\theta), L(\theta), \nabla L(\theta), \mathbf{H}(\theta)$ exist for all $\theta \in \mathcal{H}$, and

$$\nabla L(\theta) = \mathbb{E}\left[\nabla \ell_z(\theta)\right], \quad \mathbf{H}(\theta) = \mathbb{E}\left[\nabla^2 \ell_z(\theta)\right].$$

Finally, $\mathbf{H}(\theta)$ is trace-class, that is its trace is finite for any $\theta \in \mathcal{H}$. The same properties hold when considering $\widehat{\rho}$ instead of ρ, that is for the quantities $\widehat{L}(\theta), \nabla \widehat{L}(\theta)$ and $\widehat{\mathbf{H}}(\theta)$.

We make three key assumptions to obtain our learning rate.

Assumption 7 (Existence of a minimizer)**.** *There exists a minimizer of* L *in* $\mathcal{H}$*. There is* $\theta^\star \in \mathcal{H}$ *s.t*

$$L(\theta^\star) = \inf_{\theta \in \mathcal{H}} L(\theta^\star).$$

Assumption 8 (Source condition)**.** *There exists* $r > 0$ *and* $\nu \in \mathcal{H}$ *s. t*

$$\theta^\star = \mathbf{H}^r(\theta^\star)\nu.$$

The third assumption qualifies the ill-posedness of the problem:

Assumption 9 (Capacity condition)**.** *There exists* $\alpha > 1, s, S > 0$ *s.t*

$$s\lambda^{-1/\alpha} \leqslant \mathrm{df}_\lambda \leqslant S\lambda^{-1/\alpha}.$$

To understand the source and capacity condition, one must pay attention to the counterpart of the covariance operator for GSC loss function, that is the expected hessian at optimality. It is denoted with $\mathbf{H}(\theta^\star)$ throughout the paper. The source and capacity conditions are assumptions on the eigendecomposition of this operator. To better quantify these assumptions, take σ_j, e_j an eigenbasis of $\mathbf{H}(\theta^\star)$, with $\sigma_j > \sigma_{j+1}$.

The source condition is a smoothness assumption on $\theta^\star$. It amounts to assuming that the eigendecomposition of $\theta^\star$ on the basis of the Hessian decays faster than its spectrum. Indeed, rewriting assumption 8 we obtain

$$\|\nu\|^2 = \sum_{j \geqslant 1} \sigma_j^{-2r} \langle \theta^\star, e_j \rangle^2 < +\infty.$$

Assuming $r = 0$ simplifies to $\theta^\star \in \mathcal{H}$. Bigger r implies that the optimum can be well approximated by the first few eigenvectors (as $(\sigma_j^{-2r})_j$ goes quickly to infinity).

Similarly, the capacity condition is an assumption on the decay of the spectrum of the Hessian. Specifically, it assumes that the spectrum decays polynomially, i.e $\sigma_j \sim j^{-\alpha}$. As this operator is compact, we have $\alpha > 1$ for the $\sum_j j^{-\alpha}$ to be summable. Bigger α gives easier input space $\mathcal{X}$.

See section 4.2 in the main body of the paper for a discussion on the significance of these assumptions.

A.1.2 Basic results on GSC loss functions

Here, we present Prop. 4 of [21], which we then extend with an additional lemma.

Proposition 3 (Properties of GSC functions)**.** *Let* $\theta, \nu \in \mathcal{H}$, $\lambda \geqslant 0$*. The following properties hold:*

$$\widehat{\mathbf{H}}_\lambda(\theta) \preceq e^{t(\theta-\nu)} \widehat{\mathbf{H}}_\lambda(\nu) \tag{22}$$

$$\left\| \nabla \widehat{L}_\lambda(\theta) - \widehat{L}_\lambda(\nu) \right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta)} \leqslant \|\theta - \nu\|_{\widehat{\mathbf{H}}_\lambda(\theta)} \, \overline{\phi}\left(t\|\theta - \nu\|\right) \tag{23}$$

$$L_\lambda(\theta) - L_\lambda(\nu) - \nabla L_\lambda(\nu) \cdot (\theta - \nu) \leqslant \Psi\left(t\|\theta - \nu\|\right) \|\theta - \nu\|^2_{\mathbf{H}_\lambda(\theta)} \tag{24}$$

where $\phi : t \mapsto (1 - e^{-t})/t$ *and* $\Psi : t \mapsto (e^t - t - 1)/t^2$*. Moreover, if* $\nu, \xi \in \mathcal{H}$, $A : \mathcal{H} \to \mathcal{H}$ *commutes with* $\mathbf{H}(\xi)$*, then the following holds:*

$$e^{-t(\theta-\xi)} \underline{\phi}\left(t\|\nu - \theta\|\right) \|A(\nu - \theta)\|_{\mathbf{H}_\lambda(\xi)} \leqslant \|A(\nabla L_\lambda(\nu) - \nabla L_\lambda(\theta))\|_{\mathbf{H}_\lambda^{-1}(\xi)} \tag{25}$$

We slightly modify the lower bound gradient, which is crucial for obtaining higher qualification with IT.

Lemma 2 (Stacking operator on gradient bounds). *Let $\theta, \nu, \xi \in \mathcal{H}, \lambda > 0$. If $A : \mathcal{H} \to \mathcal{H}$ commutes with $\mathbf{H}(\xi)$, the following holds:*

$$e^{-t(\theta-\xi)}\underline{\phi}\left(t(\nu-\theta)\right)\|A(\nu-\theta)\|_{\mathbf{H}_\lambda(\xi)} \leqslant \|A(\nabla L_\lambda(\nu)-\nabla L_\lambda(\theta))\|_{\mathbf{H}_\lambda^{-1}(\xi)}.$$

Proof. Defining $\nu_s = \theta + s(\nu-\theta)$ for $s \in \{0,1\}$, we have:

$$A^2\left(\nabla L_\lambda(\nu)-\nabla L_\lambda(\theta)\right) = A^2\int_0^1 \mathbf{H}_\lambda(\nu_s)\left(\nu-\theta\right)ds,$$

$$\text{which implies}\quad \left\langle A^2\left(\nabla L_\lambda(\nu)-\nabla L_\lambda(\theta)\right), \nu-\theta\right\rangle = A^2\int_0^1 \left\langle \mathbf{H}_\lambda(\nu_s)\left(\nu-\theta\right), \nu-\theta\right\rangle ds.$$

We may then use the lower bound on the Hessian from eq. (22),

$$\mathbf{H}_\lambda(\nu_s) \succeq \mathbf{H}_\lambda(\xi)e^{-t(\nu_s-\xi)} \succeq \mathbf{H}_\lambda(\xi)e^{-t(\theta-\xi)}e^{-s\,t(\nu-\theta)},$$

where the second inequality comes from t satisfying the triangle inequality. Plugging this in the previous equation and using the fact that $\mathbf{H}(\xi)$ and A commute, we have that:

$$\begin{aligned}\int_0^1 \left\langle A^2\mathbf{H}_\lambda(\nu_s)\left(\nu-\theta\right), \nu-\theta\right\rangle ds &\geqslant e^{-t(\theta-\xi)}\int_0^1 e^{-s\,t(\nu-\theta)}ds\ \left\langle \mathbf{H}_\lambda(\xi)A\left(\nu-\theta\right), A\left(\nu-\theta\right)\right\rangle \\ &= e^{-t(\theta-\xi)}\underline{\phi}\left(t(\nu-\theta)\right)\left\langle \mathbf{H}_\lambda(\xi)A(\nu-\theta), A(\nu-\theta)\right\rangle,\end{aligned}$$

which gives the lower bound

$$e^{-t(\theta-\xi)}\underline{\phi}\left(t(\nu-\theta)\right)\|A(\nu-\theta)\|^2_{\mathbf{H}_\lambda(\xi)} \leqslant \left\langle A^2\left(\nabla L_\lambda(\nu)-\nabla L_\lambda(\theta)\right), \nu-\theta\right\rangle. \tag{26}$$

On the other hand, with Cauchy Schwartz inequality, we obtain:

$$\left\langle A^2(\nabla L_\lambda(\nu)-\nabla L_\lambda(\theta)), \nu-\theta\right\rangle \leqslant \|A(\nabla L_\lambda(\nu)-\nabla L_\lambda(\theta))\|_{\mathbf{H}_\lambda^{-1}(\xi)}\|A(\nu-\theta)\|_{\mathbf{H}_\lambda(\xi)}. \tag{27}$$

Combining the inequalities eqs. (26) and (27) and dividing by $\|A(\nu-\theta)\|_{\mathbf{H}_\lambda(\xi)}$, we obtain the result needed. □

A.2 Proof of Theorem 1

A.2.1 Error decomposition

Thanks to eq. (24), the excess risk is bounded by the distance between estimate in $\mathbf{H}(\theta^\star)$ norm with

$$L_\lambda(\widehat{\theta}^t_\lambda) - L_\lambda(\theta^\star) \leqslant \Psi\left(t(\widehat{\theta}^t_\lambda-\theta^\star)\right)\left\|\widehat{\theta}^t_\lambda-\theta^\star\right\|^2_{\mathbf{H}_\lambda(\theta^\star)}.$$

In order to compute $\|\widehat{\theta}^t_\lambda-\theta^\star\|_{\mathbf{H}(\theta^\star)}$, we need to go through an intermediate quantity ϑ. In the context of least squares and spectral filters, such quantity is usually defined to be

$$\vartheta = g_\lambda(\widehat{T})\widehat{S}^*\widehat{S}\theta^\star, \tag{28}$$

where:

- $\widehat{T} = \widehat{S}^*\widehat{S}$ is the *empirical covariance operator*, equal to $\sum_{i=1}^n \Psi(x_i)\otimes\Psi(x_i)$ when $\mathcal{H}$ is a RKHS with feature map Ψ (see remark 4);
- $\widehat{S} : \mathcal{H} \to \mathbb{R}^n$ is the *sampling operator*, with $\widehat{S}\theta = 1/\sqrt{n}(\theta(x_i),\dots,\theta(x_n))$;
- Its dual is $\widehat{S}^* : \mathbb{R}^n \to \mathcal{H}$, with $\widehat{S}^*y = 1/\sqrt{n}\sum_{i=1}^n y_i\Phi(x_i)$;

see [6] for details. Thus, the quantity in eq. (15) can be seen as the estimator trained on the *empirical noiseless distribution*, where we use $\widehat{S}\theta^\star$ instead of $y = (y_i)_{1\leqslant i\leqslant n}$. It is optimal in the sense that its bias $\|\vartheta - \theta^\star\|_{\widehat{T}}$ will be of the order of $\lambda^{r+1/2}$ and its variance $\left\|\widehat{\theta}_\lambda^t - \vartheta\right\|_{\widehat{T}}$ of the order of df_λ / n, leading to the optimal rates for least squares [8].

Expressing the quantity above as a proximal sequence is the key insight of the proof. It turns out that the following quantity obtains the same optimal decomposition.

Definition 6 (Error decomposition)**.** *Define the following quantity:*

$$\begin{aligned}\vartheta_\lambda^0 &= \theta^\star\\ \vartheta_\lambda^{k+1} &= \mathsf{prox}_{\widehat{L}/\lambda}(\vartheta_\lambda^k), \quad k \geqslant 0\end{aligned}$$

Remark 5. *In fact, the estimator above, when expressed with filters, has its (bias, variance) equals to the (variance, bias) of the estimator of eq.* (28)*. It is easy to change the intermediate quantity of definition 6 to match, but it introduces unnecessary burden with the notations.*

The purpose of next sections is to bound

$$\left\|\widehat{\theta}_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)} \leqslant \left\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\right\|_{\mathbf{H}(\theta^\star)} + \left\|\vartheta_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)}. \tag{29}$$

The first term will be the *bias* of the estimator (decreases with λ/t) while the second one will be the *variance* (decreases with t/λ and n). The intermediate quantity of definition 6 being very close to the one of eq. (28) used in [6], it is natural that the proof follows similarly.

A.2.2 Bounding the bias

Here, we proceed in bounding the bias, that is the quantity $\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\|_{\mathbf{H}(\theta^\star)}$.

Theorem 2 (Improved qualification of Iterated Tikhonov estimator)**.** *Let $\delta \in (0, 1]$. Recall the source condition of parameter $r, \|\nu\|$. Define the following conditions on the number of samples:*

$$\begin{aligned}\mathsf{H}_1 &: n \geqslant 24\frac{B_2^\star}{\lambda}\log\frac{16B_2^\star}{\lambda\delta},\\ \mathsf{H}_{1b} &: n \geqslant 8\frac{B_2^{\star 2}}{\lambda^2}\log^2\frac{4}{\delta},\\ \mathsf{H}_2 &: n \geqslant 2\left[1 \vee \left(\frac{2B_2^\star(t-1/2)^r}{\lambda^{s-1/2}}\right)^2\right]\log\frac{4}{\delta},\end{aligned}$$

Now assume:

$$\begin{aligned}\mathsf{H}_1 \quad & \text{if } r \leqslant 1/2,\\ \mathsf{H}_1 + \mathsf{H}_{1b} \quad & \text{if } 1/2 < r \leqslant 1,\\ \mathsf{H}_1 + \mathsf{H}_2 \quad & \text{if } r > 1.\end{aligned}$$

Then, with probability greater than $1-\delta$:

$$\left\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\right\|_{\mathbf{H}(\theta^\star)} \leqslant \sqrt{2}\mathsf{T}(r,t)\mathsf{P}_\lambda^t\lambda^s, \tag{30}$$

with $s = (r+1/2)\wedge t$,

$$\mathsf{T}(r,t) = \begin{cases}\|\nu\|\,(1\vee(B_2^\star+\lambda))2^r & \text{if } r\leqslant 1,\\ \|\nu\|\,\frac{w(r)+r}{(t-1/2)^r} & \text{if } r>1 \text{ and } r+1/2<t\\ \|\nu\|\,\frac{w(r)}{(t-1/2)^r} + B_2^{\star r-t+1/2} & \text{if } r>1 \text{ and } r+1/2\geqslant t,\end{cases} \tag{31}$$

$w(r) = r2^{\lfloor r\rfloor+1}B_2^{\star r}$*, and:*

$$\mathsf{P}_\lambda^t \overset{\text{def.}}{=} \prod_{k=1}^{t}\underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k)\right)e^{\widehat{t}(\vartheta_\lambda^k-\theta^\star)}.$$

This term is the optimal bias for LS with the usual excess risk decomposition. The saturation effect is explicit: we go from a bias decay in λ^t when $t \leqslant r+1/2$ to λ^r when the source condition saturates IT's regularization. That is, IT's estimator has a qualification of t, in the sense that it can exploits source condition up to $r = t - 1/2$. If $r > t - 1/2$, the estimator saturates and the learning rate becomes suboptimal.

Proof. This proof simply relies on the upper bound on gradients enabled by GSC functions. We will use lemma 2 for that purpose. Also, we will use the definition of a proximal sequence; that is, we have that

$$\forall k \leqslant t, \quad \nabla \widehat{L}(\widehat{\theta}_\lambda^k) + \lambda(\widehat{\theta}_\lambda^k - \widehat{\theta}_\lambda^{k-1}) = 0,$$

which is just another way of saying that we perform implicit gradient steps of size $1/\lambda$. Note that if you consider the *regularized* empirical loss $\widehat{L}_\lambda$, the previous equation gives

$$\nabla \widehat{L}_\lambda(\widehat{\theta}_\lambda^k) = \nabla \widehat{L}(\widehat{\theta}_\lambda^k) + \lambda \widehat{\theta}_\lambda^k = -\lambda(\widehat{\theta}_\lambda^k - \widehat{\theta}_\lambda^{k-1}) + \lambda \widehat{\theta}_\lambda^k = \lambda \widehat{\theta}_\lambda^{k-1}. \tag{32}$$

Changing the norm. We first change the norm we operate on:

$$\begin{aligned}
\left\| \widehat{\theta}_\lambda^t - \vartheta_\lambda^t \right\|_{\mathbf{H}(\theta^\star)} &\leqslant \left\| \widehat{\mathbf{H}}_\lambda^{-1/2}(\theta^\star) \mathbf{H}^{1/2}(\theta^\star) \right\| \left\| \widehat{\theta}_\lambda^t - \vartheta_\lambda^t \right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} \\
&\leqslant \left\| \widehat{\mathbf{H}}_\lambda^{-1/2}(\theta^\star) \mathbf{H}_\lambda^{1/2}(\theta^\star) \right\| \left\| \widehat{\theta}_\lambda^t - \vartheta_\lambda^t \right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}.
\end{aligned}$$

We bound the operator norm using proposition 9 in section A.4, with $\mathcal{F}_\lambda = B_2^\star/\lambda$. We obtain:

$$H_1 : n \geqslant 24 \frac{B_2^\star}{\lambda} \log \frac{8 B_2^\star}{\lambda \delta} \implies \left\| \widehat{\mathbf{H}}_\lambda^{-1/2}(\theta^\star) \mathbf{H}_\lambda^{1/2}(\theta^\star) \right\| \leqslant \sqrt{2}. \tag{33}$$

We now proceed in bounding the distance between estimates, that is the quantity $\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}$. We denote

$$s = (r + 1/2) \wedge t. \tag{34}$$

Upper bound on gradients. Use lemma 2 on $\widehat{L}_\lambda$ along with eq. (32) to have:

$$\begin{aligned}
\left\| \widehat{\theta}_\lambda^t - \vartheta_\lambda^t \right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} &\leqslant \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^t - \vartheta_\lambda^t) \right) e^{\widehat{t}(\vartheta_\lambda^t - \theta^\star)} \left\| \nabla \widehat{L}_\lambda(\widehat{\theta}_\lambda^t) - \nabla \widehat{L}_\lambda(\vartheta_\lambda^t) \right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \\
&= \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^t - \vartheta_\lambda^t) \right) e^{\widehat{t}(\vartheta_\lambda^t - \theta^\star)} \left\| \lambda(\widehat{\theta}_\lambda^{t-1} - \vartheta_\lambda^{t-1}) \right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \\
&= \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^t - \vartheta_\lambda^t) \right) e^{\widehat{t}(\vartheta_\lambda^t - \theta^\star)} \left\| \lambda \widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)(\widehat{\theta}_\lambda^{t-1} - \vartheta_\lambda^{t-1}) \right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}.
\end{aligned}$$

Let us detail the recursion. Let $k \leqslant t$. Then, the following inequality holds, thanks to lemma 2:

$$\begin{aligned}
\left\| \lambda^k \widehat{\mathbf{H}}_\lambda^{-k}(\theta^\star)(\widehat{\theta}_\lambda^{t-k} - \vartheta_\lambda^{t-k}) \right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} &\leqslant \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^{t-k} - \vartheta_\lambda^{t-k}) \right) e^{\widehat{t}(\vartheta_\lambda^{t-k} - \theta^\star)} \\
&\quad \left\| \lambda^k \widehat{\mathbf{H}}_\lambda^{-k}(\theta^\star) \left(\nabla \widehat{L}_\lambda(\widehat{\theta}_\lambda^{t-k}) - \nabla \widehat{L}_\lambda(\vartheta_\lambda^{t-k}) \right) \right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \\
&= \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^{t-k} - \vartheta_\lambda^{t-k}) \right) e^{\widehat{t}(\vartheta_\lambda^{t-k} - \theta^\star)} \\
&\quad \left\| \lambda^{k+1} \widehat{\mathbf{H}}_\lambda^{-k}(\theta^\star) \left(\widehat{\theta}_\lambda^{t-(k+1)} - \vartheta_\lambda^{t-(k+1)} \right) \right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \\
&= \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^{t-k} - \vartheta_\lambda^{t-k}) \right) e^{\widehat{t}(\vartheta_\lambda^{t-k} - \theta^\star)} \\
&\quad \left\| \lambda^{k+1} \widehat{\mathbf{H}}_\lambda^{-(k+1)}(\theta^\star) \left(\widehat{\theta}_\lambda^{t-(k+1)} - \vartheta_\lambda^{t-(k+1)} \right) \right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}.
\end{aligned}$$

Thus, unfolding the recursion, we obtain:

$$\begin{aligned} \left\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} &\leqslant \mathsf{P}_\lambda^t \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-t}(\theta^\star)\theta^\star\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}, \\ \text{with} \quad \mathsf{P}_\lambda^t &\overset{\text{def.}}{=} \prod_{k=1}^{t} \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k)\right) e^{\widehat{t}(\vartheta_\lambda^k - \theta^\star)}. \end{aligned} \tag{35}$$

We now use the source condition on $\theta^\star$. Recall that it gives

$$\theta^\star = \mathbf{H}^r(\theta^\star)\nu,$$

for some $\nu \in \mathcal{H}$. Thus, we have:

$$\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-t}(\theta^\star)\theta^\star\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} = \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\mathbf{H}^r(\theta^\star)\nu\right\| \tag{36}$$

$$\leqslant \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\| \|\nu\| \tag{37}$$

We need to distinguish between $r \leqslant 1$ and $r > 1$ to bound the operator norm

$$\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\|.$$

Case $r \leqslant 1$. We use the following decomposition:

$$\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\| \leqslant \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\widehat{\mathbf{H}}_\lambda^r(\theta^\star)\right\| \left\|\widehat{\mathbf{H}}_\lambda^{-r}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\|.$$

The first term is bounded like this:

$$\begin{aligned} \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)+r}(\theta^\star)\right\| &\leqslant \sup_{\widehat{\sigma}_{\min} < \sigma \leqslant \mathsf{B}_2^\star} \frac{\lambda^t}{(\sigma+\lambda)^{t-1/2-r}} \\ &\leqslant \lambda^s \begin{cases} 1 & \text{if } r + 1/2 < t \\ \mathsf{B}_2^\star + \lambda & \text{if } t = 1 \text{ and } r > 1/2. \end{cases} \end{aligned}$$

This illustrates that Tikhonov regularization ($t = 1$) saturates at $r = 1/2$.

For the second term, write

$$\left\|\widehat{\mathbf{H}}_\lambda^{-r}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\| \leqslant \left\|\widehat{\mathbf{H}}_\lambda^{-r}(\theta^\star)\mathbf{H}_\lambda^r(\theta^\star)\right\|$$

Then, use the Hermitian inequalities of eq. (68) in lemma 4, then use the concentration inequalities of proposition 9. Both can be found in section A.4. In details:

- If $r \leqslant 1/2$, use then the concentration inequality of eq. (65):

$$\begin{aligned} \left\|\widehat{\mathbf{H}}_\lambda^{-r}(\theta^\star)\mathbf{H}_\lambda^r(\theta^\star)\right\| &\leqslant \left\|\widehat{\mathbf{H}}_\lambda^{-1/2}(\theta^\star)\mathbf{H}_\lambda^{1/2}(\theta^\star)\right\|^{2r} \\ &\leqslant 2^{r/2} \text{ if } \mathsf{H}_1. \end{aligned}$$

  with confidence $1 - \delta$.

- If $r > 1/2$, use the concentration inequality of eq. (66):

$$\begin{aligned} \left\|\widehat{\mathbf{H}}_\lambda^{-r}(\theta^\star)\mathbf{H}_\lambda^r(\theta^\star)\right\| &\leqslant \left\|\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)\mathbf{H}_\lambda(\theta^\star)\right\|^r \\ &\leqslant 2^r \text{ if } \mathsf{H}_{1b} : n \geqslant 8\frac{\mathsf{B}_2^{\star 2}}{\lambda^2}\log^2\frac{2}{\delta}. \end{aligned}$$

All in all, after simplification, the bound on the operator norm when $r \leqslant 1$ reads

$$\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\| \leqslant \lambda^s(1 \vee (\mathsf{B}_2^\star + \lambda))2^r \quad \text{if} \begin{cases} \mathsf{H}_1 & \text{when } r \leqslant 1/2 \\ \mathsf{H}_{1b} & \text{when } r > 1/2, \end{cases} \tag{38}$$

with confidence $1 - \delta$. We now turn to the case $r > 1$.

Case $r > 1$**.** We tackle this case with a different decomposition:

$$\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\| \leqslant \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\widehat{\mathbf{H}}^r(\theta^\star)\right\| + \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)(\mathbf{H}^r(\theta^\star) - \widehat{\mathbf{H}}^r(\theta^\star))\right\|$$

Looking at the first term; recalling that $\widehat{\mathbf{H}}(\theta^\star) \leqslant B_2^\star$, we have:

$$\begin{aligned}
\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\widehat{\mathbf{H}}^r(\theta^\star)\right\| &\leqslant \sqrt{\lambda} \sup_{0<\sigma\leqslant B_2^\star} \left(\frac{\lambda}{\lambda+\sigma}\right)^{t-1/2} \sigma^r \\
&\leqslant \lambda^s \begin{cases} \frac{r}{(t-1/2)^r} & \text{if } \ r+1/2 < t \\ \frac{{B_2^\star}^r}{\left({B_2^\star}^r+\lambda\right)^{t-1/2}} & \text{otherwise} \end{cases} \\
&\leqslant \lambda^s \begin{cases} \frac{r}{(t-1/2)^r} & \text{if } \ r+1/2 < t \\ {B_2^\star}^{r-t+1/2} & \text{otherwise} \end{cases}
\end{aligned}$$

where we used the computation of lemma 5. The second term can be upper bounded as follows:

$$\begin{aligned}
\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)(\mathbf{H}^r(\theta^\star) - \widehat{\mathbf{H}}^r(\theta^\star))\right\| &\leqslant \left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\right\| \left\|\mathbf{H}^r(\theta^\star) - \widehat{\mathbf{H}}^r(\theta^\star)\right\| \\
&\leqslant w(r)\sqrt{\lambda}\left\|\mathbf{H}(\theta^\star) - \widehat{\mathbf{H}}(\theta^\star)\right\| \\
&\leqslant w(r)\frac{\lambda^s}{(t-1/2)^r} \text{ if } H_2 : n \geqslant 2\left(1 \vee \left(\frac{2B_2^\star(t-1/2)^r}{\lambda^{s-1/2}}\right)^2\right)\log\frac{2}{\delta}
\end{aligned}$$

with confidence $1-\delta$. We applied eq. (69) in lemma 4 on the second inequality, and eq. (67) in proposition 9 for the last inequality, both of which can be found in section A.4. We used:

$$w(r) = r2^{\lfloor r\rfloor+1}{B_2^\star}^r. \tag{39}$$

Thus, the bound on the operator norm when $r > 1$ reads:

$$\left\|\lambda^t \widehat{\mathbf{H}}_\lambda^{-(t-1/2)}(\theta^\star)\mathbf{H}^r(\theta^\star)\right\| \leqslant \lambda^s \begin{cases} \frac{w(r)+r}{(t-1/2)^r} & \text{if } \ r+1/2 < t \\ \frac{w(r)}{(t-1/2)^r} + {B_2^\star}^{r-t+1/2} & \text{otherwise} \end{cases} \quad \text{if } H_2, \tag{40}$$

with confidence $1-\delta$.

Gluing things together. We proceed to the conclusion. Define the following conditions:

$$\begin{aligned}
H_1 &: n \geqslant 24\frac{B_2^\star}{\lambda}\log\frac{16B_2^\star}{\lambda\delta}, \\
H_{1b} &: n \geqslant 8\frac{{B_2^\star}^2}{\lambda^2}\log^2\frac{4}{\delta}, \\
H_2 &: n \geqslant 2\left[1 \vee \left(\frac{2B_2^\star(t-1/2)^r}{\lambda^{s-1/2}}\right)^2\right]\log\frac{4}{\delta},
\end{aligned}$$

where we replace δ by $\delta/2$ in order to have bounds with confidence $1-\delta/2$, so that the overall bound holds with confidence $1-\delta$ (in fact, $1-\delta/2$ in the first case). Now assume the following:

$$\begin{aligned}
H_1 \quad & \text{if } \ r \leqslant 1/2, \\
H_1 + H_{1b} \quad & \text{if } \ 1/2 < r \leqslant 1, \\
H_1 + H_2 \quad & \text{if } \ 1 < r.
\end{aligned}$$

Then, we can chain the inequalities of eqs. (33), (37), (38) and (40). We obtain:

$$\left\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\right\|_{\mathbf{H}(\theta^\star)} \leqslant \sqrt{2}\,\|\nu\|\,P_\lambda^t\lambda^s \begin{cases} (1 \vee (B_2^\star + \lambda))2^r & \text{if } \ r \leqslant 1, \\ \frac{w(r)+r}{(t-1/2)^r} & \text{if } \ r > 1 \text{ and } r+1/2 < t, \\ \frac{w(r)}{(t-1/2)^r} + {B_2^\star}^{r-t+1/2} & \text{if } \ r > 1 \text{ and } r+1/2 \geqslant t, \end{cases} \tag{41}$$

with confidence $1-\delta$. □

A.2.3 Bounding the variance

After bounding the bias, we study the variance term: $\|\vartheta_\lambda^t - \theta^\star\|_{\mathbf{H}(\theta^\star)}$.

Theorem 3 (Optimal variance of Iterated Tikhonov estimator)**.** *Let $\delta \in (0, 1]$. Recall the definition of the degrees of freedom df_λ. Define the following conditions on the number of samples:*

$$\mathsf{H}_1 : n \geqslant 24 \frac{B_2^\star}{\lambda} \log \frac{16 B_2^\star}{\lambda \delta},$$
$$\mathsf{H}_3 : n \geqslant 2 \frac{B_1^{\star 2}}{\lambda \, \mathsf{df}_\lambda} \log \frac{4}{\delta}.$$

Then, with probability greater than $1 - \delta$:

$$\left\|\vartheta_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)} \leqslant 4\sqrt{2} t R_\lambda^t \sqrt{\frac{\mathsf{df}_\lambda}{n}} \cdot \sqrt{\log 2/\delta},$$

where we introduced:

$$R_\lambda^t \overset{\text{def.}}{=} \prod_{k=1}^{t} \underline{\phi}^{-1}\left(\widehat{t}(\vartheta_\lambda^k - \theta^\star)\right).$$

Proof. The proof begins similarly to the study of the bias term theorem 2.

Changing the norm. We have the following bound (proof of theorem 2, eq. (33)):

$$\mathsf{H}_1 : n \geqslant 24 \frac{B_2^\star}{\lambda} \log \frac{8 B_2^\star}{\lambda \delta} \implies \left\|\vartheta_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)} \leqslant \sqrt{2} \left\|\vartheta_\lambda^t - \theta^\star\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}. \tag{42}$$

Upper bounds on gradient. To ease the notation, we denote by $a_k = \underline{\phi}^{-1}\left(\widehat{t}(\vartheta_\lambda^k - \theta^\star)\right)$. We have, thanks to the lower bound on gradient of eq. (25):

$$\begin{aligned}
\left\|\vartheta_\lambda^t - \theta^\star\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} &\leqslant a_t \left\|\nabla\widehat{L}_\lambda(\vartheta_\lambda^t) - \nabla\widehat{L}_\lambda(\theta^\star)\right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \\
&= a_t \left\|\lambda(\vartheta_\lambda^{t-1} - \theta^\star) - \nabla\widehat{L}(\theta^\star)\right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \\
&= a_t \left\|\lambda\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)(\vartheta_\lambda^{t-1} - \theta^\star)\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} + a_t \left\|\nabla\widehat{L}(\theta^\star)\right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \\
&\leqslant a_t a_{t-1} \left\|\lambda^2\widehat{\mathbf{H}}_\lambda^{-2}(\theta^\star)(\vartheta_\lambda^{t-2} - \theta^\star)\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} + \left[a_t + a_{t-1}\left\|\lambda\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)\right\|\right] \left\|\nabla\widehat{L}(\theta^\star)\right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)}
\end{aligned}$$

We can unfold the recursion. The first term will disappears thanks to $\vartheta_\lambda^0 = \theta^\star$, and we are left with:

$$\left\|\vartheta_\lambda^t - \theta^\star\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} \leqslant \sum_{k=0}^{t-1} \left(\prod_{i=t-k}^{t} a_i\right) \left\|\lambda^k\widehat{\mathbf{H}}_\lambda^{-k}(\theta^\star)\right\| \left\|\nabla\widehat{L}(\theta^\star)\right\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \tag{43}$$

$$\leqslant R_\lambda^t \left\|\widehat{\mathbf{H}}_\lambda^{-1/2}(\theta^\star)\mathbf{H}_\lambda^{1/2}(\theta^\star)\right\| \left(\sum_{k=0}^{t-1} \left\|\lambda^k\widehat{\mathbf{H}}_\lambda^{-k}(\theta^\star)\right\|\right) \left\|\nabla\widehat{L}(\theta^\star)\right\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)}, \tag{44}$$

$$\text{where} \quad R_\lambda^t \overset{\text{def.}}{=} \prod_{k=1}^{t} \underline{\phi}^{-1}\left(\widehat{t}(\vartheta_\lambda^k - \theta^\star)\right). \tag{45}$$

Consider the prefactor of $\|\nabla\widehat{L}(\theta^\star)\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)}$. We will bound $\|\widehat{\mathbf{H}}_\lambda^{-1/2}(\theta^\star)\mathbf{H}_\lambda^{1/2}(\theta^\star)\|$ by $\sqrt{2}$ with the same concentration argument as for the bias. The sum is more difficult to deal with. By computing the supremum of $\sigma \mapsto \lambda^k/(\sigma + \lambda)^k$ we would find that the first $\lfloor t/2 \rfloor$ terms have their maximum in 0. We would end up with a bound for the sum of the order of $t/2$. We rather use the simpler, if not optimal, following bound:

$$\sum_{k=0}^{t-1} \left\|\lambda^k\widehat{\mathbf{H}}_\lambda^{-k}(\theta^\star)\right\| \leqslant t.$$

It is suboptimal, but of the same order of an exact computation of the operator norm. Thus, we now have:

$$\left\|\vartheta_\lambda^t - \theta^\star\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} \leqslant \sqrt{2} t R_\lambda^t \left\|\nabla\widehat{L}(\theta^\star)\right\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)} \quad \text{when } H_1 \tag{46}$$

Bounding the gradient $\|\nabla\widehat{L}(\theta^\star)\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)}$. We use a plain Bernstein inequality to bound the gradient, as in proposition 9:

$$\mathbf{H}_\lambda^{-1/2}(\theta^\star)\nabla\widehat{L}(\theta^\star) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{H}_\lambda^{-1/2}(\theta^\star)\nabla\ell_{z_i}(\theta^\star).$$

We have

$$\sup_{z\in \mathsf{Supp}\,\rho} \|\nabla\ell_z(\theta^\star)\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)} \leqslant \frac{B_1^\star}{\sqrt{\lambda}},$$
$$\text{and} \quad \mathbb{E}_{z\sim\rho}\left[\|\nabla\ell_z(\theta^\star)\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)}\right]^2 \overset{\text{def.}}{=} \mathsf{df}_\lambda\,.$$

With confidence $1-\delta$, we now have

$$\left\|\nabla\widehat{L}(\theta^\star)\right\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)} \leqslant \frac{B_1^\star}{\sqrt{\lambda}}\frac{2\log 2/\delta}{n} + \sqrt{\mathsf{df}_\lambda \frac{2\log 2/\delta}{n}}.$$

We simplify this equation. Assuming

$$H_3 : n \geqslant 2\frac{B_1^{\star 2}}{\lambda\,\mathsf{df}_\lambda}\log\frac{2}{\delta}$$

we get the bound:

$$\left\|\nabla\widehat{L}(\theta^\star)\right\|_{\mathbf{H}_\lambda^{-1}(\theta^\star)} \leqslant 2\sqrt{2}\sqrt{\mathsf{df}_\lambda \frac{2\log 2/\delta}{n}}. \tag{47}$$

Gluing things together. All in all, we can glue together the inequalities in eqs. (42), (46) and (47). We obtain:

$$\left\|\vartheta_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)} \leqslant 4\sqrt{2} t R_\lambda^t \sqrt{\frac{\mathsf{df}_\lambda}{n}} \cdot \sqrt{\log 2/\delta} \quad \text{when } H_1 + H_3$$

with confidence $1-2\delta$. We obtain the statement of the theorem by replacing δ with $\delta/2$, so that the result holds with confidence $1-\delta$. □

A.2.4 Conditions for non-exponentials prefactors

The prefactors P_λ^t and R_λ^t are hard to bound; they can depend exponentially on $\|\theta^\star\|$ in the worst case [21]. The purpose of this section is to give sufficient conditions on the number of samples n for those quantities to turn constant. The key quantity to compare to is the *Dikin radius* [21, 20].

Definition 7 (Dikin radius)**.** *For $\theta \in \mathcal{H}$ and $\lambda > 0$, define $r_\lambda(\theta)$ s.t*

$$\frac{1}{r_\lambda(\theta)} = \sup_{z\in\mathsf{Supp}\,\rho}\ \sup_{g\in\phi(z)} \|g\|_{\mathbf{H}_\lambda^{-1}(\theta)}\,. \tag{48}$$

The inverse of the Dikin radius can be upper bounded by $R/\sqrt{\lambda}$. However, we prefer keeping bounds in $r_\lambda^\star$. Indeed, they take into account the geometry of the loss function around the optimum, and are thus much more precise.

Note that in the following, we might be content with the *empirical* Dikin radius $\widehat{r_\lambda(\theta)}$, ie. replacing ρ by $\widehat{\rho}$ in the previous definition. So as not to laden the notations and have something independent of the sampling, we use the fact that $\mathsf{Supp}\,\widehat{\rho} \subset \mathsf{Supp}\,\rho$ to ensure that:

$$\frac{1}{\widehat{r_\lambda(\theta)}} \leqslant \frac{1}{r_\lambda(\theta)} \quad \text{and} \quad \widehat{t}(\cdot) \leqslant t(\cdot).$$

Finally, we will use the following notation:

$$r_\lambda^\star \overset{\text{def.}}{=} \underset{\lambda}{r}(\theta^\star). \tag{49}$$

Pefactor of the variance We first proceed with the prefactor of the variance R_λ^t.

Proposition 4 (Constant prefactor for the variance)**.** *The following condition:*

$$H_4 : n \geqslant 8(et)^2 \left(4 \vee C^2 t^2\right) \frac{df_\lambda}{r_\lambda^\star} \log 2/\delta, \tag{50}$$

where $C \leqslant 0.8$ is a constant, is sufficient to guarantee that

$$R_\lambda^t \overset{\text{def.}}{=} \prod_{k=1}^{t} \underline{\phi}^{-1}\left(\widehat{t}(\vartheta_\lambda^k - \theta^\star)\right) \leqslant e. \tag{51}$$

Proof.

A first bound. Note that:

$$\begin{aligned} t(\vartheta_\lambda^t - \theta^\star) &= \sup_{z \in \operatorname{Supp} \rho} \sup_{g \in \phi(z)} \left| g \cdot (\vartheta_\lambda^t - \theta^\star) \right| \\ &\leqslant \sup_{z \in \operatorname{Supp} \rho} \sup_{g \in \phi(z)} \|g\|_{\widehat{\mathbf{H}}_\lambda^{-1}(\theta^\star)} \left\|\vartheta_\lambda^t - \theta^\star\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}, \end{aligned}$$

which gives us a bound we will use multiple times:

$$t(\vartheta_\lambda^t - \theta^\star) \leqslant \frac{\|\vartheta_\lambda^t - \theta^\star\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}}{r_\lambda^\star}. \tag{52}$$

We simply used the definition of the Dikin radius in eq. (48).

We now use an upper bound of the numerator, available in the proof of theorem 3:

$$\begin{aligned} & t(\vartheta_\lambda^t - \theta^\star) \leqslant R_\lambda^t \left[2\sqrt{2} t \sqrt{\log 2/\delta}\right] \sqrt{\frac{df_\lambda}{n r_\lambda^\star}} \\ \iff \quad & t(\vartheta_\lambda^t - \theta^\star) \underline{\phi}\left(t(\vartheta_\lambda^t - \theta^\star)\right) \leqslant R_\lambda^{t-1} \left[2\sqrt{2} t \sqrt{\log 2/\delta}\right] \sqrt{\frac{df_\lambda}{n r_\lambda^\star}} \overset{\text{def.}}{=} X_{t-1}. \end{aligned}$$

Now, using the fact that $x\underline{\phi}(x) = 1 - e^{-x}$, we get that

$$\begin{aligned} t(\vartheta_\lambda^t - \theta^\star) &\leqslant -\log(1 - X_{t-1}) \\ a_t \overset{\text{def.}}{=} \underline{\phi}^{-1}\left(t(\vartheta_\lambda^t - \theta^\star)\right) &\leqslant -X_{t-1}^{-1} \log(1 - X_{t-1}) \overset{\text{def.}}{=} h(X_{t-1}). \end{aligned} \tag{53}$$

Recursion hypotheses. The idea is to ensure:

1. $X_{k-1} \leqslant 1/2$ so that
$$h(X_{k-1}) \leqslant 1 + C X_{k-1}$$
with C a numeric constant s.t $h(1/2) = 1 + C/2$, which implies that $C \leqslant 0.8$. We are simply upper bounding h which is convex on $[0, 1/2]$.

2. $a_k \leqslant 1 + 1/t$ for all $k \leqslant t$, so that we can have:
$$\begin{aligned} R_\lambda^t = \prod_{k=1}^{t} a_k &= \exp \sum_{k=1}^{t} \log(a_k) \\ &\leqslant \exp \sum_{k=1}^{t} \log(1 + 1/t) \leqslant e. \end{aligned}$$

Recursion. Set $k = 1$. Then $R_\lambda^0 = 1$ and to have

$$X_0 \leqslant 1/2 \quad \text{that is} \quad \left[2\sqrt{2}t\sqrt{\log 2/\delta}\right]\sqrt{\frac{df_\lambda}{nr_\lambda^\star}} \leqslant \frac{1}{2},$$

it is sufficient to have

$$n \geqslant N_0 \overset{\text{def.}}{=} 32t^2\frac{df_\lambda}{r_\lambda^\star}\log 2/\delta.$$

We want to enforce

$$a_1 \leqslant 1 + 1/t.$$

A sufficient condition is

$$\begin{aligned} h(X_0) \leqslant 1 + CX_0 \leqslant 1 + 1/t &\iff X_0 \leqslant 1/tC \\ &\impliedby n \geqslant N_0' \overset{\text{def.}}{=} 8t^4C^2\frac{df_\lambda}{r_\lambda^\star}\log 2/\delta. \end{aligned}$$

Now, let $k < n$. Assume the two conditions hold at step $k-1$. Then, $R_\lambda^{k-1} \leqslant e$ and

$$n \geqslant N_{k-1} \overset{\text{def.}}{=} 32(et)^2\frac{df_\lambda}{r_\lambda^\star}\log 2/\delta \quad \text{implies} \quad X_{k-1} \leqslant \frac{1}{2}.$$

Likewise,

$$n \geqslant N_{k-1}' \overset{\text{def.}}{=} 8t^4(Ce)^2\frac{df_\lambda}{r_\lambda^\star}\log 2/\delta$$

gives

$$X_{k-1} \leqslant 1/tC, \quad \text{so that} \quad a_k \leqslant 1 + 1/k.$$

Conclusion. All in all, requiring

$$H_4 : n \geqslant 8(et)^2\left(4 \vee C^2t^2\right)\frac{df_\lambda}{r_\lambda^\star}\log 2/\delta$$

is sufficient to have $R_\lambda^k \leqslant e$, for any $k \leqslant t$. □

Pefactor of the bias The prefactor of the bias can be treated similarly. The only difficulty comes from the large number of subcases. Remember from theorem 2 that we have, with appropriate hypotheses,

$$\left\|\widehat{\theta}_\lambda^t - \vartheta_\lambda^t\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} \leqslant P_\lambda^t T(r,t)\lambda^s, \quad \text{with } s = (r+1/2)\wedge t. \tag{54}$$

Proposition 5 (Constant prefactor for the bias)**.** *Assume* H_4 *and*

$$H_5 : \lambda \leqslant L \overset{\text{def.}}{=} \left[e^{t+2}T(r,t)(2\wedge Ct)\right]^{-1/(r+1/2\wedge 1)}.$$

Then

$$P_\lambda^t \leqslant e^{t+2}.$$

Proof. The proof is almost identical to the proof of proposition 4. Let us simply point out the differences. We will drop the dependance of T on r, t in the notation for simplicity.

A first bound. Here, we have that:

$$t\left(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k\right) \leqslant \frac{\left\|\widehat{\theta}_\lambda^k - \vartheta_\lambda^k\right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}}{r_\lambda^\star} \leqslant \frac{P_\lambda^t T \lambda^s}{r_\lambda^\star}, \quad \text{with } s = r + 1/2 \wedge k. \tag{55}$$

We used eq. (54) in the second inequality. Recall the definition

$$P_\lambda^t \overset{\text{def.}}{=} \prod_{k=1}^{t} \underline{\phi}^{-1}\left(\widehat{t}(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k)\right) e^{\widehat{t}(\vartheta_\lambda^k - \theta^\star)}.$$

Thanks to H_4, we have $R_\lambda^t \leqslant e$. Specifically, noting that $\underline{\phi}^{-1}(x) \geqslant x$, we have from the proof of proposition 4:

$$1 + 1/t \geqslant \underline{\phi}^{-1}\left(t(\vartheta_\lambda^k - \theta^\star)\right) \geqslant t(\vartheta_\lambda^k - \theta^\star) \implies \prod_{k=1}^{t} e^{t(\vartheta_\lambda^k - \theta^\star)} \leqslant e^{t+1}.$$

Thus, we have that

$$P_\lambda^k \leqslant \underbrace{\prod_{i=1}^{k} \underline{\phi}^{-1}\left(t(\widehat{\theta}_\lambda^i - \vartheta_\lambda^i)\right)}_{Q_\lambda^k} e^{t+1}.$$

Dividing both sides in the eq. (55) with $\underline{\phi}^{-1}\left(t(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k)\right)$, we obtain that

$$t\left(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k\right) \underline{\phi}\left(t(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k)\right) \leqslant \frac{Q_\lambda^{k-1} T e^{t+1} \lambda^s}{r_\lambda^\star},$$

and we can apply the same reasoning as for the variance. Using $t\underline{\phi}(t) = 1 - e^{-t}$, we have

$$a_k \overset{\text{def.}}{=} \underline{\phi}^{-1}\left(t(\widehat{\theta}_\lambda^k - \vartheta_\lambda^k)\right) \leqslant -X_{k-1}^{-1} \log(1 - X_{k-1})$$
$$X_{k-1} \overset{\text{def.}}{=} Q_\lambda^{k-1} T e^{t+1} \lambda^s / r_\lambda^\star.$$

Recursion. We then do the exact same reasoning to the variance, that is require at each step $X_{k-1} \leqslant 1/2$ and $a_k \leqslant 1 + 1/t$. Here, this amounts to require

$$\lambda \leqslant L_s \overset{\text{def.}}{=} \left[e^{t+2} T(2 \wedge Ct)\right]^{-1/s}.$$

The L_s is increasing with s. So

$$\forall k \leqslant t, \quad L_s \leqslant L_{r+1/2\wedge 1} \overset{\text{def.}}{=} L, \quad \text{with } s = r + 1/2 \wedge k.$$

Conclusion. Requiring

$$H_5 : \lambda \leqslant L \overset{\text{def.}}{=} \left[e^{t+2} T(2 \wedge Ct)\right]^{-1/(r+1/2\wedge 1)}$$

is sufficient to ensure $Q_\lambda^t \leqslant e$, so that $P_\lambda^t \leqslant e^{t+2}$. □

A.2.5 Optimal rates for IT estimator

The bound on the bias and the variance holds if the number of samples is "high enough". The purpose of next proposition is to merge all these hypotheses together. Precisely, the hypotheses requires in each regime are summed up in table 2.

Table 2: Hypotheses needed to bound the bias and the variance, depending on the source condition parameter r.

Source condition	Bias	Variance	Numerical prefactors
$0 < r \leqslant 1/2$	H_1		
$1/2 < r < 1$	$H_1 + H_{1b}$	$H_1 + H_3$	$H_4 + H_5$
$r \geqslant 1$	$H_1 + H_2$		

Proposition 6 (Satisfying the hypotheses H_{1-5} with bounds on n and λ)**.** *The following relations hold:*

$$n \geqslant N_0 \overset{\text{def.}}{=} \frac{2}{\lambda}\left[12B_2^\star \vee \frac{B_1^{\star 2}}{df_\lambda}\right]\log\frac{4}{\delta}\left[1 \vee \frac{4B_2^\star}{\lambda}\right] \implies H_1 + H_3,$$

$$n \geqslant N_{1/2} \overset{\text{def.}}{=} \frac{2}{\lambda}\left[12B_2^\star \vee \frac{B_1^{\star 2}}{df_\lambda} \vee \frac{4B_2^\star}{\lambda}\right]\log^2\frac{4}{\delta}\left[1 \vee \frac{4B_2^\star}{\lambda}\right] \implies H_1 + H_{1b} + H_3,$$

$$n \geqslant N_1 \overset{\text{def.}}{=} \frac{2}{\lambda}\left[12B_2^\star \vee \frac{B_1^{\star 2}}{df_\lambda} \vee \lambda \vee \left(\frac{2B_2^\star(t-1/2)^r}{\lambda^{r-1/2}}\right)^2\right]\log\frac{4}{\delta}\left[1 \vee \frac{4B_2^\star}{\lambda}\right] \implies H_1 + H_2 + H_3,$$

$$n \geqslant \overline{N} \overset{\text{def.}}{=} 8(et)^2\left(4 \vee C^2t^2\right)\frac{df_\lambda}{r_\lambda^\star}\log 2/\delta \implies H_4,$$

$$\lambda \leqslant L \overset{\text{def.}}{=} \left[e^{t+2}T(r,t)(2 \wedge Ct)\right]^{-1/(r+1/2\wedge 1)} \implies H_5.$$

Recall that T *is defined in theorem 2. Moreover, having*

$$\lambda = Kn^{-\frac{\alpha}{1+\alpha(2r+1)}}$$

with K *a constant not depending on* n *make all these conditions possible.*

Proof. The expression of the constant boils down to taking the maximum of each expression. Recall that:

- H_1, H_{1b}, H_2 are defined in theorem 2;
- H_3 is defined in theorem 3;
- H_4 is defined in proposition 4;
- H_5 is defined in proposition 5.

About the fact they are attainable, we need to check that the power of n is smaller than 1, in order that

$$\exists n, \quad n \geqslant N(\lambda) \quad \text{and} \quad \lambda = Kn^{-\frac{\alpha}{1+\alpha(2r+1)}},$$

with $N(\lambda)$ chosen among $\left\{N_0, N_{1/2}, N_1, \overline{N}\right\}$. In the following, $\sim$ denotes equality up to log factors between two quantities. Recall that $s\lambda^{-1/\alpha} \leqslant df_\lambda \leqslant S\lambda^{-1/\alpha}$, and assume

$$\lambda = Kn^{-\frac{\alpha}{1+\alpha(2r+1)}}$$

for some K a positive constant.

- When $r \leqslant 1/2$, $N_0 \sim \lambda^{-(1+1/\alpha)} \sim n^{\frac{1+\alpha}{1+\alpha(2r+1)}}$ and $\frac{1+\alpha}{1+\alpha(2r+1)} < 1$.
- When $1/2 < r \leqslant 1$, $N_{1/2} \sim \lambda^{-2} \sim n^{\frac{2\alpha}{1+\alpha(2r+1)}}$ and $\frac{2\alpha}{1+\alpha(2r+1)} < 1$ as $\alpha(2r+1) > 2\alpha$.
- Finally, when $r > 1$, $N_1 \sim \lambda^{-2r} \sim n^{\frac{\alpha(2r)}{1+\alpha(2r+1)}}$ and $\alpha(2r) < 1 + \alpha(2r+1)$.
- For $\overline{N}$, use the upper bound $\frac{df_\lambda}{r_\lambda^\star} \leqslant SR\lambda^{-(1/\alpha+1/2)}$. Then, $\overline{N} \sim n^{\frac{\alpha+2}{2(1+\alpha(2r+1))}}$ and $\frac{\alpha+2}{2(1+\alpha(2r+1))} = 1 - \frac{\alpha(4r+1)}{\alpha(4r+2)+2} \leqslant 1$.

□

Having bounded the bias and the variance of the estimator, we are now in shape to state our main result.

Theorem 4 (Optimal rates of IT estimator)**.** *Let* $\delta \in (0,1]$, $\lambda > 0$ *and choose* n *so that* H_3 *and the following holds:*

$$\begin{aligned} H_1 \quad & \text{if } r \leqslant 1/2, \\ H_1 + H_{1b} \quad & \text{if } 1/2 < r \leqslant 1, \\ H_1 + H_2 \quad & \text{if } r > 1. \end{aligned}$$

Then we can bound the excess risk with probability greater than $1-\delta$ *as*

$$L(\widehat{\theta}_\lambda^t) - L(\theta^\star) \leqslant C_{\text{bias}} \lambda^{2s} + C_{\text{var}} \frac{df_\lambda}{n}, \quad \text{with } s = (r+1/2) \wedge t.$$

If we further assume that the capacity condition holds and that the estimator does not saturate, that is $t \geqslant r + 1/2$*, then setting*

$$\lambda = \left[\left(\frac{C_{\text{var}}}{C_{\text{bias}}} \right)^2 S \right]^{\frac{\alpha}{1+\alpha(2r+1)}} n^{-\frac{\alpha}{1+\alpha(2r+1)}}$$

makes the following holds with confidence 2δ*:*

$$L(\widehat{\theta}_\lambda^t) - L(\theta^\star) \leqslant 2 \left[\left(\frac{C_{\text{var}}}{C_{\text{bias}}} \right)^2 S \right]^{\frac{\alpha(2r+1)}{1+\alpha(2r+1)}} n^{-\frac{\alpha(2r+1)}{1+\alpha(2r+1)}},$$

where the constants $C_{\text{bias}}, C_{\text{var}}$ *are bounded by quantities only depending on* $r, t, B_2^\star, \delta$ *as soon as hypotheses* H_4 *and* H_5 *are satisfied.*

Proof.

Decomposition of the risk. We use the decomposition of the risk:

$$\begin{aligned} L(\widehat{\theta}_\lambda^t) - L(\theta^\star) &\leqslant \Psi\left(t(\widehat{\theta}_\lambda^t - \theta^\star)\right) \left\| \widehat{\theta}_\lambda^t - \theta^\star \right\|_{\mathbf{H}(\theta^\star)}^2 \\ &\leqslant 2\Psi\left(t(\widehat{\theta}_\lambda^t - \vartheta_\lambda^t) + t(\vartheta_\lambda^t - \theta^\star)\right) \left[\left\| \widehat{\theta}_\lambda^t - \vartheta_\lambda^t \right\|_{\mathbf{H}(\theta^\star)}^2 + \left\| \vartheta_\lambda^t - \theta^\star \right\|_{\mathbf{H}(\theta^\star)}^2 \right], \end{aligned}$$

where we applied proposition 3, and used that $(a+b)^2 \leqslant 2(a^2+b^2)$.

Bias and variance prefactors. We introduce the following quantities:

$$\begin{aligned} C_{\text{bias}}^2 &= 2\Psi\left(t(\widehat{\theta}_\lambda^t - \vartheta_\lambda^t) + t(\vartheta_\lambda^t - \theta^\star)\right) T(r,t) P_\lambda^t, \\ C_{\text{var}}^2 &= 2\Psi\left(t(\widehat{\theta}_\lambda^t - \vartheta_\lambda^t) + t(\vartheta_\lambda^t - \theta^\star)\right) \left[4\sqrt{2} t R_\lambda^t \sqrt{\log 2/\delta} \right], \end{aligned}$$

where P_λ^t, R_λ^t are defined in theorems 2 and 3 respectively. Then the bound on the excess risk reads:

$$L(\widehat{\theta}_\lambda^t) - L(\theta^\star) \leqslant C_{\text{bias}} \begin{cases} \lambda^{2r+1} & \text{if } r + 1/2 \leqslant t \\ \lambda^{2t} & \text{otherwise} \end{cases} + C_{\text{var}} \frac{df_\lambda}{n}$$

with confidence 2δ with the appropriate hypothesis H_1, H_2 or H_3, depending on r, see table 2.

Optimal λ. Further assume $t \geqslant r + 1/2$ and the capacity condition holds with parameters S, α. Then, setting:

$$\lambda^{\frac{1+\alpha(2r+1)}{\alpha}} = \left(\frac{C_{\text{var}}}{C_{\text{bias}}}\right)^2 \frac{S}{n} \iff \lambda = \left[\left(\frac{C_{\text{var}}}{C_{\text{bias}}}\right)^2 S\right]^{\frac{\alpha}{1+\alpha(2r+1)}} n^{-\frac{\alpha}{1+\alpha(2r+1)}},$$

makes the following bound holds with probability $1 - 2\delta$:

$$L(\widehat{\theta}^t_\lambda) - L(\theta^\star) \leqslant 2\left[\left(\frac{C_{\text{var}}}{C_{\text{bias}}}\right)^2 S\right]^{\frac{\alpha(2r+1)}{1+\alpha(2r+1)}} n^{-\frac{\alpha(2r+1)}{1+\alpha(2r+1)}}.$$

Explicit prefactors. Assume Hyp. H_4 and H_5 hold, and $\lambda \leqslant B_2^\star$. Then the quantities $C_{\text{bias}}, C_{\text{var}}$ only depend on r, t up to the term $\Psi\left(t(\widehat{\theta}^t_\lambda - \vartheta^t_\lambda) + t(\vartheta^t_\lambda - \theta^\star)\right)$. Noting that:

$$1 + 1/t \geqslant \underline{\phi}^{-1}(x) \geqslant x \quad \text{implies} \quad 1 + 1/t \geqslant x,$$

and Ψ is increasing we have

$$\Psi\left(t(\widehat{\theta}^t_\lambda - \vartheta^t_\lambda) + t(\vartheta^t_\lambda - \theta^\star)\right) \leqslant \Psi(4) \leqslant 4.$$

In the end $C_{\text{bias}}, C_{\text{var}}$ only depend on r, t and the parameters of the problem:

$$\begin{aligned} C^2_{\text{bias}} &\leqslant 8T(r,t)e^{t+2} \\ C^2_{\text{var}} &\leqslant 32te\sqrt{\log 2/\delta}, \end{aligned} \tag{56}$$

where $T(r,t)$ was introduced previously in theorem 2:

$$T(r,t) = \begin{cases} \|v\|\,(1 \vee (B_2^\star + \lambda))2^r & \text{if } r \leqslant 1, \\ \|v\|\,\frac{w(r)+r}{(t-1/2)^r} & \text{if } r > 1 \text{ and } r + 1/2 < t, \\ \|v\|\,\frac{w(r)}{(t-1/2)^r} + {B_2^\star}^{r-t+1/2} & \text{if } r > 1 \text{ and } r + 1/2 \geqslant t. \end{cases}$$

Proof of theorem 1 in the paper. We took the maximum on the lower bounds on the samples to simplify the result in the main body. Simply define:

$$N = \overline{N} \vee \begin{cases} N_0 & \text{if } r \leqslant 1/2 \\ N_{1/2} & \text{if } 1/2 < r < 1 \\ N_1 & \text{otherwise} \end{cases}$$

$$\text{and} \quad C_{\text{risk}} = \left[\left(\frac{C_{\text{var}}}{C_{\text{bias}}}\right)^2 S\right]^{\frac{\alpha}{1+\alpha(2r+1)}}.$$

Again, we highlight that the observation made in proposition 6 is key to ensure that these constants are attainable, in the sense that they are not in contradiction with the optimal rate in n. □

A.3 Statistical guarantees with inexact solvers

This section is devoted to finding a rule on the tolerance enforced at each step of the proximal sequence. Given a tolerance ϵ, we look for $\overline{\epsilon}_1, \ldots, \overline{\epsilon}_n$, the tolerance to ensure at each proximal step. It leads to proposition 1 in the main body of the article.

Important remark on the notation. So as to simplify the notation, we drop the hat $\widehat{\ }$ on the loss function. That is, we simply take a loss function L assumed to be GSC. In practice, this function is of course the empirical loss $\widehat{L}$. We denote with a bar the quantity we compute at each step, and whose aim is to approximate the estimator of L.

Tikhonov regularization. For a GSC function L, we define:

$$\begin{aligned}
\theta_\mu^1 &= \mathsf{prox}_{L/\mu}(0) = \arg\min_\theta L_\mu(\theta), & L_\mu(\theta) &\overset{\text{def.}}{=} L(\theta) + \frac{\mu}{2}\|\theta\|^2 \\
\theta_\mu^{k+1} &= \mathsf{prox}_{L/\mu}(\theta_\lambda^k) = \arg\min_\theta L_\mu^{\lambda,k}(\theta), & L_\mu^{\lambda,k}(\theta) &\overset{\text{def.}}{=} L(\theta) + \frac{\mu}{2}\left\|\theta - \theta_\lambda^k\right\|^2 \\
\overline{\theta}_\mu^{k+1} &= \mathsf{prox}_{L/\mu}(\overline{\theta}_\lambda^k) = \arg\min_\theta \overline{L}_\mu^{\lambda,k}(\theta), & \overline{L}_\mu^{\lambda,k}(\theta) &\overset{\text{def.}}{=} L(\theta) + \frac{\mu}{2}\left\|\theta - \overline{\theta}_\lambda^k\right\|
\end{aligned}$$

so that we can refer easily to the function which has to be minimized when evaluating the proximal operator.

A.3.1 Definitions

We use the following notations for the *Newton decrement*:

- The theoretical quantity writes:

$$\nu_\mu^{\lambda,k}(\theta) = \left\|\nabla L_\mu^{\lambda,k}(\theta)\right\|_{\mathbf{H}_\lambda^{-1}(\theta)} = \left\|\nabla L(\theta) + \mu(\theta - \theta_\lambda^{\lambda,k})\right\|_{\mathbf{H}_\mu^{-1}(\theta)};$$

- The normalized Newton decrement is defined with:

$$\widetilde{\nu}_\lambda^{k-1}(\theta) = \frac{\nu_\lambda^{k-1}(\theta)}{r_\lambda(\theta)};$$

- The quantity we compute is:

$$\overline{\nu}_\mu^{\lambda,k}(\theta) = \left\|\nabla L_\mu^{\lambda,k}(x)\right\|_{\mathbf{H}_\lambda^{-1}(\theta)} = \left\|\nabla L(\theta) + \mu(\theta - \overline{\theta}_\lambda^{\lambda,k})\right\|_{\mathbf{H}_\mu^{-1}(\theta)}.$$

We also recall some definition and properties. R is defined with

$$R = \sup_{z\in \mathsf{Supp}\,\rho}\ \sup_{g\in\phi(z)} \|g\| \quad \text{so that} \quad \underset{\lambda}{r}(\theta) \geqslant R/\sqrt{\lambda}$$

and $r_\lambda(\theta)$ is given in definition 7. The Dikin ellipsoid, as in [20], reads

$$\forall c \in \mathbb{R}, \quad D_\lambda k - 1(c) = \left\{\theta \in \mathcal{H}; \widetilde{\nu}_\lambda^{k-1}(\theta) \leqslant c\right\}.$$

We provide a short lemma to show how controlling the *normalized* Newton decrement enables to control quantities depending on t.

Lemma 3 (Localization properties with the Newton decrement)**.** *Let* $k \leqslant t$ *and* $c > 0$*. Assume*

$$\overline{\theta}_\lambda^k \in D_\lambda k - 1(c), \quad \textit{that is} \quad \widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) \leqslant c.$$

Then, we have

$$\underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^k - \theta_\lambda^k)\right) \leqslant -\frac{1}{c}\log(1-c) \overset{\text{def.}}{=} \kappa_c. \tag{57}$$

Proof. The proof combines inequalities we already used, replacing the normalized gradient with the Newton decrement. Recall eq. (52), which states that

$$t(\overline{\theta}_\lambda^k - \theta_\lambda^k) \leqslant \frac{\left\|\overline{\theta}_\lambda^k - \theta_\lambda^k\right\|_{\widehat{\mathbf{H}}_\lambda(\overline{\theta}_\lambda^k)}}{r_\lambda(\overline{\theta}_\lambda^k)}. \tag{58}$$

Using the lower bound on gradient of lemma 2 gives

$$\left\|\overline{\theta}_\lambda^k - \theta_\lambda^k\right\|_{\widehat{\mathbf{H}}_\lambda(\overline{\theta}_\lambda^k)} \leqslant \underline{\phi}^{-1}(t(\overline{\theta}_\lambda^k - \theta_\lambda^k))\left\|\nabla L_\lambda^{k-1}(\overline{\theta}_\lambda^k)\right\|_{\mathbf{H}_\lambda^{-1}(\overline{\theta}_\lambda^k)},$$

and using the definition of the Newton decrement in the previous equation gives

$$\left\|\overline{\theta}_\lambda^k - \theta_\lambda^k\right\|_{\widehat{\mathbf{H}}_\lambda(\overline{\theta}_\lambda^k)} \leqslant \underline{\phi}^{-1}(t(\overline{\theta}_\lambda^k - \theta_\lambda^k))\nu_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right). \tag{59}$$

Plugging eq. (59) in eq. (58) implies

$$\underline{\phi}(t(\overline{\theta}_\lambda^k - \theta_\lambda^k))\, t(\overline{\theta}_\lambda^k - \theta_\lambda^k) \leqslant \frac{\nu_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right)}{r_\lambda(\overline{\theta}_\lambda^k)} \overset{\text{def.}}{=} \widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right).$$

Use the fact that $\underline{\phi}(x)x = 1 - e^{-x}$ combined with the definition of the normalized Newton decrement to simplify both sides of the previous equation. After simplification, we obtain

$$t(\overline{\theta}_\lambda^k - \theta_\lambda^k) \leqslant -\log\left(1 - \widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right)\right).$$

Apply $\underline{\phi}^{-1}$ on both side to have

$$\underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^k - \theta_\lambda^k)\right) \leqslant -\widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right)^{-1} \log\left(1 - \widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right)\right),$$

and the conclusion follows with the fact that this is an increasing function of the normalized Newton decrement, which is upper bounded by c. □

This lemma will be useful in the following derivation, and provide some intuition on GSC loss function.

Remark 6. *Intuition for GSC loss function. The purpose of working with Generalized self-concordant loss functions is to be able to control the deviation of the function with their local quadratic approximation. For $\theta \in \mathcal{H}$, lemma 3 gives us that we can bound quantities depending on t in the inequalities of GSC loss functions of proposition 3. When θ is deep into D_λ^{k-1}, then $t \to 1/2$ and the bounds of proposition 3 are tight. On the contrary, when θ leaves this ellipsoid, the upper bound diverges exponentially to infinity while the lower bound goes exponentially to 0, making the deviation from the quadratic approximation very loose.*

To conclude, a GSC function with high R has small Dikin ellipsoids, and is far from its quadratic approximation. On the contrary, a GSC function with low R will be close to its quadratic approximation; the Dikin ellipsoid is large. The extreme case is obtained when ℓ is the square loss. Then, $\phi = \{0\}$, so $R = 0$, and the Dikin ellipsoid spans the whole space for any $\theta \in \mathcal{H}$. This implies e.g. that the lower and upper bound on the gradient matches, making the quadratic approximation tight.

A.3.2 Error propagation

In this section, we give a sufficient condition for achieving an ϵ error on a sequence of proximal operators. Indeed, we aim at minimizing L_λ^{t-1}, but we do not have access to this function; only to its approximation $\overline{L}_\lambda^{t-1}$. Relating both is the purpose of the next result.

Proposition 7 (Error propagation with proximal sequence)**.** *Let $c > 0$. Assume that you can solve each subproblem with precision $\overline{\epsilon}_k$ and that you have a guarantee on the exact normalized decrement:*

$$\forall k \in \{1, \dots, t\}, \quad \begin{cases} \overline{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) \leqslant \overline{\epsilon}_k \\ \overline{\theta}_\lambda^k \in D_\lambda k - 1(c) \iff \widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) \leqslant c \end{cases}$$

Then requiring:

$$\forall k \in \{1, \dots, t\}, \quad \overline{\epsilon}_k = \epsilon \frac{\kappa_c^{k-t}}{t}$$

with $\kappa_c = -1/c \log(1-c)$ suffice to achieve an error ϵ:

$$\nu_\lambda^{t-1}\left(\overline{\theta}_\lambda^t\right) \leqslant \epsilon.$$

We can replace the condition $\overline{\theta}_\lambda^k \in D_\lambda k - 1(c)$ with $\epsilon \leqslant c\sqrt{\lambda}/R$.

Proof. Let us track the error step by step. Denote by ϵ_k the Newton decrement of the exact function at each step:

$$\forall k, \quad \epsilon_k \overset{\text{def.}}{=} \nu_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right).$$

Consider the following decomposition at step k:

$$\begin{aligned}
\nu_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) &= \left\|\nabla L(\overline{\theta}_\lambda^k) + \lambda\left(\overline{\theta}_\lambda^k - \theta_\lambda^{k-1}\right)\right\|_{\mathbf{H}_\lambda^{-1}(\overline{\theta}_\lambda^k)} \\
&\leqslant \left\|\nabla L(\overline{\theta}_\lambda^k) + \lambda\left(\overline{\theta}_\lambda^k - \overline{\theta}_\lambda^{k-1}\right)\right\|_{\mathbf{H}_\lambda^{-1}(\overline{\theta}_\lambda^k)} + \left\|\lambda\left(\overline{\theta}_\lambda^{k-1} - \theta_\lambda^{k-1}\right)\right\|_{\mathbf{H}_\lambda^{-1}(\overline{\theta}_\lambda^k)} \\
&\leqslant \overline{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) + \lambda\left\|\mathbf{H}_\lambda^{-1/2}(\overline{\theta}_\lambda^k)\mathbf{H}_\lambda^{-1/2}(\theta_\lambda^k)\right\|\left\|\overline{\theta}_\lambda^{k-1} - \theta_\lambda^{k-1}\right\|_{\mathbf{H}_\lambda(\overline{\theta}_\lambda^{k-1})} \\
&\leqslant \overline{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) + \underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^{k-1} - \theta_\lambda^{k-1})\right)\nu_\lambda^{k-2}\left(\overline{\theta}_\lambda^{k-1}\right)
\end{aligned}$$

In the last inequality we used that $\left\|\mathbf{H}_\lambda^{-1/2}(\overline{\theta}_\lambda^k)\mathbf{H}_\lambda^{-1/2}(\theta_\lambda^k)\right\| \leqslant 1/\lambda$ and the relation between the distance in Hessian's norm and the Newton decrement of eq. (59). Introducing the notation with epsilon, the last line is by definition

$$\epsilon_k \leqslant \overline{\epsilon}_k + \underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^{k-1} - \theta_\lambda^{k-1})\right)\epsilon_{k-1}. \tag{60}$$

The first term $\overline{\epsilon}_k$ is the error we can control at each step whereas ϵ_{k-1} is the error of interest which increases with k. Using the fact that $\overline{\theta}_\lambda^{k-1} \in D_\lambda^{k-2}(c)$, we have

$$\underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^{k-1} - \theta_\lambda^{k-1})\right) \leqslant -\frac{1}{c}\log(1-c) \overset{\text{def.}}{=} \kappa_c$$

thanks to lemma 3. Thus, eq. (60) becomes

$$\epsilon_k \leqslant \overline{\epsilon}_k + \kappa_c\epsilon_{k-1}. \tag{61}$$

This being valid for all $i \leqslant k$ and since $\overline{\nu}_\lambda^0\left(\overline{\theta}_\lambda^1\right) = \nu_\lambda^0\left(\overline{\theta}_\lambda^1\right)$ we obtain that

$$\epsilon_k \leqslant \sum_{i=1}^{k}\overline{\epsilon}_i\kappa_c^{k-i}. \tag{62}$$

Now plug the assumption of the proposition, namely that each problem is solved with precision

$$\overline{\epsilon}_k = \epsilon\frac{\kappa_c^{k-t}}{t}$$

and use eq. (62) at step t to obtain

$$\epsilon_t \leqslant \sum_{i=1}^{t}\kappa_c^{i-t}\kappa_c^{t-i}\frac{\epsilon}{t} = \epsilon. \tag{63}$$

Replacing $\overline{\theta}_\lambda^k \in D_\lambda k-1(c)$ with $\epsilon \leqslant c\sqrt{\lambda}/R$. Let $k \geqslant 1$. Then, having

$$\overline{\theta}_\lambda^k \in D_\lambda k-1(c)$$

amounts by definition to have

$$\widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) \leqslant c,$$

which is also equivalent to

$$\nu_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) \leqslant c\,\underline{r}_\lambda(\overline{\theta}_\lambda^k).$$

We can use the crude lower bound $r_\lambda(\cdot) \geqslant \sqrt{\lambda}/R$. Thus, the following implication holds:

$$\epsilon_k \overset{\text{def.}}{=} \nu_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right) \leqslant c\frac{\sqrt{\lambda}}{R} \implies \overline{\theta}_\lambda^k \in D_\lambda k-1(c). \tag{64}$$

Now, assume $\epsilon \leqslant c\sqrt{\lambda}/R$. Then, we have that

$$\epsilon_1 = \overline{\epsilon}_1 = \epsilon\frac{\kappa_c^{1-t}}{t} \leqslant c\sqrt{\lambda}/R\frac{\kappa_c^{1-t}}{t},$$

which gives

$$\epsilon_1 \leqslant c\sqrt{\lambda}/R,$$

which implies $\overline{\theta}_\lambda^1 \in D_\lambda^0(c)$ following eq. (64). Then eq. (61) holds with $k = 2$:

$$\epsilon_2 \leqslant \overline{\epsilon}_2 + \kappa_c \epsilon_1.$$

For bigger k, proceed by induction. Let $k < t$ and assume for any $i < k$ that

$$\epsilon_{i+1} \leqslant \overline{\epsilon}_{i+1} + \kappa_c \epsilon_i.$$

Then, we have that

$$\epsilon_k \leqslant \sum_{i=1}^{k} \overline{\epsilon}_i \kappa_c^{k-i}$$

which gives the following bound, thanks to the assumption on ϵ and the $\overline{\epsilon}_i$:

$$\epsilon_k \leqslant \sum_{i=1}^{k} \epsilon \frac{\kappa_c^{i-t}}{t} \kappa_c^{k-i} \leqslant c\sqrt{\lambda}/R.$$

This implies $\overline{\theta}_\lambda^k \in D_\lambda k-1(c)$ following eq. (64), and eq. (61) holds at step $k+1$. Thus the induction hypothesis holds for all k and the conclusion of eq. (63) holds.

Proof of proposition 1. This result is a direct application of the previous one, where we set $c = 1/2$. □

We see that the requirement $\epsilon \leqslant c\sqrt{\lambda}/R$ is simply to ensures that a bound on the Newton decrement $\nu_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right)$ translates to a bound on the *normalized* Newton decrement $\widetilde{\nu}_\lambda^{k-1}\left(\overline{\theta}_\lambda^k\right)$ *via* the crude bound on the Dikin radius $r_\lambda(\overline{\theta}_\lambda^k) \geqslant R/\sqrt{\lambda}$. Thus, the requirement on ϵ can be dropped if we assume $\overline{\theta}_\lambda^k \in D_\lambda k-1(c)$. Such condition is enforced in solver such as the one developed in [20].

Finally, we put in application this result with next proposition, which gives a bound on the excess risk with inexact solver.

Proposition 8 (Bound on the excess risk with inexact solver)**.** *Assume that:*

- *the requirement of proposition 7 hold;*
- *the requirement of theorem 4 hold, namely* H_{1-5}*;*

The first is an hypothesis on the optimization procedure*, while the second in an hypothesis on the* statistics *of the learning task. Then, denoting* $\overline{\theta}_\lambda^t$ *the approximation of* $\widehat{\theta}_\lambda^t$ *as defined in proposition 7, we have the following bound on the excess risk:*

$$L(\overline{\theta}_\lambda^t) - L(\theta^\star) \leqslant C_{\text{bias}}\lambda^{2s} + C_{\text{var}}\frac{df_\lambda}{n} + E_c\,\epsilon, \quad s = (r+1/2) \wedge t,$$

with:

$$E_c \overset{\text{def.}}{=} 4\Psi(4 - \log(1-c))\frac{e^4}{1-c}\kappa_c^2, \quad \textit{e.g.} \quad E_{1/2} \leqslant 4.3 \cdot 10^3.$$

Proof. The proof boils down to combining the statistical results held in theorem 4 with the optimization result of proposition 7. Begin by writing

$$\begin{aligned} L(\widehat{\theta}_\lambda^t) - L(\theta^\star) &\leqslant \Psi\left(t(\overline{\theta}_\lambda^t - \theta^\star)\right)\left\|\overline{\theta}_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)}^2 \\ &\leqslant 2\Psi\left(t(\widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t) + t(\widehat{\theta}_\lambda^t - \theta^\star)\right)\left[\left\|\widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t\right\|_{\mathbf{H}(\theta^\star)}^2 + \left\|\widehat{\theta}_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)}^2\right]. \end{aligned}$$

We know how to handle the statistical term $\left\|\widehat{\theta}_\lambda^t - \theta^\star\right\|_{\mathbf{H}(\theta^\star)}^2$.

Bound on $t(\widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t)$. As in the beginning of the proof of proposition 4, we write:

$$\begin{aligned}
t(\widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t) &\leqslant \frac{1}{r_\lambda(\overline{\theta}_\lambda^t)} \left\| \overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t \right\|_{\mathbf{H}_\lambda(\overline{\theta}_\lambda^t)} \\
&\leqslant \frac{1}{r_\lambda(\overline{\theta}_\lambda^t)} \underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t)\right) \left\| \nabla \widehat{L}_\lambda^{t-1}(\overline{\theta}_\lambda^t) \right\|_{\mathbf{H}_\lambda^{-1}(\overline{\theta}_\lambda^t)} \\
&= \underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t)\right) \widetilde{\nu}_\lambda^{t-1}\left(\overline{\theta}_\lambda^t\right) \\
&\leqslant \underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t)\right) c
\end{aligned}$$

where we used the fact that $\overline{\theta}_\lambda^t \in D_\lambda^{t-1}(c)$, an assumption of proposition 7. With the same reasoning of eq. (53), we conclude:

$$t(\widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t) \leqslant -\log(1-c).$$

Bound on $\left\| \widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t \right\|^2_{\mathbf{H}(\theta^\star)}$. Use a similar reasoning as we used for the variance. Under H_1, we have (Proof of theorem 2, 1st point)

$$\left\| \widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t \right\|^2_{\mathbf{H}(\theta^\star)} \leqslant 2 \left\| \widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t \right\|^2_{\widehat{\mathbf{H}}_\lambda(\theta^\star)}.$$

First write:

$$\left\| \widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t \right\|_{\widehat{\mathbf{H}}_\lambda(\theta^\star)} \leqslant e^{t(\widehat{\theta}_\lambda^t - \theta^\star)/2} e^{t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t)/2} \left\| \widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t \right\|_{\widehat{\mathbf{H}}_\lambda(\overline{\theta}_\lambda^t)},$$

then, for each term, use:

- $t(\widehat{\theta}_\lambda^t - \theta^\star) \leqslant 4$ (end of theorem 4) so that $e^{t(\widehat{\theta}_\lambda^t - \theta^\star)/2} \leqslant e^2$;
- $t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t) \leqslant -\log(1-c)$ so that $e^{t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t)/2} \leqslant (1-c)^{-1/2}$;
- and finally:

$$\begin{aligned}
\left\| \widehat{\theta}_\lambda^t - \overline{\theta}_\lambda^t \right\|_{\widehat{\mathbf{H}}_\lambda(\overline{\theta}_\lambda^t)} &\leqslant \underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t)\right) \left\| \nabla \widehat{L}_\lambda^{t-1}(\overline{\theta}_\lambda^t) \right\|_{\mathbf{H}_\lambda^{-1}(\overline{\theta}_\lambda^t)} \\
&\leqslant \underline{\phi}^{-1}\left(t(\overline{\theta}_\lambda^t - \widehat{\theta}_\lambda^t)\right) \nu_\lambda^{t-1}\left(\overline{\theta}_\lambda^t\right) \\
&\leqslant \left[-\frac{1}{c}\log(1-c) \right] \epsilon \overset{\text{def.}}{=} \kappa_c \epsilon.
\end{aligned}$$

Putting it all together. Thus, using the upper bound on the excess risk, we have with probability greater than $1-2\delta$

$$L(\overline{\theta}_\lambda^t) - L(\theta^\star) \leqslant C_{\text{bias}} \lambda^{2s} + C_{\text{var}} \frac{df_\lambda}{n} + 4\Psi(4 - \log(1-c)) \frac{e^4}{1-c} \kappa_c^2 \epsilon, \quad s = (r+1/2) \wedge t.$$

Taking $c = 1/2$, we have $\Psi(4-\log(1-c)) \leqslant 5$ and $\kappa_c \leqslant 1.4$, which allows bounding the quantity in front of ϵ. □

A.4 Technical lemmas

A.4.1 Concentration of Hermitian operators

In this section, we import results from [21] and [6]. The former provides a bound on $\left\| \widehat{\mathbf{H}}_\lambda^{-1/2}(\theta) \mathbf{H}_\lambda^{1/2}(\theta) \right\|$. The latter provides a bound on $\left\| \widehat{\mathbf{H}}_\lambda^{-1}(\theta) \mathbf{H}_\lambda(\theta) \right\|$, which is more difficult to obtain. They use the

fact that $\mathsf{df}_\lambda = \operatorname{Tr} \mathbf{H}_\lambda(\theta)\mathbf{H}(\theta)$ for least square, but we can't use this very convenient relation here. Thus, we only use their result in the case $1/2 < r < 1$, which makes optimal rate still possible.

We will only use

$$\operatorname{Tr} \mathbf{H}_\lambda^{-1}(\theta)\widehat{\mathbf{H}}(\theta) \leqslant \frac{\mathsf{B}_2(\theta)}{\lambda}.$$

Proposition 9 (Concentration bound)**.** *Let $\delta \in (0,1]$ and $\lambda > 0$. The following holds:*

$$n \geqslant 24\frac{\mathsf{B}_2(\theta)}{\lambda}\log\frac{8\mathsf{B}_2(\theta)}{\lambda\delta} \quad \Longrightarrow \quad \left\|\widehat{\mathbf{H}}_\lambda^{-1/2}(\theta)\mathbf{H}_\lambda^{1/2}(\theta)\right\| \leqslant \sqrt{2}, \tag{65}$$

$$n \geqslant 8\frac{\mathsf{B}_2(\theta)^2}{\lambda^2}\log^2\frac{2}{\delta} \quad \Longrightarrow \quad \left\|\widehat{\mathbf{H}}_\lambda^{-1}(\theta)\mathbf{H}_\lambda(\theta)\right\| \leqslant 2, \tag{66}$$

$$n \geqslant 2\left(1 \vee \frac{4\mathsf{B}_2(\theta)^2}{\lambda^{2s}}\right)\log\frac{2}{\delta} \quad \Longrightarrow \quad \left\|\mathbf{H}(\theta) - \widehat{\mathbf{H}}(\theta)\right\|_{\mathrm{HS}} \leqslant \lambda^s, \tag{67}$$

where each bound hold with confidence $1-\delta$.

Proof. The first equation is Lemma 6 of [21]. The second equation can be adapted from Proposition 5.4 of [6], except that we use

$$\operatorname{Tr} \mathbf{H}_\lambda(\theta)\mathbf{H}(\theta) \leqslant \frac{\mathsf{B}_2(\theta)}{\lambda}$$

instead of df_λ. For the last inequality, use Bernstein inequality for random vectors. With probability $1-\delta$:

$$\left\|\mathbf{H}(\theta) - \widehat{\mathbf{H}}(\theta)\right\|_{\mathrm{HS}} \leqslant \frac{2\mathsf{B}_2(\theta)\log 2/\delta}{n} + \mathsf{B}_2(\theta)\sqrt{\frac{2\log 2/\delta}{n}}.$$

Assuming $n \geqslant 2\log 2/\delta$, this bound becomes

$$\left\|\mathbf{H}(\theta) - \widehat{\mathbf{H}}(\theta)\right\|_{\mathrm{HS}} \leqslant 2\mathsf{B}_2(\theta)\sqrt{\frac{2\log 2/\delta}{n}}.$$

Let $s > 0$. Further requiring $n \geqslant 8\mathsf{B}_2(\theta)\lambda^{-2s}\log 2/\delta$ gives:

$$\left\|\mathbf{H}(\theta) - \widehat{\mathbf{H}}(\theta)\right\|_{\mathrm{HS}} \leqslant \lambda^s$$

which completes the proof. □

A.4.2 Inequalities on Hermitian operators

The following results are given in [6]. We redo the proof to track down and upper bound the constants which are discarded in the original paper.

Lemma 4 (Hermitian operator inequalities)**.** *Let A, B be two non-negative self-adjoint operators on $\mathcal{H}$. Assume $\|A\|, \|B\| \leqslant \kappa$, where $\|\cdot\|$ denotes the operator norm. Then:*

$$\forall r \leqslant 1, \quad \|A^r - B^r\| \leqslant \|A - B\|^r \tag{68}$$

$$\forall r > 1, \quad \|A^r - B^r\| \leqslant w(r)\,\|A - B\| \tag{69}$$

$$\forall r \leqslant 1, \quad \|A^r B^r\| \leqslant \|AB\|^r \tag{70}$$

with $r2^{\lfloor r \rfloor + 1}\kappa^r$.

Proof. For the first point, refer to [5] Theorem X.1.I, Eq. (X.2). For the third point, refer to Theorem IX.2.1 of the same book. It is also known as Cordes inequality [12]. The proofs involve positive semidefinite matrices but are directly applicable to non-negative self-adjoint Hermitian operators.

For the second point, assume $\|A\|, \|B\| \leqslant 1$. Consider the function $f(x) = (1-x)^r$, defined for $|x| \leqslant 1$. Its Taylor expansion reads:

$$f(x) = \sum_{n \geqslant 0} a_n x^n, \quad a_n = \frac{(-1)^n}{n!}\prod_{k=1}^{n}(r-k+1)$$

We have:

$$\left|\frac{a_{n+1}x^{n+1}}{a_n x^n}\right| = \left|\frac{r-n}{n+1}\cdot x\right| \underset{n\to\infty}{\to} |x|$$

so applying d'Alembert's rule, we have that the radius of the series is 1. Now, we have that:

$$A^r - B^r = f(\mathbf{I}-A) - f(\mathbf{I}-B) = \sum_{n\geqslant 0} a_n\left[(\mathbf{I}-A)^n - (\mathbf{I}-B)^n\right]$$
$$\implies \|A^r - B^r\| \leqslant \sum_{n\geqslant 0} |a_n|\,\|(\mathbf{I}-A)^n - (\mathbf{I}-B)^n\|$$

Using that $(\mathbf{I}-A)^n - (\mathbf{I}-B)^n = (\mathbf{I}-A)(\mathbf{I}-A)^{n-1} - (\mathbf{I}-B)^{n-1} - (B-A)(\mathbf{I}-B)^{n-1}$, we obtain:

$$\begin{aligned}\|(\mathbf{I}-A)^n - (\mathbf{I}-B)^n\| &\leqslant \left\|(\mathbf{I}-A)(\mathbf{I}-A)^{n-1} - (\mathbf{I}-B)^{n-1}\right\| + \left\|(B-A)(\mathbf{I}-B)^{n-1}\right\| \\ &\leqslant \left\|(\mathbf{I}-A)^{n-1} - (\mathbf{I}-B)^{n-1}\right\| + 1 \\ &\leqslant n\,\|A-B\|\end{aligned}$$

Denoting $g(x) = (1-x)^{r-1} = \sum b_n x^n$, we have $f'(x) = -rg(x)$ which gives $n\,|a_n| = r\,|b_n|$. Then:

$$\begin{aligned}\|A^r - B^r\| &\leqslant \|A-B\| \sum_{n\geqslant 0} n\,|a_n| \\ &\leqslant r\,\|A-B\| \sum_{n\geqslant 0} |b_n|\end{aligned}$$

We can somewhat painfully upper bound this last term. Notice that for $n > r$, all the b_n have the same sign $s = (-1)^{\lfloor r\rfloor}$. Thus, for $N > r$:

$$\begin{aligned}\sum_{n=0}^{N} |b_n| &= \sum_{n=0}^{\lfloor r\rfloor} |b_n| + s\sum_{n=\lfloor r\rfloor}^{N} b_n \\ &= \sum_{n=0}^{\lfloor r\rfloor} |b_n| + s\lim_{x\to 1}\sum_{n=\lfloor r\rfloor}^{N} b_n x^n \\ &\leqslant 2\sum_{n=0}^{\lfloor r\rfloor} |b_n| + \lim_{x\to 1} g(x) \\ &\leqslant 2\sum_{n=0}^{\lfloor r\rfloor} \frac{1}{n!}\prod_{k=1}^{n}(r-k+1) \\ &\leqslant 2\sum_{n=0}^{\lfloor r\rfloor} \binom{\lfloor r\rfloor}{n} = 2^{\lfloor r\rfloor+1}\end{aligned}$$

Finally, apply these properties to $A/\kappa, B/\kappa$ to obtain in general:

$$\|A^r - B^r\| \leqslant r2^{\lfloor r\rfloor+1}\kappa^r\,\|A-B\|$$

□

A.4.3 Basic calculus

This is a few line of computation, but useful in multiple places.

Lemma 5 (Bound on residual of IT's spectral function)**.** *Let $r, t > 0$. Consider the following function defined on $[0,\kappa]$:*

$$h(\sigma) = \left(\frac{\lambda}{\lambda+\sigma}\right)^t \sigma^r.$$

Then:

$$\sup_{0\leqslant\sigma\leqslant\kappa} h(\sigma) \leqslant \begin{cases}\left(r\cdot\frac{\lambda}{t}\right)^r & \text{if } r < t \\ \left(\frac{\lambda}{\kappa+\lambda}\right)^t \kappa^r & \text{otherwise .}\end{cases}$$

Proof. h is differentiable and

$$h'(\sigma) = \frac{\lambda^t \sigma^{r-1}}{(\sigma+\lambda)^{t+1}} \left[\sigma(r-t) + r\lambda\right].$$

If $t \leqslant r$, the regularization saturates and the maximum is in $\widehat{\sigma} = \kappa$, which gives

$$\sup_\sigma h(\sigma) \leqslant \left(\frac{\lambda}{\kappa+\lambda}\right)^t \kappa^r \underset{\lambda\to 0}{\sim} \lambda^t \kappa^{r-t}.$$

Otherwise, if $t > r$, the maximum is in $\widehat{\sigma} = \frac{r\lambda}{t-r}$ and it reads

$$\begin{aligned} \sup_\sigma h(\sigma) &\leqslant \left(\frac{t-r}{t}\right)^t \left(\frac{r\lambda}{t-r}\right)^r \\ &= \left(\frac{t-r}{t}\right)^{t-r} r^r \left(\frac{\lambda}{t}\right)^r. \end{aligned} \tag{71}$$

We can rewrite the prefactor in front of $(\lambda/t)^r$. First,

$$\left(\frac{t-r}{t}\right)^{t-r} r^r = \left(\frac{t-r}{t}\right)^t \left(\frac{rt}{t-r}\right)^r.$$

Then, use

$$\left(\frac{t-r}{t}\right)^t \leqslant e^{-r} \quad \text{when} \quad r < t. \tag{72}$$

Also,

$$\left(\frac{rt}{e(t-r)}\right)^r = \left(e\left(\frac{1}{r} - \frac{1}{t}\right)\right)^{-r} \leqslant (e/r)^{-r} \leqslant r^r. \tag{73}$$

Use eq. (72) and eq. (73) on the upper bound of eq. (71), and the result is obtained. □

A.5 Experiments

A.5.1 Technical details

Splines. The spline kernel of order q is defined on $[0,1]^2$ as

$$\Lambda_q(x,z) = \sum_{k\in\mathbb{Z}} \frac{e^{2i\pi k(x-z)}}{|k|^q}.$$

A closed form expression is available when q is an even integer:

$$\Lambda_q(x,z) = 1 + \frac{(-1)^{q/2-1}}{q!} B_q(|x-z|).$$

B_q are Bernoulli polynomial of order q. They can be implemented easily. We also have the relation

$$\langle \Lambda_q(x,\cdot), \Lambda_{q'}(x',\cdot)\rangle_{L_2(\mathcal{X},\rho_x)} = \Lambda_{q+q'}(x,x')$$

Our choice of r, α reflects the constraints on α and $(r+1/2)\alpha + 1/2$ to be even integers.

Regularization. For both least square and logistic regression, the regularization λ is chosen among 50 log spaced values between 10^{-4} and 1.

Resources. Computation was carried by a `Intel(R) Xeon(R) CPU E5-1620 v2 @ 3.70GHz`, with 32GB of RAM.

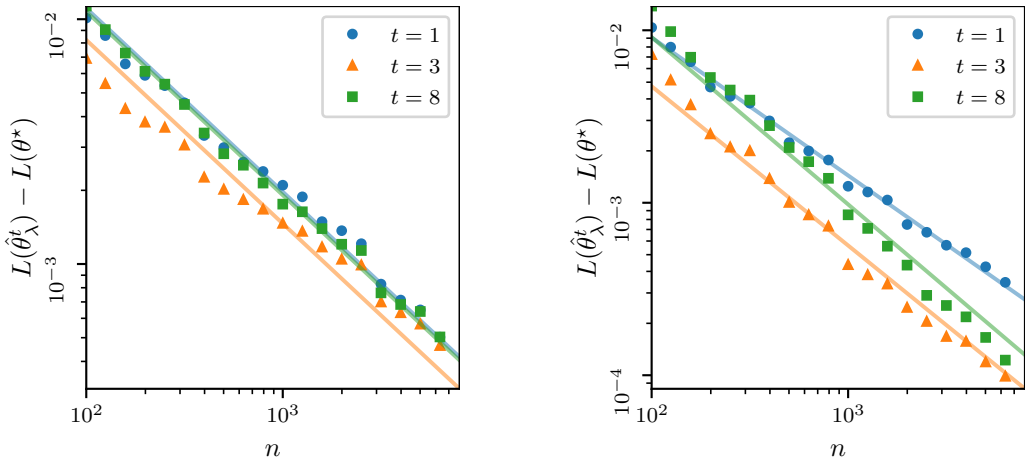


Figure 2: Excess risk with **least square** for various Iterated Tikhonov estimator, function of n. **Colors**: $t = 1$ (Tikhonov) estimator is shown in orange; $t = 2, 3$ in green, red. **Left**: from a difficult problem, $r = 1/4, \alpha = 2$. **Right**: easy problem, $r = 41/4, \alpha = 2$. Plain lines are predicted by theory, with slope $-\alpha(1+2s)/1+\alpha(1+2s)$, $s = \min\{r, t - 1/2\}$ (see main text). All plots are averaged over 100 different initialization.

A.5.2 Simulations with least square

Estimating $\widehat{\theta}^t_\lambda$. We leverage the very convenient filter interpretation with least square. We diagonalize the kernel matrix $K = UDU^\top$ once, then evaluate the estimator with

$$\widehat{\theta}^t_\lambda = \sum_{i=1}^n \alpha_i \phi(x_i),$$
$$\alpha = \frac{1}{n} U g^t_\lambda(D/n) D^\top y,$$

where g^t_λ is IT's filter, defined in (8).

Simulations. The simulations are reported in figs. 2 and 3. The same broad conclusion as for the classification task with the logistic loss apply. Surprisingly, IT(8) seems to suffer from higher constant than its counterpart with low t.

A.5.3 Synthetic binary task

Derivation of the noise. We have $\theta^\star(x) = \Lambda_{(r+1/2)\alpha+\epsilon}(x, 0)$ a function of smoothness $r + 1/2$ in $L_2(\mathcal{X}, \rho_x)$. We want to use logistic regression. Thus, we need to choose the noise $\rho(y \mid x)$ so that

$$\theta^\star(x) = \arg\min_z \int_{\mathcal{Y}} \ell(y, z) d\rho_{y|x}(y).$$

To keep things simple, we restrict the output space to $\mathcal{Y} = \{-1, 1\}$. Denote $a(x) = \mathbb{P}(y = 1 \mid x)$. We will have $\mathbb{P}(y = -1 \mid x) = 1 - a(x)$. Now we need to choose a s.t

$$a \in [0, 1] \quad \text{and} \quad \theta^\star(x) = \arg\min_z h(z) \stackrel{\text{def.}}{=} \log(1 + e^z)(1 - a) + \log(1 + e^{-z})a.$$

Having $a > 0, 1 - a > 0$ implies that h has a unique minimizer z^*. Then

$$h'(z) = \frac{1}{1 + e^z}\left((1 - a)e^z - a\right) \implies a = \frac{e^{z^*}}{1 + e^{z^*}}.$$

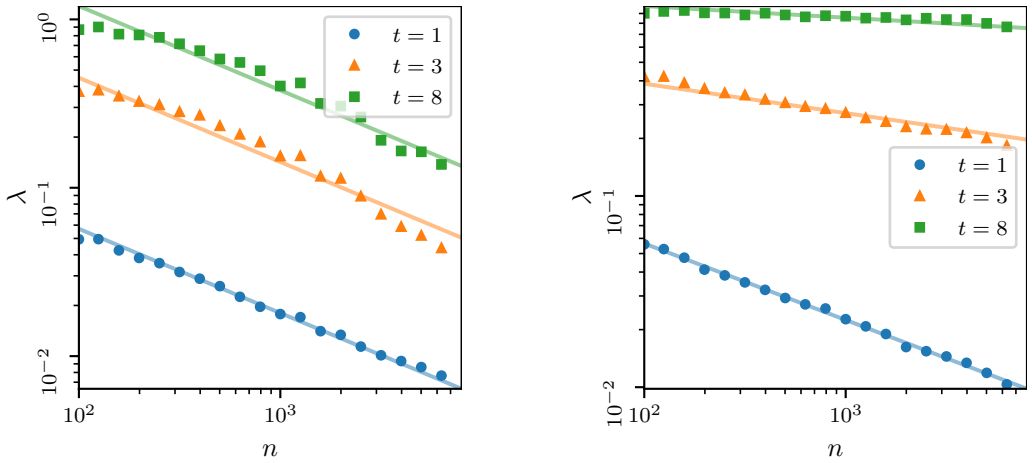


Figure 3: Chosen regularization λ with **least square** for various Iterated Tikhonov estimator, function of n. **Colors**: $t = 1$ (Tikhonov) estimator is shown in orange; $t = 3, 8$ in green, red. **Left**: from a difficult problem, $r = 1/4$, $\alpha = 2$. **Right**: easy problem, $r = 41/4$, $\alpha = 2$. Plain lines are predicted by theory, with slope $-\alpha/1+\alpha(1+2s)$, $s = \min\{r, t - 1/2\}$ (see main text). All plots are averaged over 100 different initialization.

Having required that $\theta^\star(x) = \arg\min_z h(z) \overset{\text{def.}}{=} z^*$, we can use the following output distribution:

$$\begin{aligned} \mathcal{Y} &= \{-1, 1\} \\ \mathbb{P}(y = 1 \mid x) &= \frac{1}{1 + e^{-\theta^\star(x)}} \\ \mathbb{P}(y = -1 \mid x) &= \frac{1}{1 + e^{+\theta^\star(x)}} \end{aligned}$$

which, in turn, ensures that $a(x) \in [0, 1]$.

Newton or first-order methods. In practice, the proximal operator is evaluated with a Newton method, or we use the toolbox Cyanure for big n [19]. Both are used with tolerance 10^{-10}, that is machine precision for single precision. Generally speaking, first-order methods are considered more performant than Newton methods. However, both practical and theoretical considerations motivate the use of second-order scheme in our statement of proposition 1. Firstly, preconditionated iterative solver such as the one used in [20] provide very efficient results for ill-conditioned problems. Secondly, the analysis of GSC loss functions is well-suited to second-order scheme, as the Newton decrement is a natural quantity to keep track of the optimization error. Measuring the error differently would require additional assumption on the loss function.

Estimating the excess risk. The excess risk is estimated with Monte Carlo sampling, with 10^4 points:

$$\mathrm{ER}(\theta) - \mathrm{ER}(\theta^\star) \approx \frac{1}{n_{\mathrm{MC}}} \sum_{i=1}^{n_{\mathrm{MC}}} \frac{1}{1 + e^{-\theta^\star(x_i)}} \log\left(\frac{1 + e^{-\theta(x_i)}}{1 + e^{-\theta^\star(x_i)}}\right) + \frac{1}{1 + e^{\theta^\star(x_i)}} \log\left(\frac{1 + e^{\theta(x_i)}}{1 + e^{\theta^\star(x_i)}}\right)$$

Additional results. We report here the regularization λ chosen function of n and t for various IT regularized estimators. We confirm that the penalty used for IT is larger than of Tikhonov, to compensate for the fitting induced by the additional proximal steps. We also compare the excess risk achieved by IT with the excess risk of Tikhonov, and observe consistent improvement for easy task with a sufficiently high number of samples.

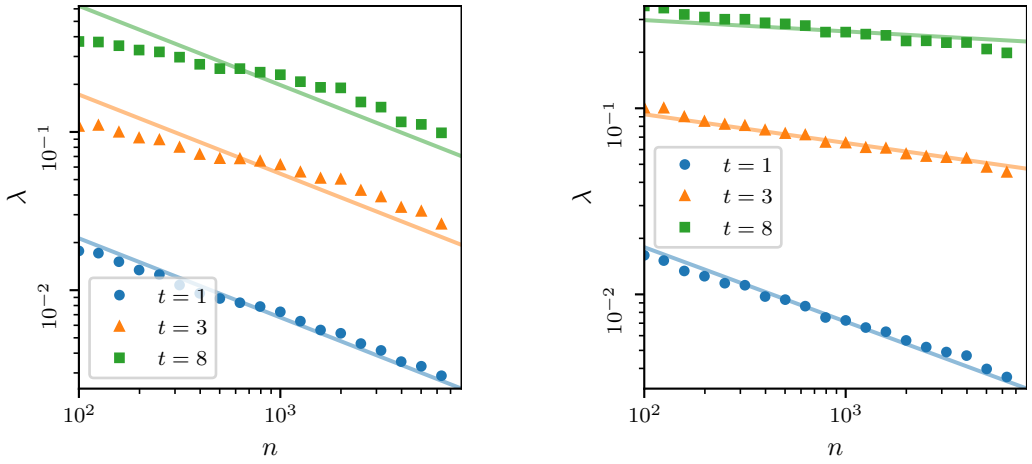


Figure 4: Chosen regularization λ for various Iterated Tikhonov estimator, function of n. **Colors**: $t = 1$ (Tikhonov) estimator is shown in orange; $t = 3, 8$ in green, red. **Left**: from a difficult problem, $r = 1/4$, $\alpha = 2$. **Right**: easy problem, $r = 41/4$, $\alpha = 2$. Plain lines are predicted by theory, with slope $-\alpha/1+\alpha(1+2s)$, $s = \min\{r, t - 1/2\}$ (see main text). All plots are averaged over 100 different initialization.

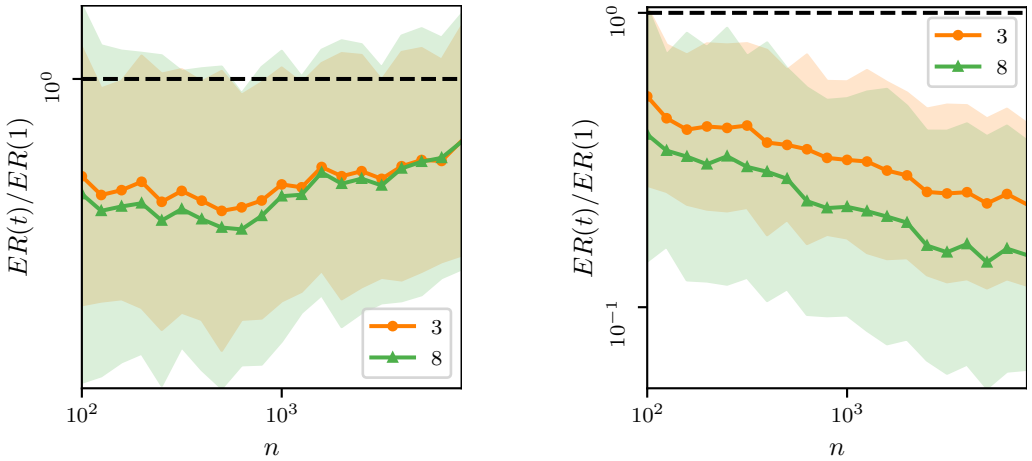


Figure 5: Ratio of IT's excess risk over Tikhonov's excess risk, function of n. **Left**: from a difficult problem, $r = 1/4$, $\alpha = 2$. **Right**: easy problem, $r = 41/4$, $\alpha = 2$. Whereas we expect the ratio to be consistently lower than 1, IT performs worse than Tikhonov in isolated cases, probably due to the optimization process and the chosen regularization path. Yet, it provides lower excess risk than Tikhonov overall, with up to an order of magnitude of improvement with as few as 1000 samples. All plots are averaged over 100 different initialization.

References for part I

[1] Y. Averyanov and A. Celisse. Early stopping and polynomial smoothing in regression with reproducing kernels. *arXiv preprint arXiv:2007.06827*, 2020.

[2] F. Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384 – 414, 2010. doi: 10.1214/09-EJS521. URL https://doi.org/10.1214/09-EJS521.

[3] F. Bauer, S. Pereverzev, and L. Rosasco. On regularization algorithms in learning theory. *Journal of complexity*, 23(1):52–72, 2007.

[4] Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. Beyond Tikhonov: Faster Learning with Self-Concordant Losses via Iterative Regularization. In *Advances in Neural Information Processing Systems*, October 2021.

[5] R. Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[6] G. Blanchard and N. Mücke. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18, 2016. doi: 10.1007/s10208-017-9359-7.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[8] A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007. doi: 10.1007/s10208-006-0196-8.

[9] Yair Carmon, Arun Jambulapati, Qijia Jiang, Yujia Jin, Yin Tat Lee, Aaron Sidford, and Kevin Tian. Acceleration with a ball optimization oracle. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/dba4c1a117472f6aca95211285d0587e-Abstract.html.

[10] C. Ciliberto, L. Rosasco, and A. Rudi. A general framework for consistent structured prediction with implicit loss embeddings. *Journal of Machine Learning Research (JMLR)*, 21 (98):1–67, 2020. URL http://jmlr.org/papers/v21/20-097.html.

[11] A. Dieuleveut, N. Flammarion, and F. Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *Journal of Machine Learning Research (JMLR)*, 18(101):1–51, 2017. URL http://jmlr.org/papers/v18/16-335.html.

[12] J. Fujii, M. Fujii, T. Furuta, and R. Nakamoto. Norm inequalities equivalent to heinz inequality. In *Proceedings of the American Mathematical Society*, 1993.

[13] L. Lo Gerfo, L. Rosasco, F. Odone, E. De Vito, and A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008. doi: 10.1162/neco.2008.05-07-517. URL https://app.dimensions.ai/details/publication/pub.1045202542andhttp://www.dima.unige.it/~devito/pub_files/spectral_finale.pdf.

[14] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer, 2006.

[15] J. Thomas King and Chillingworth D. Approximation of generalized inverses by iterated regularization. *Numerical Functional Analysis and Optimization*, 1(5):499–513, 1979. doi: 10.1080/01630567908816031. URL https://doi.org/10.1080/01630567908816031.

[16] A. Kulunchakov and J. Mairal. Estimate sequences for stochastic composite optimization: Variance reduction, acceleration, and robustness to noise. *Journal of Machine Learning Research (JMLR)*, 21(155):1–52, 2020. URL http://jmlr.org/papers/v21/19-073.html.

[17] H. Lin, J. Mairal, and Z. Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research (JMLR)*, 18(1):7854–7907, 2018.

[18] S. Lin, Y. Lei, and D. Zhou. Boosted kernel ridge regression: Optimal learning rates and early stopping. *Journal of Machine Learning Research (JMLR)*, 20(46):1–36, 2019. URL http://jmlr.org/papers/v20/18-063.html.

[19] J. Mairal. Cyanure: An open-source toolbox for empirical risk minimization for Python, C++, and soon more, 2019.

[20] U. Marteau-Ferey, F. Bach, and A. Rudi. Globally convergent newton methods for ill-conditioned generalized self-concordant losses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[21] U. Marteau-Ferey, D. Ostrovskii, F. Bach, and A. Rudi. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. In *Conference on Learning Theory (COLT)*, 2019. URL http://proceedings.mlr.press/v99/marteau-ferey19a.html.

[22] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[23] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[24] A. Rudi and L. Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[25] A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[26] B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426, 2001.

[27] B. Schölkopf and A. Smola. Support vector machines and kernel algorithms. *Encyclopedia of Biostatistics*, 04 2002.

[28] Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: A recipe for newton-type methods, 2018.

[29] E. De Vito, L. Rosasco, A. Caponnetto, U. De Giovannini, and F. Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research (JMLR)*, 6(30):883–904, 2005. URL http://jmlr.org/papers/v6/devito05a.html.

[30] G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990.

[31] Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007. ISSN 1432-0940. doi: 10.1007/s00365-006-0663-2. URL https://doi.org/10.1007/s00365-006-0663-2.

[32] Tong Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49(3):682–691, 2003. doi: 10.1109/TIT.2002.808136.

Part II
Influence of the Learning Rate on Generalization in Neural Networks

Section 5 in this part is based on our second article [6],

> Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. On the Benefits of Large Learning Rates for Kernel Methods. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 254–282. PMLR, June 2022.

As highlighted in the introduction, the ultimate goal of machine learning is less about perfecting the optimization of the empirical loss, but to develop an estimator that exhibits strong generalization capabilities for unseen data. While considerable efforts are focused on creating algorithms that rapidly converge to the optimum, an intriguing question arises: Is it possible to design an algorithm that deliberately optimizes more slowly, yet enhances its ability to generalize? This concept is explored in section 5, where we demonstrate that employing larger learning rates in gradient descent can, albeit counterintuitive to optimization, lead to an estimator with improved generalization.

One effective strategy to facilitate this is by steering the empirical risk's optimum towards a subset of estimators with particular traits related to good generalization. Similar to how weight decay nudges the optimization outcome towards solutions with lower norms in a specific functional space, some efforts was put into identifying characteristics that positively impact generalization. A key attribute believed to contribute to this are flat minima, as suggested in various studies [11, 12, 13, 16, 26]. The premise is that flat minima, akin to large-margin classifiers, enhance generalization due to the associated margin. Notably, it has been observed that flat minima correlate with larger initial step sizes in gradient descent training of neural networks [19]. Additionally, some optimizers focus on minimizing a loss function regularized with a term linked to the Hessian's trace, a method analyzed using tools similar to ours in [4].

5 On the Benefits of Large Learning Rates for Kernel Methods

Abstract

This section studies an intriguing phenomenon related to the good generalization performance of estimators obtained by using large learning rates within gradient descent algorithms. First observed in the deep learning literature, we show that such a phenomenon can be precisely characterized in the context of kernel methods, even though the resulting optimization problem is convex. Specifically, we consider the minimization of a quadratic objective in a separable Hilbert space, and show that with early stopping, the choice of learning rate influences the spectral decomposition of the obtained solution on the Hessian's eigenvectors. This extends an intuition described by Nakkiran [23] on a two-dimensional toy problem to realistic learning scenarios such as kernel ridge regression. While large learning rates may be proven beneficial as soon as there is a mismatch between the train and test objectives, we further explain why it already occurs in classification tasks without assuming any particular mismatch between train and test data distributions.

5.1 Introduction

Gradient descent methods are omnipresent in machine learning, and a lot of effort has been devoted to better understand their theoretical properties. Optimal rates of convergence have been well characterized for minimizing convex functions in various contexts, including, for instance, stochastic optimization [24]. For supervised learning, one is however more interested in the statistical optimality of the resulting estimator rather than in the ability to quickly optimize a training objective [8]. When considering both optimization and statistical questions, gradient descent methods were proven to be optimal under many assumptions [34, 27].

An important observation for this section is that gradient descent algorithms typically require to tune some learning rate, or step size, to achieve the best performance. This has been thoroughly investigated in the optimization literature. For convex smooth problems in particular, the influence of step size on convergence rates is well understood [25]. However, recent empirical studies have highlighted a surprising aspect of this parameter: when using gradient descent methods on neural networks, *large* learning rates were found to be useful for obtaining *good generalization properties*, or in other words, good statistical performance [20], even though they may be sub-optimal from an optimization point of view.

This section aims at understanding this phenomenon from a broad but simple perspective, where both the function F we optimize and the function R used to evaluate the statistical performance are quadratic forms of some separable Hilbert space $\mathcal{H}$. Specifically, we assume that

$$\forall \theta \in \mathcal{H}, \;\; F(\theta) = \frac{1}{2}\|\theta - \theta^\star\|_T^2 + \mathsf{cst}, \quad \text{and} \quad R(\theta) = \frac{1}{2}\|\theta - \nu^\star\|_U^2, \tag{1}$$

where $\|\cdot\|_A$ denotes the norm on $\mathcal{H}$ induced by a positive definite operator A, *i.e.*, $\|\theta\|_A^2 = \langle \theta, A\theta \rangle$ for any θ in $\mathcal{H}$. With eq. (1), F and R are characterized by positive definite operators T and U, along with their minimizers denoted by $\theta^\star$ and $\nu^\star$, respectively. The constant value cst does not affect the optimization problem and can be safely ignored in the rest of this presentation. The model from eq. (1) captures a large class of problems such as learning with kernels, detailed in section 5.4, but we give here a simple example with ridge regression.

Example 1 (T and U with Ridge Regression.)**.** *Let $x_1, \dots, x_n$ be data points in $\mathbb{R}^d$, and $y_1, \dots, y_n$ prediction variables, with $n \geqslant d$. Define $X \in \mathbb{R}^{n \times d}$ the data matrix. We consider the ridge regression estimator with regularization $\lambda > 0$, which is defined as the minimum of*

$$\forall \theta \in \mathbb{R}^d, \;\; F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(\theta^\top x_i - y_i)^2 + \frac{\lambda}{2}\|\theta\|^2 = \frac{1}{2n}\|X\theta - y\|^2 + \frac{\lambda}{2}\|\theta\|^2 .$$

F *is a quadratic function of* θ*, which can be rewritten as in eq.* (1) *with*

$$\forall \theta \in \mathbb{R}^d,\ F(\theta) = \frac{1}{2}\|\theta - \nu^\star\|_T^2 + \mathsf{cst},\ \textit{with}\ \begin{cases} \nu^\star & = \frac{1}{n}\left(\frac{1}{n}X^\top X + \lambda \mathbf{I}_d\right)^{-1} X^\top y \\ T & = \frac{1}{n}(X^\top X + \lambda \mathbf{I}_d). \end{cases}$$

Assuming that the output can be written $y_i = x_i^\top \nu^\star + \epsilon_i$ *with* $\nu^\star \in \mathbb{R}^d$ *and* ϵ_i *some independent, zero-mean noise, then the* population loss *is* $\mathcal{P}(\theta) = \mathbb{E}\, 1/2\left(\theta^\top x_i - y_i\right)^2$*, and the* excess risk *defined by* $R(\theta) = \mathcal{P}(\theta) - \inf_\nu \mathcal{P}(\nu)$ *is given with*

$$R(\theta) = \mathbb{E}\left[\frac{1}{2}\left((\theta - \nu^\star)^\top x\right)^2\right] = \frac{1}{2}\|\theta - \nu^\star\|_U^2,\ \textit{with}\ U = \mathbb{E}\left[xx^\top\right].$$

In this example, a discrepancy between train and test losses (between T and U) may occur in particular situations (*e.g.*, presence of data augmentation during training, or simply mismatch between train and test distributions). The next example shows that such a mismatch may be in fact frequent for classification problems, even when train and test distributions do match.

Example 2 (Discrepancy between train and test losses in classification with separable classes.)**.** *The scenario described in eq.* (1) *is particularly evident in the context of binary classification, when the classes are separable by a non-zero margin. This is considered a typical situation in many learning scenarios of interest, as classification over natural images—motivating the wide use of large-margin based classifiers in the field [29]. We highlight here, that in this context, the loss we are using for training is not the best loss to consider for the test error, as discussed next. More precisely, consider a classification problem with two classes with non-zero margin. Let* $\mathcal{X} \subseteq \mathbb{R}^d$ *and* $\mathcal{Y} = \{-1, 1\}$ *be the input and the output space. Denote by* $\rho(x, y) = \rho_{\mathcal{X}}(x)\rho(y|x)$ *the probability distribution describing the classification problem, where* $\rho_{\mathcal{X}}$ *is the marginal probability over* $\mathcal{X}$*, while* $\rho(y|x)$ *is the conditional probability of* y *given* x*. The error that we would like to minimize is the binary error on the population, i.e.* $B(\theta) = \mathbb{P}[\mathrm{Sign}[\theta(x)] \neq y]$ *for a model* θ*. Let* $\nu^\star$ *be a function minimizing the binary error and* $\mathcal{H}$ *be the class of models under consideration. Assume, for simplicity, that* $\mathcal{H}$ *is a RKHS with norm* $\|\cdot\|$ *[1],* i.e.*, there exists a map* $\phi : \mathcal{X} \to \mathcal{H}$ *such that the function in* $\mathcal{H}$ *are characterized as* $\theta(x) = \langle \phi(x), \theta\rangle$*, for any* θ *in* $\mathcal{H}$*. It has been shown by Pillaud-Vivien et al. [27] (in particular, Lemma 1 and Appendix A, Theorem 13), that in the context of two classes separated by a non-zero margin and whose conditional probability is regular enough, then* $\nu^\star$ *is in* $\mathcal{H}$ *and moreover* $B(\theta) - B(\nu^\star) \leqslant e^{-c/\|\theta - \nu^\star\|}$ *for some constant* c*. Therefore, the binary error decreases exponentially in terms of the Hilbert norm* $\|\cdot\|$*. On the other hand, the norm minimized at training time is some smooth convex surrogate of the binary loss, as, for example, the quadratic loss.*

In this case, the trained vector may be obtained by minimizing the population loss (in fact, a regularized empirical version, but we omit this fact here for simplicity) such that $F(\theta) = \mathbb{E}(\theta(x) - y)^2$*. Noting that* $\int(\theta(x) - \theta^\star(x))^2 d\rho(x) = \|T^{1/2}(\theta - \theta^\star)\|^2 = \|\theta - \theta^\star\|_T$ *for* $T = \int \phi(x)\phi(x)^\top d\rho_{\mathcal{X}}(x)$*, and* $\theta^\star = \nu^\star$ *under the considered conditions (see the same paper), we have*

$$F(\theta) - F(\theta^\star) = \int(\theta(x) - \theta^\star(x))^2 d\rho_{\mathcal{X}}(x) = \|\theta - \theta^\star\|_T^2.$$

This is a typical case, where there is a discrepancy between the error of interest

$$R(\theta) - R(\nu^\star) = \|\theta - \nu^\star\|^2,$$

which, if optimized, would lead to an exponential decrease of the classification error, and the loss that instead is the one approximately optimized by the algorithm at training time, i.e. F*, for which we have only the slower rate* $B(\theta) - B(\nu^\star) \leqslant (F(\theta) - F(\nu^\star))^\alpha$*, with* $\alpha \in (1/2, 1)$ *is known (see, e.g. Audibert [2] or Audibert and Tsybakov [3] for what concerns the CAR assumption).*

In this paper, we are interested in understanding in which regime large learning rates with early stopping could be useful for kernel methods, even if they are suboptimal from an optimization point of view. We consider indeed the optimization of F in eq. (1) with plain gradient descent, starting from a vector θ_0 in $\mathcal{H}$ with step-size η, and we distinguish between two cases: having a *small* learning rate η_s or a *large* learning rate η_b, the range of both is to be detailed later. A simple intuition was suggested by Nakkiran [23] on a two-dimensional toy problem, showing that large learning rates may be beneficial as soon as there is a mismatch

between F and R (meaning, what we train on does not correspond to what we test on). We show that such an insight can be extended beyond toy problems to realistic scenarios with traditional kernel methods, and that, perhaps surprisingly, this phenomenon occurs already in simple classification tasks.

Theorem 1 (Informal version of our main result)**.** *Under a few assumptions described later in this paper, consider the target accuracy α and large and small step sizes θ_b and θ_s (these quantities being defined in the aforementioned assumptions). Consider the gradient descent iterations $\theta_{t+1} = \theta_t - \eta T(\theta_t - \theta^\star)$ either with step size $\eta = \eta_b$ or $\eta = \eta_s$, and stop the procedure as soon as $F(\theta_{t+1}) \leqslant \alpha$, resulting in two estimators θ_b or θ_s. Then,*

$$R(\theta_b) - R(\nu^\star) \;\leqslant\; 34 \frac{\kappa_U}{\kappa_T} (R(\theta_s) - R(\nu^\star)), \tag{2}$$

where κ_U and κ_T are the condition numbers of the operators U and T, respectively, restricted to $\mathcal{H}_n$.

Note that $R(\nu^\star) = 0$ by definition of R; we made this quantity explicit in the bound for clarity purposes. The main conclusion from the theorem is that with early stopping (and with a target accuracy that is reasonable according to statistical learning theory, as discussed later), large learning rates can provide better estimators than small ones, even though the quantity η_s may yield much faster convergence for minimizing the objective function F than η_b (a fact also discussed later). This phenomenon occurs when the condition number κ_T is much larger than κ_U, which may already arise in classification tasks, as mentioned earlier in Example 2 where $\kappa_U = 1$.

Note that eq. (2) raises several questions and could be easily misinterpreted, since such a relation may suggest that an arbitrarily small risk $R(\theta_b)$ could be obtained by considering minimization problems that are arbitrarily badly conditioned. Unfortunately, but not surprisingly, there is however no free lunch here, as discussed in the next remark.

Remark 1 (The issue of ill-conditioning.)**.** *A naive observation is that $R(\theta_b)$ could be arbitrarily small by making the problem more ill-conditioned. However, the bound on $R(\theta_b)$ in eq. (2) is relative to $R(\theta_s)$. Notably, a careful reading of the proof shows that $R(\theta_s)$ is an increasing function of the conditioning number, for the chosen level sets α.*

Summary of contributions. Our first contribution is the relation described in eq. (2), highlighting potential benefits of large learning rate strategies when the training objective has a worse condition number than the one used to evaluate the quality of the estimator. This is illustrated in fig. 1, a figure inspired by Nakkiran [23]. Our second contribution is to show that such a mismatch systematically occurs in simple classification scenarios with low noise, where the quantity of interest to minimize may not be the population risk, as discussed earlier. Overall, this allows us to show that the previous phenomenon occurs in realistic learning scenarios with kernels, which we also check in practice through numerical experiments.

5.2 Related Work

Our main motivation is to better understand the role of learning rate in obtaining good generalization for supervised learning. Even though empirical benefits of large learning rates were first described for neural networks, a few recent works have studied this phenomenon for convex problems. We review here some relevant work.

Setting the learning rate in neural networks. Stochastic gradient descent has become the standard tool for optimizing neural networks. When the learning rate is very small, the network evolves in a so-called "lazy-regime" where its dynamics are well understood [9, 17] but which fails to capture the good generalization performance observed with large learning rate. Specifically, this phenomenon has been empirically observed numerous times [see, for instance 30, 20, 18]; common strategies consist of using first a large learning rate, before annealing it to a smaller value. As a first step towards proving theoretically the effect of choosing large learning rates for training neural networks, Li et al. [21] devise a two-layer neural network model with different set of features where the order in which they are learnt matters, where the previous annealing strategy could be shown to be useful in theory.

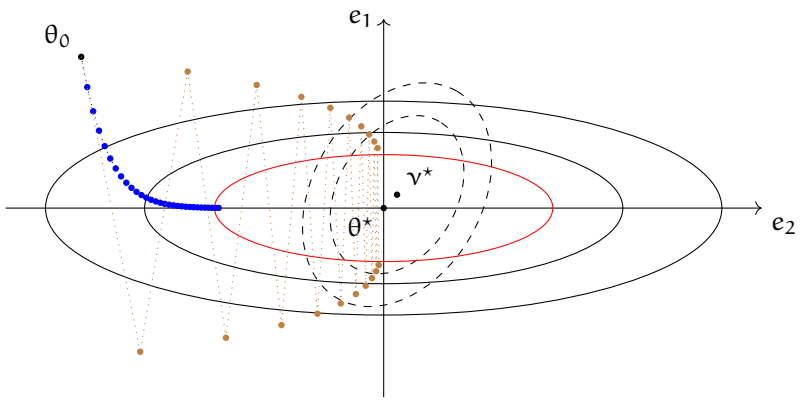


Figure 1: We optimize the quadratic F *(level sets are filled lines, centered in* $\theta^\star$*)* with gradient descent, starting from θ_0 until we reach the level sets α *(filled line, red)*. However, we evaluate the quality of the estimator through R *(level sets are dashed lines, centered in* $v^\star$*)*. Doing small step size *(blue dots)* optimizes the direction e_1 first, and yields an estimate which is far from $v^\star$ in U norm; doing big step size *(brown dots)* oscillates in the direction e_1, but ultimately yields an estimator which is close to $v^\star$ in U norm.

A convex perspective. Recently, different papers tried to reproduce this phenomenon in convex settings. This is probably thanks to the observation made by [23], where a toy dataset is exhibited, which was the main motivation for this work. However, it fails to capture realistic scenarios where the data distribution is not isotropic, or with non linear data embeddings. Wu et al. [32] aimed at filling these gaps but again, relies on the data distribution to be linear, isotropic, with the number of dimension going to infinity in order to have all data points approximately orthogonal; we do not make any of those assumptions. Finally, we highlight that we use *plain* gradient descent, and do not need stochasticity to exhibit the big learning rate phenomenon. This is consistent with recent work [14] which shows that SGD is not necessary to obtain state of the art performances, and that GD simply needs a better fine tuning of hyper parameters.

5.3 Main Result

In this section, we show that by performing standard gradient descent on the empirical loss F, choosing a big learning rate will first optimize the smallest eigencomponent of T. That is, the resulting estimator is mostly located on the biggest eigenvector of T. On the other hand, the smaller the learning rate, the more will the solution be located on the small eigencomponents, with biggest eigenvectors of T being learnt first.

5.3.1 Settings and notations

Gradient descent updates. We perform standard gradient descent on the empirical loss F, starting from some $\theta_0 \in \mathcal{H}$, with step size η. We obtain

$$\forall t \geqslant 0, \quad \theta_{t+1} = \theta_t - \eta T(\theta_t - \theta^\star), \quad \text{thus} \quad \theta_t - \theta^\star = (\mathbf{I} - \eta T)^t(\theta_0 - \theta^\star). \tag{3}$$

This enables a very simple analysis of the training in the eigenbasis of T. We now give a more precise definition of the model in eq. (1).

Assumption 1 (Representer theorem assumption)**.** *There is a* n*-dimensional subspace* $\mathcal{H}_n \subseteq \mathcal{H}$ *that is invariant by* T*—that is,* $T\theta$ *is in* $\mathcal{H}_n$ *for all* θ *in* $\mathcal{H}_n$ *and such that* $v^\star$ *is in* $\mathcal{H}_n$*.*

We denote by (σ_i, e_i) *the eigenbasis of the p.d. operator* T *restricted to* $\mathcal{H}_n$*, with* $\sigma_1 > \cdots > \sigma_n > 0$*, assuming eigenvalues are distinct from each other, and we call* $\kappa_T = \sigma_1/\sigma_n$ *the condition number. Similarly, the restriction of* U *to* $\mathcal{H}_n$ *is a positive definite operator whose spectrum is* $\sigma_1 > \cdots > \sigma_n$*, with condition number* $\kappa_U = \sigma_1/\sigma_n$*. Since, rescaling the objectives* F *and* R *by constant factors does not change their minimizers, we also safely assume that* $\sigma_1 = \sigma_1 = 1$*.*

The model described by assumption 1 is quite natural, and ensures that a representer theorem holds when learning on a finite training set of n points. It is notably satisfied in classical learning formulations with kernels.

With the notations of assumption 1, we can now rewrite the update of eq. (3) along a specific direction e_i:

$$\forall t \geqslant 0, \quad \langle \theta_t - \theta^\star, e_i \rangle_{\mathcal{H}} = (1 - \eta\sigma_i)^t \langle \theta_0 - \theta^\star, e_i \rangle_{\mathcal{H}} . \tag{4}$$

Consider the quantity $|1 - \eta\sigma_i|$. The closer to 0, the smaller will the i-th component of $\theta_t - \theta^\star$ on the eigenbasis be when the number of steps t increases. We plot $|1 - \eta\sigma_i|$ in fig. 2.

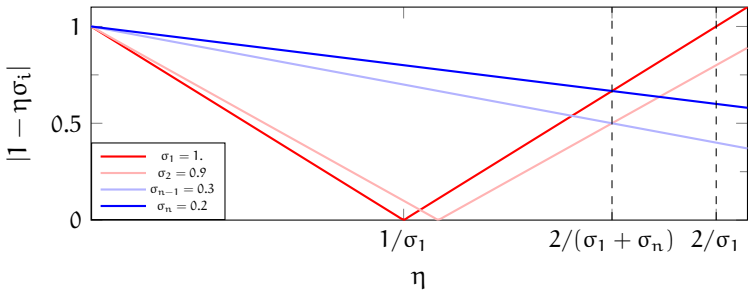


Figure 2: Attenuation coefficient $|1 - \eta\sigma_i|$ function of the step size η, for 4 eigenvalues. For $1 \leqslant i \leqslant n$, the attenuation is the quantity by which decays the projection of $\theta - \theta^\star$ on e_i at each step. The closer to 0, the faster will the direction e_i of F be learnt. In our analysis, the learning rate must satisfy $\eta_s < 2/(\sigma_1 + \sigma_n) < \eta_b < 2/\sigma_1$.

Specifically, two ranges of learning rate naturally appear. With a *small* learning rate satisfying $\eta_s < 2/(\sigma_1 + \sigma_n)$, we see on fig. 2 that the attenuation $|1 - \eta\sigma_i|$ is biggest for the smallest eigenvalue. On the other hand, with a *big* learning rate satisfying $\eta_b > 2/(\sigma_1 + \sigma_n)$, we see that the attenuation is biggest for the biggest eigenvalue. This motivates the next assumption.

Assumption 2 (Learning rate)**.** *The learning rates satisfy*

$$0 < \eta_s < \frac{2}{\sigma_1 + \sigma_n} < \eta_b < \frac{2}{\sigma_1}. \tag{5}$$

Note that the quantity $\eta = \frac{2}{\sigma_1 + \sigma_n}$ which naturally appears in fig. 2 is a classical upper bound for proving the convergence of σ_1-smooth and σ_n-strongly convex function, of which F belongs to, see *e.g.* Thm 2.1.15 in [25]. The rate $1/\sigma_1$ is the classical one when we do not have a strong convexity assumption. This means that in our model, the concept of "small" learning rate simply means being of the order of the best possible learning rates available from an optimization point of view, while the concept of "large" means being close to values leading to diverging algorithms.

Remark 2 (Biggest learning rate before divergence.)**.** *In a recent work, Cohen et al. [10] observed that neural networks trained with gradient descent and "good" constant step size η were often in a regime where σ_1 – the maximum value of the Hessian of the loss – hovered just around $2/\eta$. It is surprisingly analogous to our model: the range of learning rate we consider for big step sizes in Assumption 2 enforces $2/\eta_b$ to be close to σ_1.*

Remark 3 (Oscillating weights.)**.** *Geometrically, having $\eta > 1/\sigma_1$ means that the estimator will oscillate along the direction e_1, i.e. $\langle \theta_t - \theta^\star, e_1 \rangle$ will change sign at each iteration. Such behavior was observed for neural networks trained with classical learning rate strategies, where the weights' sign change in the early phase of training [33].*

Then, the following technical assumption is needed to ensure that there is a signal on the lowest and biggest eigendirection. It is satisfied *e.g.* as soon as the initialization is chosen at random.

Assumption 3 (Initialization)**.** *We assume*

$$\langle \theta_0 - \theta^\star, e_1 \rangle_{\mathcal{H}} \neq 0, \quad \langle \theta_0 - \theta^\star, e_n \rangle_{\mathcal{H}} \neq 0.$$

Finally, we assume that the target accuracy in terms of optimization is not too small compared to the model error $R(\nu^\star)$:

Assumption 4 (Target accuracy and model error)**.** *Consider some learning rates η_b and η_s chosen in the range of Assumption 2. We assume that the target accuracy α satisfies $\alpha \leqslant \alpha_1$, where α_1 is given in Definition 3 and only depends on the spectrum of T and the learning rates. Furthermore, we assume that*

$$\frac{R(\theta^\star)}{\alpha} \leqslant \min\left\{\frac{1}{4}, \frac{\kappa_T}{72\kappa_U}\right\}. \tag{6}$$

This assumption is twofold, providing both an *upper* bound and a *lower* bound on α. The *upper bound* stems from the proof technique of our main result in Theorem 2, a brief sketch of which is available in section 5.3.4. It relies on the fact that sufficiently many steps t_s, t_b are made before the gradient descent is stopped. To ensure this, we can either make the learning rate smaller, or take the target error α sufficiently small. We choose the latter, hence the assumption $\alpha \leqslant \alpha_1$. The *lower bound* in eq. (6) simply ensures that the model error $R(\theta^\star)$ is small respectively to the target accuracy α. In other words, we want $\theta^\star$ to be a good proxy for $v^\star$ on the level sets α. Indeed, if $R(\theta^\star) \neq 0$ in the limit case where $\alpha \to 0$ we have $\theta_s, \theta_b \to \theta^\star$: both estimator suffer the same loss w.r.t R, and we cannot hope having $R(\theta_b)$ smaller than $R(\theta_s)$.

5.3.2 The Main Theorem

We now give our main theorem, whose proof is given in section B.1.

Theorem 2 (Benefits of large learning rates)**.** *Consider the different quantities defined in Assumptions 1, 2, 3 and 4. Then, perform the gradient descent updates of eq.* (3)*, with either small step size η_s or big step size η_b, and stop as soon as $F(\theta_t) \leqslant \alpha$, assuming that the resulting estimators satisfy $F(\theta_s) \geqslant \alpha/2$ and $F(\theta_b) \geqslant \alpha/2$.*[1] *Then,*

$$R(\theta_b) - R(v^\star) \;\leqslant\; 34\frac{\kappa_U}{\kappa_T}(R(\theta_s) - R(v^\star)). \tag{7}$$

Recall that R has minimum 0, so eq. (7) essentially guarantees better performance of θ_b. The estimator obtained with big step size is better that the one obtained with small step size as soon as the operator T is ill-conditioned. This is notably the case when doing classification with kernel methods with the ridge estimator, as we discuss in section 5.4.

5.3.3 Discussion

Implications in classification with separable classes. In the context of the example discussed in the introduction, we see clearly that, under the simplifying hypothesis that the population error behaves similarly to the empirical error, the choice of the learning step has the unexpected impact of reducing the Hilbert norm by a multiplicative constant that can be significantly smaller than 1, leading to an exponential improvement in the classification error.

Comparison with analysis techniques based on learning rate annealing. Most recent approaches to explain the role of the learning rate in the generalization [21, 32] rely on *annealing* the learning rate: the first phase of the training is carried out with a large step size, before it is discounted to a lower value. We do not need such mechanism in our theoretical analysis, which turns to be simpler with a unique value for the step size. With annealing, our analysis could sum up the following way: do t steps with learning rate greater or equal to $2/\sigma_1$, so that all attenuation coefficient $|1 - \eta_b\sigma_i|$ in eq. (4) are smaller than 1 except for the first (few) eigencomponent. Doing so, all eigendirections would be optimized except for the first (few) ones. Then, anneal the learning rate until the α level set of the loss are reached.

Discussion on the complexity. The fact that the result of Theorem 2 relies on the condition number can be somewhat surprising. Indeed, we may wonder why the other eigenvalues of the spectrum do not play a role in the result. This is in fact due to to proof technique, which relies on comparing the estimator mostly located on e_1 (for big learning rates) and the one mostly on e_n (for the small learning rates). Thus, the distance $\sigma_{n-1} - \sigma_n$ and $\sigma_1 - \sigma_2$ play a role in the *complexity* of the gradient descent, which is highlighted by next lemma, which is a consequence of Lemmas 2 and 3 in section B.1.

[1] This is a mild assumption that could be removed at the price of cumbersome technical details.

Lemma 1 (Computational complexity)**.** *Under the settings of Theorem 2, denote* t_s *(resp.* t_b*) the number of steps necessary to obtain* θ_s *(resp.* θ_b*). Then,*

$$t_s = O\left(\log \frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}\right), \quad t_b = O\left(\log \frac{|\eta_b\sigma_1 - 1|}{\max\{\eta_b\sigma_2 - 1, 1-\eta_b\sigma_n\}}\right). \tag{8}$$

In particular, if $s = \sigma_{n-1} - \sigma_n$, then $t_s = O(1/s)$, and the same holds for t_b with $s = \sigma_1 - \sigma_2$. However, we emphasize that Lemma 1 is of purely theoretical interest. In practice, the benefits of big learning rate is observed as soon as the solution is located on the top k eigenvectors (and not *only* on the first one). This scenario could be covered at the price of more involved proof techniques.

5.3.4 Sketch of proof for the main result

The detailed proof is delayed to section B.1. The idea is the following: by tuning the number of steps, we can have the estimator trained with small (resp. big) step size mostly aligned with the smallest (resp. biggest) eigenvector.

The directional bias induced by the step size. As shown in fig. 2, having the learning rates satisfying Assumption 2 ensures that the quantities

$$\epsilon_b^2 = \frac{\sum_{2\leqslant i\leqslant n}\langle\theta_b - \theta^\star, e_i\rangle^2}{\langle\theta_b - \theta^\star, e_1\rangle^2}, \quad \epsilon_s^2 = \frac{\sum_{1\leqslant i\leqslant n-1}\langle\theta_s - \theta^\star, e_i\rangle^2}{\langle\theta_s - \theta^\star, e_n\rangle^2}, \tag{9}$$

can be made arbitrarily small, while Assumption 3 ensures they are well defined. ϵ_b (resp. ϵ_s) quantifies to what extent is $\theta_b - \theta^\star$ (resp. $\theta_s - \theta^\star$) mostly on e_1 (resp. e_n). For instance, in the extreme case where $\epsilon_b = 0$ (resp. $\epsilon_s = 0$), then $\theta_b - \theta^\star = xe_1, x \in \mathbb{R}$ (resp. $\theta_s - \theta^\star = ye_n, y \in \mathbb{R}$). To see why it can be made small, refer to the closed-form expression of $\theta_{(\eta,t)}$ in eqs. (3) and (4), and assume for simplification that $\langle\theta_0 - \theta^\star, e_i\rangle = c_0$ for all i, with c_0 some constant factor. This gives

$$\epsilon_b^2 = \frac{\sum_{2\leqslant i\leqslant n}(1-\eta_b\sigma_i)^{2t}}{(1-\eta_b\sigma_1)^{2t}}, \quad \epsilon_s^2 = \frac{\sum_{1\leqslant i\leqslant n-1}(1-\eta_s\sigma_i)^{2t}}{(1-\eta_s\sigma_n)^{2t}}. \tag{10}$$

Following the discussion of Assumption 2, we have that $|1-\eta_b\sigma_1| > |1-\eta_b\sigma_i|$ for all $i > 1$, and $|1-\eta_s\sigma_n| > |1-\eta_s\sigma_n|$ for all $i < n$. Thus, we have that for any $\delta > 0$,

$$t_b \geqslant \frac{1}{2}\frac{\log 1/\delta^2}{\log \frac{\eta_b\sigma_1 - 1}{\max_{2\leqslant i\leqslant n}|1-\eta_b\sigma_i|}} \implies \epsilon_b^2 \leqslant \delta^2, \quad t_s \geqslant \frac{1}{2}\frac{\log 1/\delta^2}{\log \frac{1-\eta_s\sigma_n}{\max_{1\leqslant i\leqslant n-1}|1-\eta_s\sigma_i|}} \implies \epsilon_s^2 \leqslant \delta^2. \tag{11}$$

Different risk on the α-level sets. In this paragraph, assume *(A)* $\theta_b = xe_1$, and $\theta_s = ye_n$, that is $\epsilon_b = \epsilon_s = 0$, *(B)* that $R(\theta^\star) = 0$ and *(C)* that

$$\alpha/2 \leqslant F(\theta_b) \leqslant \alpha, \quad \alpha/2 \leqslant F(\theta_s) \leqslant \alpha. \tag{12}$$

Then, we have for θ_b that

$$R(\theta_b) = \frac{1}{2}\|xe_1\|_U^2 \leqslant \frac{1}{2}\sigma_1 x^2 \leqslant \frac{\alpha}{2}\frac{\sigma_1}{\sigma_1}, \tag{13}$$

where we used eq. (12) to bound $\alpha \geqslant F(\theta_b) = 1/2 \cdot \sigma_1 x^2$. We do the same with θ_s, this time using $1/2 \cdot \sigma_n y^2 = F(\theta_s) \leqslant \alpha$ to obtain

$$R(\theta_s) = \frac{1}{2}\|ye_n\|_U^2 \geqslant \frac{1}{2}\sigma_n y^2 \geqslant \alpha\frac{\sigma_n}{\sigma_n}, \tag{14}$$

Finally, combining eq. (13) with eq. (14), we obtain

$$R(\theta_b) \leqslant 2\frac{\kappa_U}{\kappa_T}R(\theta_s).$$

Ensuring both conditions can be met together. We now point out the main differences between this simplified sketch of proof and the rigorous proof in section B.1. First of all, we do not have *(A)* but rather an approximation of it, with $\epsilon_s \leqslant \delta$ and $\epsilon_b \leqslant \delta$. Second, we do not have *(B)* and rather take into account the error $R(\theta^\star)$ to derive Theorem 2. Finally, and most importantly, we check that we can have *low* ϵ *and* $F(\theta) \geqslant \alpha/2$ *at the same time*. Indeed, we need a big number of iterations to achieve low ϵ_b or ϵ_s. This implies better optimization of the objective function F. To prevent this, we can either tune the learning rate (having η_s close to 0 and η_b close to $2/\sigma_1$) or provide an upper bound on α. We choose the later, hence the hypothesis $\alpha \leqslant \alpha_1$ in Assumption 4.

5.4 Comparison with Results in Kernel Regression

To provide some intuition over the result of Theorem 2, we consider its implications in a supervised learning setting, specifically classification on a low-noise dataset.

5.4.1 Background on the kernel ridge regression estimator

We consider standard settings. We denote by $\mathcal{X} \subseteq \mathbb{R}^d$ the input space, and $\mathcal{Y} = \{-1, 1\}$ the output space. We draw n i.i.d samples $(x_i, y_i)_{1 \leqslant i \leqslant n}$ from an unknown distribution ρ on $\mathcal{X} \times \mathcal{Y}$, and we search a prediction function $\theta \in \mathcal{H}$, where $\mathcal{H}$ is a RKHS with feature map $\phi : \mathcal{X} \to \mathcal{H}$. We assume the kernel to be bounded by a constant C_K. See [1] for a precise account on RKHS. We use the square loss as loss function. In order to find a function θ which maps elements of $\mathcal{X}$ to $\mathcal{Y}$, we optimize the (regularized) *empirical risk* F, defined for all $\theta \in \mathcal{H}$ and $\lambda \geqslant 0$ a regularization parameter with

$$F(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (\theta(x_i) - y_i)^2 + \frac{\lambda}{2} \|\theta\|^2 . \tag{15}$$

The minimizer $\theta^\star$ of F is always well defined as soon the training samples x_i are distinct (in the case $\lambda = 0$), which we assume now. We will be minimizing F with gradient descent when we are in fact interested in minimizing the *test error*

$$B(\theta) = \mathbb{P}[\mathrm{Sign}[\theta(x) \neq y]]. \tag{16}$$

We will relate the test error and the empirical risk to quadratics forms in $\mathcal{H}$ by means of other quantities. To do that, we first define the population loss along with its regularized version with

$$\forall \theta, \;\; \mathcal{P}(\theta) = \int_{\mathcal{X} \times \mathcal{Y}} \frac{1}{2} (\theta(x) - y)^2 \, d\rho(x, y), \;\; \mathcal{P}_\lambda(\theta) = \mathcal{P}(\theta) + \frac{\lambda}{2} \|\theta\|^2 . \tag{17}$$

The minimizer of $\mathcal{P}$ on $\mathcal{L}_2(\rho_x)$ is the regression function $g^\star(x) = \mathbb{E}[y|x]$. It is an element of $\mathcal{L}_2(\rho_x)$ but not necessary of $\mathcal{H}$. We denote by $\nu^\star$ the minimizer of $\mathcal{P}_\lambda$ on $\mathcal{H}$. If $\lambda > 0$, it is always well defined; otherwise, with $\mathcal{I} : \mathcal{H} \to \mathcal{L}_2(\rho_x)$ the inclusion operator, $\nu^\star$ exists as soon as the projection of the regression function on the closure of the range of $\mathcal{I}$ belongs to the range of $\mathcal{I}$. See [31] for a precise account.

In the following, we assume $\lambda \geqslant 0$ and that $\nu^\star$ is well defined, and we consider specific assumptions on ρ, *via* assumptions on $\nu^\star$ and $g^\star$.

5.4.2 Relating supervised learning with quadratic forms in $\mathcal{H}$

To relate the problems of eqs. (15) and (16) to quadratic forms in the RKHS, we simply need to introduce the *empirical covariance operator*, with

$$T = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \otimes \phi(x_i). \tag{18}$$

Then, as optimizing the Hilbert norm is a good proxy for optimizing the test error (following Example 2 and Lemma 4 in section B.2), we define

$$F(\theta) = \frac{1}{2} \|\theta - \theta^\star\|_T^2 + \mathrm{cst}, \;\; R(\theta) = \frac{1}{2} \|\theta - \nu^\star\|_{\mathcal{H}}^2 , \;\; \text{with } \mathrm{cst} = \frac{1}{2n} y^\top \left[\mathbf{I}_n - \frac{K}{n} \left(\frac{K}{n} + \lambda \right)^{-1} \right] y. \tag{19}$$

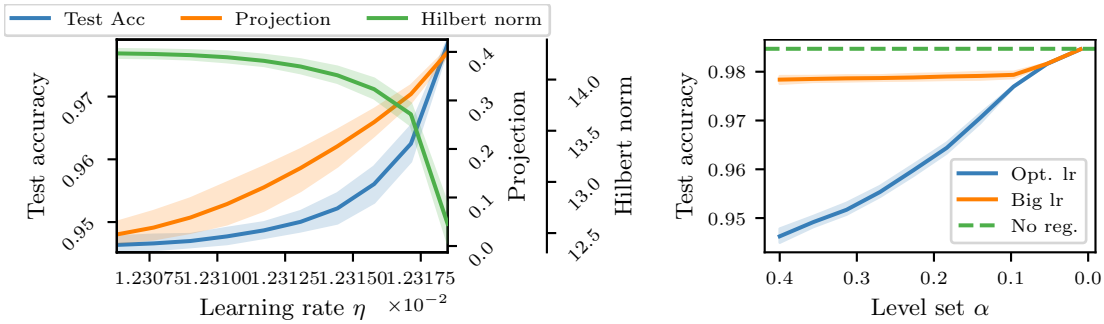


Figure 3: *(Left)* Test accuracy *(blue)* function of step size η for CKN-MNIST. As the learning rate increases, the projection on the first component *(orange)* increases, which makes the Hilbert norm *(green)* decreases. This results in predictions closest in $\mathcal{L}_\infty$ norm to the optimum.
(Right) Test accuracy function of level set α for CKN-MNIST. As we optimize more, the better performances of big step size *(orange)* compare to optimal step size *(blue)* vanish to reach the prediction of the optimum of F *(green, dashed)*.
Shaded areas show standard deviation (train set and initialization) over 10 runs.

K is the kernel matrix (K/n shares the same spectrum than T), and the minimum $F(\theta^\star) = \mathsf{cst}$ is 0 when $\lambda = 0$. We are in the settings of the model of eq. (1), and we can readily apply Theorem 2 with U being the identity operator $\mathbf{I}_{\mathcal{H}}$.

Corollary 1 (Benefit of big step size for classification task.)**.** *Under the settings of Theorem 2 with the additional assumption that* $U = \mathbf{I}_{\mathcal{H}_n}$*, we have*

$$R(\theta_b) - R(\nu^\star) \leqslant \frac{34}{\kappa_T}(R(\theta_s) - R(\nu^\star)).$$

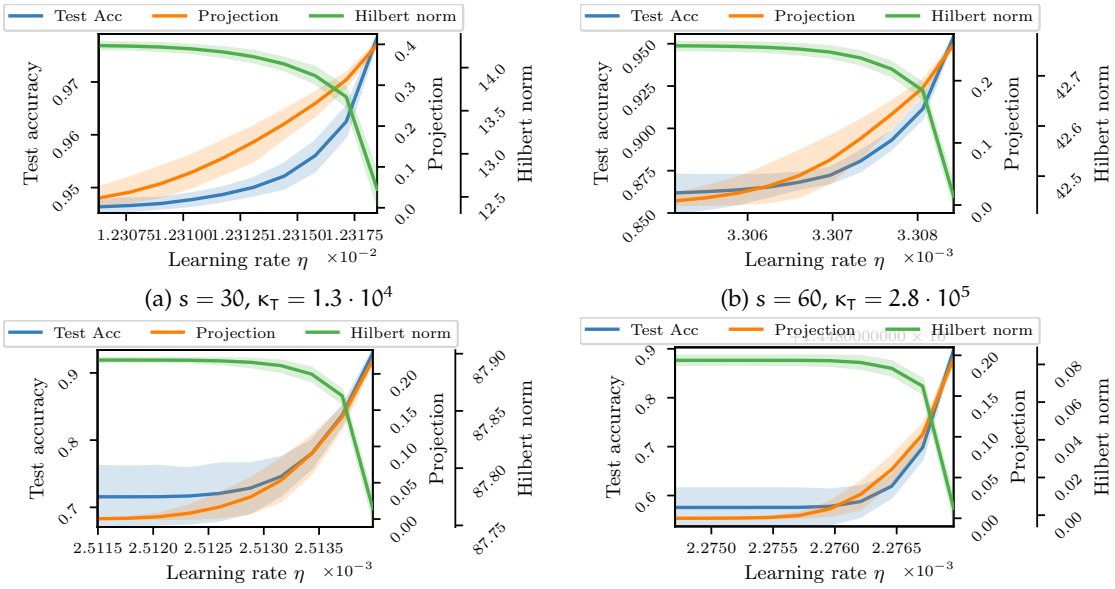
5.5 Experiments

We evaluate the claims of section 5.4 on CKN-MNIST, a dataset consisting of the MNIST dataset embedded by a convolutional kernel network [22]. It allows for a realistic use-case, with classification accuracy close to 99%, by necessitating a reasonable number of samples $n = 1000$. On CKN-MNIST, we achieve 98.5% test accuracy with the Gaussian kernel with scale parameter 30 and no regularization. Adding regularization only improves the test accuracy by 0.04%.

Test accuracy function of the step size. We define some final level set α. We plot three quantities function of the learning rate η. The *projection on the first component* $\langle \theta - \theta^\star, e_1 \rangle_{\mathcal{H}}$ must be small for moderate learning rate and big for learning rate close to $2/\sigma_1$, as the attenuation satisfies $|1 - \eta\sigma_1| \to 1$ when $\eta \to 2/\sigma_1$, see fig. 2. This makes the *Hilbert norm* $R(\theta(\eta))$ decreases as η increases, as predicted by Corollary 1. This results in better test accuracy $1 - B$ for big learning rate, following Lemma 4. This is summed up in fig. 3, left.

Test accuracy function of optimization. Our main bound in Theorem 2 relies on Assumption 4: in learning settings, it means that the optimization error α must be greater than a constant times the statistical error $R(\theta^\star)$ in order to observe improvements with big step sizes. This is shown in fig. 3, right. Additional details on the plot is available in section B.5.

Scale of the kernel. In fig. 3, the scale of the Gaussian kernel is set to $s = 30$, with which we obtained the best results on the test set. It is worth noting though that the scale of the kernel directly impacts the conditioning of the matrix. Notably, when $s \to 0$, the kernel matrix K tends to the identity (hence $\kappa_T \to 1$) while when $s \to \infty$, K tends to a rank-1 operator (hence $\kappa_T \to \infty$). A core result of Theorem 2 is that we have *bigger improvement for bigger* κ_T. We reproduce the experiment on the test accuracy with different scale in fig. 4, and notice indeed bigger improvements for larger scale s, hence worse conditioning.



(a) $s = 30$, $\kappa_T = 1.3 \cdot 10^4$

(b) $s = 60$, $\kappa_T = 2.8 \cdot 10^5$

(c) $s = 90$, $\kappa_T = 1.4 \cdot 10^6$

(d) $s = 120$, $\kappa_T = 4.0 \cdot 10^7$

Figure 4: Theorem 2 predicts that the improvement in taking big rather than small step size increases with the condition number of T. We test this claim on our dataset: larger kernel scale s makes the condition number of the kernel matrix κ_T increases, which results in a larger margin between the test accuracy of θ_b and the one of θ_s.

5.6 Conclusion

A large class of learning problems can be formulated as optimizing a function F with gradient descent while we are interested in optimizing another function R. Using simple quadratic forms to model this mismatch is a natural thing to do, while already providing lot of insight. We indeed show that the choice of large step sizes that may be suboptimal from an optimization point of view may provide better estimators than small/medium step sizes. In particular, we show that this phenomenon occurs in realistic classification tasks with low noise when learning with kernel methods. In future work, we are planning to study other variants of gradient-based algorithms, which may be stochastic, or accelerated, and perhaps exploit the insight developed in our work to design new algorithms, which would focus on statistical efficiency, exploiting prior knowledge on the test loss, rather than on optimization of the training objective.

B Appendix of On the Benefits of Large Learning Rates for Kernel Methods

Overview of the appendix

- In section B.1, we prove Thm. 2.
- In section B.2, we give additional technical information on the low-noise classification task.
- We compare the bound in Theorem 2 to existing results in the case of regression task with kernels, in section B.3. We highlight the difference between gradient descent with big step size and the estimators which write as a spectral filter.
- Section B.4 makes a simple remark, highlighting the difference between gradient descent on the train loss and gradient descent on the Hilbert norm in kernel regression.
- Finally, section B.5 gives additional information on the experiments.

B.1 Proof of main result

B.1.1 Definition and assumptions

Recall from the main text Assumptions 1, 2, 3, 4.

We can rewrite precisely the gradient descent update in eq. (3) with the following definition.

Definition 1 (Notations for the estimator)**.** *For some step size η, a number of steps t and some $\theta_0 \in \mathcal{H}_n$, we denote $\theta^{(\eta,t)}$ the estimator obtained through gradient descent from θ_0. We denote $(\mu_i^{(\eta,t)})_{1\leqslant i\leqslant n}$ the decomposition of $\theta^{(\eta,t)} - \theta^\star$ on e_i,* i.e.

$$\theta^{(\eta,t)} - \theta^\star = \sum_{i=1}^{n} \mu_i^{(\eta,t)} e_i. \tag{20}$$

Denoting the initialization with $(\iota_i)_{1\leqslant i\leqslant n}$,

$$\theta_0 - \theta^\star = \sum_{i=1}^{n} \iota_i e_i, \tag{21}$$

we have that

$$\forall i \in \{1,\dots,n\},\ \ \mu_i^{(\eta,t)} = \iota_i A_i^{(\eta,t)} = \iota_i (1-\eta\sigma_i)^t, \tag{22}$$

where A_i is the attenuation of the i-th eigencomponent at each step.

To lighten the notations, we denote $\theta_s = \theta^{(\eta_s,t_s)}$ the estimator obtained with t_s small *step size η_s, and $\theta_b = \theta^{(\eta_b,t_b)}$ the estimator obtained with t_b* big *step size η_b. Likewise, we use μ_i when it is clear from the context which of θ_s or θ_b we study.*

With these notations and Assumption 2 and 3, note that

$$\forall(\eta,t),\ \ \mu_1^{(\eta_b,t)} \neq 0,\ \ \mu_n^{(\eta_s,t)} \neq 0.$$

Also, we can now introduce the *second biggest attenuation coefficients* in the following definition.

Definition 2 (Second biggest attenuation coefficient.)**.** *We introduce the second biggest attenuation coefficients $\overline{A}_b, \overline{A}_s$ with*

$$\overline{A}_b \overset{\text{def.}}{=} \max\left\{A_2^{(\eta_b)}, A_n^{\eta_b}\right\},\ \ \overline{A}_s \overset{\text{def.}}{=} A_{n-1}^{(\eta_s)}. \tag{23}$$

Referring to fig. 2, this implies

- For η_b, we have that

$$A_1 > \max\{A_2, A_n\} \overset{\text{def.}}{=} \overline{A}_b \geqslant A_i, \ \ \forall i > 1. \tag{24}$$

  Thus, by tuning the number of steps t, we can make the ratio $(A_i/A_1)^t$ arbitrarily small for any $i > 1$.

- For η_s we have that

$$A_n > A_{n-1} \overset{\text{def.}}{=} \overline{A}_s \geqslant A_i, \ \ \forall i < n. \tag{25}$$

  Again, for a sufficiently large number of steps t, we can make $(A_i/A_n)^t$ arbitrarily small for any $i < n$.

Definition 3 (Upper bound on α.)**.** *Given some small and big learning rate η_s, η_b, we introduce α_1, a technical quantity depending on the spectrum of F and U and the initialization:*

$$\alpha_1 = \frac{1}{2}\sigma_n \iota_n^2 \exp\left(-\frac{\log\left[\|\theta_0 - \theta^\star\|_{\mathcal{H}}^2 \max\{16n\kappa_U, 4\kappa_T\} \cdot \max\left\{\frac{1}{\iota_1^2}, \frac{1}{\iota_n^2}\right\} + \frac{1}{1-\eta_s\sigma_n} + \frac{1}{\eta_b\sigma_1 - 1}\right]}{\min\left\{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}, \log\frac{A_1}{\overline{A}_b}\right\}}\right) \tag{26}$$

In particular, note that we have

$$\alpha_1 \leqslant \frac{1}{2}\sigma_1\iota_1^2 \exp\left(-\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_1^2}4n\kappa_U + \frac{1}{\eta_b\sigma_1-1}\right]}{\log\frac{A_1}{\overline{A}_b}}\right),$$

$$\alpha_1 \leqslant \frac{1}{2}\sigma_n\iota_n^2 \exp\left(-\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\max\{16n\kappa_U, 4\kappa_T\} + \frac{1}{1-\eta_s\sigma_n}\right]}{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}\right),$$

which will prove useful for the derivation of Lemmas 2 and 3.

B.1.2 An upper bound for the estimator with big learning rate

In this subsection, we denote $\mu_i^{\eta_b, t_b}$ with μ_i.

Lemma 2 (Estimator with big step size.)**.** *Set $\alpha > 0$ s.t.*

$$\alpha < \alpha_1, \tag{27}$$

where α_1 is defined in Def. 3. Define the quantity ϵ_b with

$$\epsilon_b^2 = \frac{\sum_{i>1}\mu_i^2}{\mu_1^2}. \tag{28}$$

Then, running gradient descent on F with step size η_b until the $(\alpha/2, \alpha)$ level sets are reached, i.e.

$$\frac{1}{2}\alpha \leqslant F(\theta_b) \leqslant \alpha, \tag{29}$$

ensures that

$$\epsilon_b^2 \leqslant \frac{1}{4n\kappa_U}, \ \ \textit{and} \ \ \frac{2}{5}\alpha \leqslant \frac{1}{2}\sigma_1\mu_1^2 \leqslant \alpha. \tag{30}$$

The resulting estimator is obtained with t_b steps, with

$$t_b \geqslant \frac{1}{2}\frac{\log\frac{1}{2}\frac{\sigma_1\iota_1^2}{\alpha}}{\log\frac{1}{\eta_b\sigma_1-1}} \geqslant \frac{1}{2}\frac{\log\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_1^2}4n\kappa_U}{\log\frac{A_1}{\overline{A}_b}}. \tag{31}$$

Proof.

Bound on ϵ_b. By using the definition of ϵ_b and the expression of the $(\mu_i)_{1\leqslant i\leqslant n}$ given in Def. 1, we have

$$\epsilon_b^2 = \frac{\sum_{i>1}\mu_i^2}{\mu_1^2} = \frac{\sum_{i>1}\iota_i^2 A_i^{2t}}{\iota_1^2 A_1^{2t}} \leqslant \frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_1^2}\left(\frac{\overline{A}_b}{A_1}\right)^{2t}. \tag{32}$$

Thanks to the proper choice of η_b given in Asmpt. 2, we have that $\overline{A}_b/A_1 < 1$, so that

$$\forall \delta > 0,\ \ t \geqslant t_1 \overset{\text{def.}}{=} \frac{1}{2}\frac{\log \frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\delta^2\iota_1^2}}{\log \frac{A_1}{\overline{A}_b}} \implies \epsilon_b^2 \leqslant \delta^2. \tag{33}$$

Bound on μ_1. Now, recall that the optimization error reads

$$F(\theta_b) = \frac{1}{2}\sum_{i=1}^{n}\sigma_i\mu_i^2 = \frac{1}{2}\sigma_1\mu_1^2\left(1+\frac{\sum_{i>1}\sigma_i\mu_i^2}{\sigma_1\mu_1^2}\right)$$

as we assumed that $\iota_1 \neq 0$ in Asmpt. 3. Thus, we can bound the loss in two ways. First, by definition

$$\frac{1}{2}\sigma_1\mu_1^2 \leqslant F(\theta_b) \leqslant \frac{1}{2}\sigma_1\mu_1^2(1+\epsilon_b^2) \tag{34}$$

and second, we assumed the estimator to belong to the $(\alpha/2,\alpha)$ level set of F, *i.e.*

$$\frac{\alpha}{2} \leqslant F(\theta_b) \leqslant \alpha. \tag{35}$$

Combining eq. (34) with eq. (35), we have that

$$\frac{\alpha}{2(1+\epsilon_b^2)} \leqslant \frac{1}{2}\sigma_1\mu_1^2 \leqslant \alpha. \tag{36}$$

Feasability. Finally, we consider if $\epsilon_b^2 \leqslant \frac{1}{4n\kappa_U}$ and θ_b being in the $(\alpha/2,\alpha)$ level sets can occur at the same time.

First of all, we set the value of δ^2 to $\frac{1}{4n\kappa_U}$ in eq. (34). We get

$$t_1 \overset{\text{def.}}{=} \frac{1}{2}\frac{\log \frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_1^2}4n\kappa_U}{\log \frac{A_1}{\overline{A}_b}}. \tag{37}$$

Then, we derive necessary conditions for having eq. (35). Those conditions are derived through the bounds established in eq. (34). For the lower bound, assuming $t \geqslant t_1$, so that, in particular, we have $\epsilon_b^2 \leqslant 1/4$,

$$\begin{aligned}
\frac{\alpha}{2} \leqslant F(\theta_b) &\implies \frac{\alpha}{2} \leqslant \frac{1}{2}\sigma_1\mu_1^2(1+\epsilon_b^2)\\
&\implies \frac{4\alpha}{5\sigma_1} \leqslant \mu_1^2 = \iota_1^2(\eta_b\sigma_1-1)^{2t}\\
&\implies t \leqslant t_3 \overset{\text{def.}}{=} \frac{1}{2}\frac{\log \frac{5}{4}\frac{\sigma_1\iota_1^2}{\alpha}}{\log \frac{1}{\eta_b\sigma_1-1}}.
\end{aligned}$$

Likewise, for the upper bound we have,

$$\begin{aligned}
F(\theta_b) \leqslant \alpha &\implies \frac{1}{2}\sigma_1\mu_1^2 \leqslant \alpha\\
&\implies t \geqslant t_2 \overset{\text{def.}}{=} \frac{1}{2}\frac{\log \frac{1}{2}\frac{\sigma_1\iota_1^2}{\alpha}}{\log \frac{1}{\eta_b\sigma_1-1}}.
\end{aligned}$$

Summing up, we have

- $t \geqslant t_1$ implies that the bound on ϵ_b in eq. (33) holds.

- Ensuring that the level set condition in eq. (35) holds are met implies that $t \in (t_2, t_3)$.

Thus, we need to ensure that $t_2 > t_1$ (and that $t_3 - t_2 > 1$; we assume this, as we can look at smaller level sets if necessary). To do this, we use that t_2 is an decreasing function of α. Thus, $t_2 > t_1$ as soon as

$$\alpha \leqslant \frac{1}{2}\sigma_1 \iota_1^2 \exp\left(-\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_1^2} 4n\kappa_U + \frac{1}{\eta_b\sigma_1 - 1}\right]}{\log \frac{A_1}{A_b}}\right),$$

which is exactly the purpose of the technical assumption $\alpha \leqslant \alpha_1$, with α_1 defined in Def. 3. □

B.1.3 A lower bound for the estimator with small learning rate

We now derive a similar result for θ_s. To lighten the notations, we use $\mu_i^{(\eta_s, t_s)} = \mu_i$.

Lemma 3 (Estimator with small step size.)**.** *Set $\alpha > 0$ s.t.*

$$\alpha < \alpha_1, \tag{38}$$

where α_1 is defined in Def. 3. Define the quantity ϵ_s with

$$\epsilon_s^2 = \frac{\sum_{i<n} \mu_i^2}{\mu_n^2}. \tag{39}$$

Then, running gradient descent on F with step size η_s until the $(\alpha/2, \alpha)$ level sets are reached, i.e.

$$\frac{1}{2}\alpha \leqslant F(\theta_b) \leqslant \alpha, \tag{40}$$

ensures that

$$\epsilon_s^2 \leqslant 1/(16n\kappa_U), \;\; \textit{and} \;\; \frac{2}{5}\alpha \leqslant \frac{1}{2}\sigma_n\mu_n^2 \leqslant \alpha. \tag{41}$$

The resulting estimator is obtained with t_s steps, with

$$t_s \geqslant \frac{1}{2}\frac{\log \frac{1}{2}\frac{\sigma_n \iota_n^2}{\alpha}}{\log \frac{1}{1-\eta_s\sigma_n}} \geqslant \frac{1}{2}\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\max\{16n\kappa_U, 4\kappa_T\}\right]}{\log \frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}. \tag{42}$$

Proof. The proof is very close to the one of Lemma 2. We only give the main results.

Bound on ϵ_s. This quantity can be written

$$\epsilon_s^2 = \frac{\sum_{i<n}\mu_i^2}{\mu_n^2} = \frac{\sum_{i<n}\iota_i^2 A_i^{2t}}{\iota_n^2 A_n^{2t}} \leqslant \frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\left(\frac{\overline{A}_s}{A_n}\right)^{2t},$$

with $1-\eta_s\sigma_{n-1} = \overline{A}_s > A_n = 1-\eta_s\sigma_n$ following Asmpt.2. Thus,

$$\forall \delta > 0, \;\; t \geqslant t_1 \overset{\text{def.}}{=} \frac{1}{2}\frac{\log \frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\delta^2\iota_n^2}}{\log \frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}} \implies \epsilon_s^2 \leqslant \delta^2.$$

Bound on μ_n Again, following the same paragraph in Lemma 2, we have

$$F(\theta_s) = \frac{1}{2}\sum_{i=1}^n \sigma_i\mu_i^2 = \frac{1}{2}\sigma_n\mu_n^2\left(1 + \frac{\sum_{i>1}\sigma_i\mu_i^2}{\sigma_n\mu_n^2}\right)$$

as we assumed that $\iota_n \neq 0$ in Asmpt. 3. We bound the loss in two ways. First, by definition

$$\frac{1}{2}\sigma_n\mu_n^2 \leqslant F(\theta_s) \leqslant \frac{1}{2}\sigma_n\mu_n^2(1+\kappa_T\epsilon_s^2) \tag{43}$$

and second, we assumed the estimator to belong to the $(\alpha/2, \alpha)$ level set of F, *i.e.*

$$\frac{\alpha}{2} \leqslant F(\theta_s) \leqslant \alpha. \tag{44}$$

Combining eq. (43) with eq. (44), we have that

$$\frac{\alpha}{2(1+\kappa_T\epsilon_s^2)} \leqslant \frac{1}{2}\sigma_n\mu_n^2 \leqslant \alpha. \tag{45}$$

Feasability. The discussion is the same, but with the values

$$t_1 = \frac{1}{2}\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\max\{16n\kappa_U, 4\kappa_T\}\right]}{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}$$
$$t_2 = \frac{1}{2}\frac{\log\frac{1}{2}\frac{\sigma_n\iota_n^2}{\alpha}}{\log\frac{1}{1-\eta_s\sigma_n}}$$
$$t_3 = \frac{1}{2}\frac{\log\frac{5}{4}\frac{\sigma_n\iota_n^2}{\alpha}}{\log\frac{1}{1-\eta_s\sigma_n}}.$$

Note that the addition of κ_T in the definition of t_1 is simply to ensure that

$$\forall t \geqslant t_1,\ \ \epsilon_s^2 \leqslant \frac{1}{4\kappa_T},\ \text{ so that }\ \frac{2}{5}\alpha \leqslant \frac{\alpha}{2(1+\kappa_T\epsilon_s^2)} \leqslant \frac{1}{2}\sigma_n\mu_n^2.$$

To ensure the feasibility of both bounds at the same time, we need to ensure $t_2 > t_1$. A sufficient condition for this is having

$$\alpha \leqslant \frac{1}{2}\sigma_n\iota_n^2\exp\left(-\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\max\{16n\kappa_U, 4\kappa_T\} + \frac{1}{1-\eta_s\sigma_n}\right]}{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}\right)$$

which is, again, covered with the assumption $\alpha \leqslant \alpha_1$ defined in Def. 3. □

B.1.4 Plugging the two together

Theorem 3. *Assume that the optimization error satisfy*

$$\alpha \leqslant \alpha_1, \tag{46}$$

where α_1 is defined in Definition 3. Assume that Assumption 1 on the operators T and U hold, that the condition on the learning rates η_b, η_s of Assumption 2 hold, as the condition on the initialization in Assumption 3.

Assume that gradient descent is performed until the $(\alpha/2, \alpha)$ level sets are reached. Then we have that

$$R(\theta_b) \leqslant c_\alpha R(\theta_s),\ \ \textit{with}\ \ c_\alpha = \left[\frac{1 + 2\frac{\sigma_1}{\sigma_1}\frac{R(\theta^\star)}{\alpha}}{\left(1-\sqrt{18\frac{\sigma_n}{\sigma_n}\frac{R(\theta^\star)}{\alpha}}\right)_+}\right]. \tag{47}$$

Further assume that Assumption 4 holds. Then $c_\alpha \leqslant 2$ and the bound becomes

$$R(\theta_b) \leqslant 34\frac{\kappa_U}{\kappa_T}R(\theta_s). \tag{48}$$

Proof.

Upper bound for big learning rate. We first proceed to bounding $R(\theta_b)$. In this paragraph, we use $\mu_i = \mu_i^{\eta_b, t_b}$. We have

$$\begin{aligned}
R(\theta_b) &= \frac{1}{2}\|\theta_b - \nu^\star\|_U^2 && \text{(Definition)}\\
&\leqslant \|\theta_b - \theta^\star\|_U^2 + \|\theta^\star - \nu^\star\|_U^2 && \text{(Triangular inequality)}\\
&= \|\theta_b - \theta^\star\|_U^2 + 2R(\theta^\star)\\
&\leqslant \|\mu_1 e_1\|_U^2 \left(1 + \frac{\|\sum_{i>1}\mu_i e_i\|_U}{\|\mu_1 e_1\|_U}\right)^2 + 2R(\theta^\star) && \text{(Triangular inequality)}\\
&\leqslant 2\|\mu_1 e_1\|_U^2 \left(1 + \frac{\|\sum_{i>1}\mu_i e_i\|_U^2}{\|\mu_1 e_1\|_U^2}\right) + 2R(\theta^\star) && ((a+b)^2 \leqslant 2(a^2+b^2))\\
&\leqslant 2\|\mu_1 e_1\|_U^2 \left(1 + \kappa_U \frac{|\sum_{i>1}\mu_i|^2}{|\mu_1|^2}\right) + 2R(\theta^\star) && \text{(Def. of } \kappa_U, \|e_i\| = 1)\\
&\leqslant 2\|\mu_1 e_1\|_U^2 \left(1 + \kappa_U n \epsilon_s^2\right) + 2R(\theta^\star). && \text{(Cauchy-Schwartz)}
\end{aligned}$$

Now, we use the results of Lemma 2. Firstly, we have $\kappa_U n \epsilon_b^2 \leqslant 1/4$. Secondly, we can use $\|e_1\|_U^2 \leqslant \sigma_1$. The previous inequality then turns to

$$R(\theta_b) \leqslant \frac{5\sigma_1}{2}\mu_1^2 + 2R(\theta^\star).$$

Finally, we use the fact that $\frac{1}{2}\sigma_1\mu_1^2 \leqslant \alpha$ to conclude with

$$R(\theta_b) \leqslant 5\alpha\frac{\sigma_1}{\sigma_1} + 2R(\theta^\star). \tag{49}$$

Lower bound for small learning rate. We now turn to bounding $R(\theta_s)$. Here, we use $\mu_i = \mu_i^{\eta_s, t_s}$. We have

$$\begin{aligned}
R(\theta_s) &= \frac{1}{2}\|\theta_s - \nu^\star\|_U^2 && \text{(Definition)}\\
&\geqslant \frac{1}{2}\left(\|\theta_s - \theta^\star\|_U - \|\theta^\star - \nu^\star\|_U\right)^2 && \text{(Triangular inequality)}\\
&\geqslant \frac{1}{2}\left(\|\mu_n e_n\|_U - \left\|\sum_{i<n}\mu_i e_i\right\|_U - \|\theta^\star - \nu^\star\|_U\right)^2 && \text{(Idem)}\\
&= \frac{1}{2}\|\mu_n e_n\|_U^2 \left(1 - \frac{\|\sum_{i<n}\mu_i e_i\|_U}{\|\mu_n e_n\|_U} - \sqrt{\frac{2R(\theta^\star)}{\|\mu_n e_n\|_U^2}}\right)^2\\
&\geqslant \frac{1}{2}\|\mu_n e_n\|_U^2 \left(1 - 2\frac{\|\sum_{i<n}\mu_i e_i\|_U}{\|\mu_n e_n\|_U} - \sqrt{\frac{8R(\theta^\star)}{\|\mu_n e_n\|_U^2}}\right) && \text{(As } (1-x)^2 \geqslant 1-2x)\\
&\geqslant \frac{1}{2}\sigma_n\mu_n^2 \left(1 - 2\sqrt{\kappa_U}\frac{|\sum_{i<n}\mu_i|}{|\mu_n|} - \sqrt{\frac{8R(\theta^\star)}{\sigma_n\mu_n^2}}\right) && \text{(Def. of } \kappa_U, \text{ with } \|e_i\| = 1)\\
&\geqslant \frac{1}{2}\sigma_n\mu_n^2 \left(1 - 2\sqrt{\kappa_U}\left(\frac{n\sum_{i<n}\mu_i^2}{\mu_n^2}\right)^{1/2} - \sqrt{\frac{8R(\theta^\star)}{\sigma_n\mu_n^2}}\right) && \text{(Cauchy-Schwartz)}\\
&\geqslant \frac{1}{2}\sigma_n\mu_n^2 \left(1 - 2\sqrt{\kappa_U n}\epsilon_s - \sqrt{\frac{8R(\theta^\star)}{\sigma_n\mu_n^2}}\right) && \text{(Def. of } \epsilon_s)
\end{aligned}$$

We then use Lemma 3. Firstly, we can use $\epsilon_s^2 \leqslant 1/(16n\kappa_U)$ so that $2\sqrt{\kappa_U n}\epsilon_s \leqslant 1/4$. This gives

$$R(\theta_s) \geqslant \frac{3}{8}\sigma_n\mu_n^2 \left(1 - \frac{4}{3}\sqrt{\frac{8R(\theta^\star)}{\sigma_n\mu_n^2}}\right).$$

Secondly, we have $\sigma_n \mu_n^2/2 \geqslant 2\alpha/5$. This give ultimately

$$R(\theta_s) \geqslant \frac{3}{10}\alpha\frac{\sigma_n}{\sigma_n}\left(1 - \frac{4}{3}\sqrt{\frac{8R(\theta^\star)}{\frac{4}{5}\alpha\frac{\sigma_n}{\sigma_n}}}\right) = \frac{3}{10}\alpha\frac{\sigma_n}{\sigma_n}\left(1 - \sqrt{\frac{160}{9}\frac{R(\theta^\star)}{\alpha\frac{\sigma_n}{\sigma_n}}}\right).$$

Simplifying this expression gives

$$R(\theta_s) \geqslant \frac{3}{10}\alpha\frac{\sigma_n}{\sigma_n}\left(1 - \sqrt{18\frac{R(\theta^\star)}{\alpha\frac{\sigma_n}{\sigma_n}}}\right). \tag{50}$$

Combining the two bounds. We now simply combine the upper bound of eq. (49) and the lower bound of eq. (49). We get

$$\frac{R(\theta_s)}{R(\theta_b)} \geqslant \frac{3}{50}\frac{\kappa_T}{\kappa_U}\left[\frac{1 - \sqrt{18\frac{\sigma_n}{\sigma_n}\frac{R(\theta^\star)}{\alpha}}}{1 + 2\frac{\sigma_1}{\sigma_1}\frac{R(\theta^\star)}{\alpha}}\right]. \tag{51}$$

We may prefer the other form, introducing the positive part $(x)_+ = \max(0, x)$ and using $50/3 < 17$:

$$R(\theta_b) \leqslant 17\frac{\kappa_U}{\kappa_T}\left[\frac{1 + 2\frac{\sigma_1}{\sigma_1}\frac{R(\theta^\star)}{\alpha}}{\left(1 - \sqrt{18\frac{\sigma_n}{\sigma_n}\frac{R(\theta^\star)}{\alpha}}\right)_+}\right] R(\theta_s) \overset{\text{def.}}{=} 17\frac{\kappa_U}{\kappa_T} c_\alpha R(\theta_s). \tag{52}$$

Finally, with Assumption 4 we have

$$1 + 2\frac{\sigma_1}{\sigma_1}\frac{R(\theta^\star)}{\alpha} \leqslant \frac{3}{2},$$
$$1 - \sqrt{18\frac{\sigma_n}{\sigma_n}\frac{R(\theta^\star)}{\alpha}} \geqslant \frac{1}{2},$$

so that $c_\alpha \leqslant 2$. □

B.2 Low-noise classification tasks

Taking big step size is particularly critical in classification tasks. In this section, we build on the result of [28] to relate classification performances with Hilbert norm. Recall notably the notations of section 5.4.

Assumptions. The following assumption comes from (A1) in [28]. It is well characterized in usual image classification settings.

Assumption 5 (Strong margin condition.)**.** *We have* $g^\star(x) \geqslant \delta$ *for some* $\delta \in (0, 1)$.

The second assumption characterizes the statistical optimality of $\nu^\star$. It does not assume the regression function to belong to $\mathcal{H}$, but ensures some proximity in $\mathcal{L}_\infty$ norm. It is close to (A4) in [28].

Assumption 6 (Statistical optimality of the population loss' optimum.)**.** *We have that*

$$\operatorname{Sign}(g^\star(x))\, \nu^\star(x) \geqslant \delta/2. \tag{53}$$

This assumption is satisfied as soon as the regression function can be approximated by a function of the RKHS with precision $\delta/2$ in $\mathcal{L}_\infty$ norm. For instance, a sufficient condition is the regression function $g^\star(x)$ to belong to $\mathcal{H}$. Then $g^\star = \nu^\star$ and the assumption is satisfied. Note that this always imply that $\theta^\star$ reaches 0 test error for sufficiently many samples, which is the key hindsight of [28]. Indeed, for a proper choice of regularization λ one has that

$$\|\theta^\star - \nu^\star\|_{\mathcal{H}} \lesssim n^{-\frac{br}{1+b(2r+1)}},$$

where (r, b) are the parameters of the source and capacity condition, both of which characterizes the difficulty of the learning task, see [7]. This implies that for sufficiently many training samples n, $\theta^\star$ will be close to $\nu^\star$ in Hilbert norm, which implies proximity in $\mathcal{L}_\infty$ (pointwise) norm.

Hilbert norm proximity implies statistical optimality The following lemma is very close to Lemma 1 in [28], and is a direct consquence of our assumption. We first introduce

$$\Omega_+ = \{x; g^\star(x) \geqslant \delta\}, \;\; \Omega_- = \{x; g^\star(x) \leqslant -\delta\}. \tag{54}$$

Next lemma basically relies on the decomposition

$$\|\theta - g^\star\|_{\mathcal{L}_\infty} \leqslant \|\theta - \nu^\star\|_{\mathcal{L}_\infty} + \|\nu^\star - g^\star\|_{\mathcal{L}_\infty}.$$

Lemma 4 (Small Hilbert norm implies statistical optimality)**.** *Consider an estimator θ which satisfies*

$$\|\theta - \nu^\star\|_{\mathcal{H}} \leqslant \frac{\delta}{2C_K}.$$

Then, this estimator is statistically optimal, in the sense that it has 0 excess error:

$$B(\theta) - \inf_{\theta \in \mathcal{H}} B(\theta) = 0.$$

Proof. First of all, we leverage the fact that the Hilbert norm upper bounds the L_∞ norm, with

$$\|\theta - \nu^\star\|_{L_\infty} \leqslant C_K \|\theta - \nu^\star\|_{\mathcal{H}} \leqslant \frac{\delta}{2}.$$

Then, on Ω_+, whose definition is given in eq. (54), we have that

$$\forall x \in \Omega_+, \theta_x > g^\star(x) - \|\theta - \nu^\star\|_{L_\infty} - \|\nu^\star - g^\star\|_{L_\infty} \geqslant \delta - \frac{\delta}{2} - \frac{\delta}{2} = 0,$$

so θ will have accurate prediction for all positive labels. The same goes for negative labels. Thus, θ has 0 test error. □

Thus, we see that the *Hilbert norm is a good proxy for minimizing the test error* B.

B.3 Regression tasks and comparison with spectral filters

If the downstream task is regression, then we can still apply our result by introducing the *population covariance operator*,

$$U = \int \phi(x) \otimes \phi(x) d\rho_x(x). \tag{55}$$

Then, Theorem 2 holds by considering (recall the definition of the population loss $\mathcal{P}$ in eq. (17))

$$R(\theta) = \frac{1}{2}\|\theta - \nu^\star\|_U^2 = \mathcal{P}(\theta) - \inf_{\nu \in \mathcal{H}} \mathcal{P}(\nu),$$

which is nothing but the *excess risk* of the estimator. Then, under the assumptions of Theorem 2 we have that

$$R(\theta_b) \;\leqslant\; 34 \frac{\kappa_{\tilde{U}}}{\kappa_T} R(\theta_s). \tag{56}$$

Gradient descent for kernel ridge regression has been widely studied in the past, to say the least. Equation (56) appears to be in contradiction with most of them. In this section, we emphasize the limit of our assumptions to point out that there is no conflict with existing theory.

Early stage of training. The bound in eq. (56) ensures better generalization when taking big step size, if the r.h.s is bigger than 1. However, the pioneering work of [34] established that the learning rate had no influence in the generalization capabilities of the estimator. A key difference though is that the results of Theorem 2 only holds in the early stage of training, when the optimization error α is big w.r.t to the statistical error $R(\theta^\star)$: otherwise, Assumption 4 is not satisfied. In contrast, results of the like of [34] holds for sufficiently many samples n, and require a number of steps t bounded by below by a power of n – they require an upper-bound on α, while we require a lower-bound in Assumption 4.

Is eq. (56) informative? As mentioned above, eq. (56) ensures better generalization of big step size only if the r.h.s is bigger than 1. However, in the particular scenario of kernel regression, the empirical covariance T is the *discretization* of the population covariance U. Thus, numerous results bound the discrepancy between the two, notably when the capacity condition holds, see *e.g.* Proposition 5.3 to 5.5 in [7]. In these settings, we can expect the ratio κ_T/κ_U to go to 1 for large number of samples n. Thus, we cannot conclude in better excess risk of θ_b compared to θ_s.

Comparison with spectral filters. Spectral filters are an elegant way to describe a wide family of regularization for kernel regression [15, 5]. In a nutshell, it relies on studying the class of estimator characterized by a filter function g_λ, where λ is a regularization parameter, equal to $1/t$ in the case of early stopping in GD. GD with moderate step sizes is a spectral filter; but GD with large step size is not. We now explain this difference, which helps to build an intuition on our result.

Consider the estimator $\theta = g_\lambda(T)S^*y$, where S is the so-called sampling operator defined in section B.4.1. The unregularized solution is obtained with $\lambda = 0$, for which we must have $g_{\lambda=0}(\sigma) = \sigma^{-1}$. We denote it with $\theta^\star = T^{-1}S^*y$, and we can see how does θ approaches the unregularized optimum. We have

$$\begin{aligned}\langle \theta - \theta^\star, e_i\rangle &= \left\langle g_\lambda(T)S^*y - T^{-1}S^*y, e_i\right\rangle \\ &= \left\langle \left(g_\lambda(T)T^{-1} - \mathbf{I}\right) T^{-1}S^*y, e_i\right\rangle \\ &= \left\langle \left(g_\lambda(T)T - \mathbf{I}\right) \theta^\star, e_i\right\rangle \\ &= \left(g_\lambda(\sigma_i)\sigma_i - 1\right) \langle \theta^\star, e_i\rangle \,.\end{aligned}$$

If we start from $\theta_0 \neq 0$, this relation turns to $|\langle \theta - \theta^\star, e_i\rangle| = |1 - g_\lambda(\sigma_i)\sigma_i|\,|\langle \theta_0 - \theta^\star, e_i\rangle|$ in the case of GD. We denote the residual with $r_\lambda(\sigma) = |1 - \sigma g_\lambda(\sigma)|$. We then compare r_λ for various spectral filters in fig. 5. Note that for gradient descent, we have $r_{1/t}(\sigma) = |1 - \eta\sigma|^t$ and we recover the expression we obtained from eq. (3). The key hindsight is that spectral filters will learn, *i.e optimize*, the *biggest eigendirection* first. For instance, truncated regression uses as estimator the first eigencomponents of the unregularized estimator, leaving the smaller eigencomponents untouched. This is at odds with what we aim at with big learning rate. There is no contradictions though, as we want in the end to minimize the excess risk R – a quadratic with operator U – and we assumed the empirical covariance T to be a discretization of U. Thus, in this settings F is a good proxy for R and minimizing the biggest eigendirection first will make the excess risk R decrease faster. This corresponds to having level sets well aligned in fig. 1.

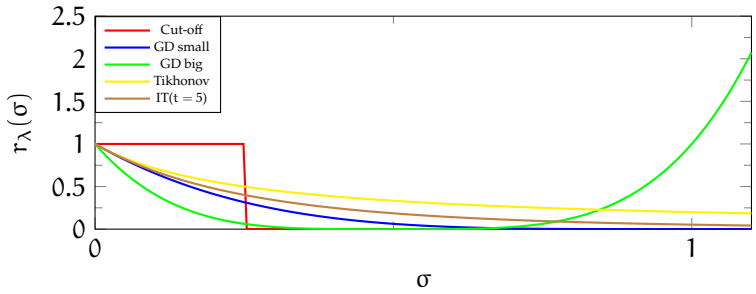


Figure 5: Residual of various spectral filters, with regularization $\lambda = 1/4$ or $t = 4$ for GD. The best filter is spectral cut-off *(red)*. The resulting estimator is purely directed along the smallest eigenvectors of T. Gradient descent with small step sizes *(blue)* and (iterated) Tikhonov *(yellow, brown)* mimick this filter. On the other hand, gradient descent with big step sizes *(green)* is not an admissible filter in the sense of [15], as it attenuates less the biggest component.

Theorem 2 in practice. The limits of this subsection – low optimization regime, low value for κ_T/κ_U and difference with spectral filters – can be mitigated for multiple reasons. First of all, as we discussed earlier kernel regression can simply be a mean in order to solve a *classification task*, in which case the statistical results of spectral filters are no longer relevant. Secondly, there

can be big discrepancies between the train and test set in practice. Indeed, the risk R with which estimators are compared is often a separate test set, with fixed condition number κ_U. Additionally, data augmentation can be used on the train set, which then introduces spurrious directions in the empirical covariance matrix T. Thus, even though spectral filters are optimal in theoretical settings, taking big step size can prove useful in practical scenari, which are covered by our settings with quadratic forms of $\mathcal{H}$.

B.4 Gradient descent updates in practice

B.4.1 Useful operators

We assume there are n training samples. If considered, the test loss consists of m samples.

We denote $\widehat{S}, \widehat{S}^\star$ the *sampling* operator and its dual, which are defined as

$$\begin{aligned} \widehat{S} : \mathcal{H} \to \mathbb{R}^n, \quad \forall f \in \mathcal{H},\ \widehat{S}(f) &= \frac{1}{\sqrt{n}} \begin{pmatrix} \langle f, \phi(x_1)\rangle_{\mathcal{H}} \\ \vdots \\ \langle f, \phi(x_n)\rangle_{\mathcal{H}} \end{pmatrix} \\ \widehat{S}^\star : \mathbb{R}^n \to \mathcal{H}, \quad \forall \alpha \in \mathbb{R}^n,\ \widehat{S}^\star(\alpha) &= \frac{1}{\sqrt{n}} \sum_{i=1}^n \alpha_i \phi(x_i), \end{aligned} \tag{57}$$

so that the covariance operator $T = \widehat{S}^\star \widehat{S}$ and the kernel matrix K write

$$\begin{aligned} T : \mathcal{H} \to \mathcal{H}, \quad T &= \widehat{S}^\star \widehat{S} \\ K/n : \mathbb{R}^n \to \mathbb{R}^n, \quad \frac{K}{n} &= \widehat{S}\widehat{S}^\star. \end{aligned} \tag{58}$$

The population version are $S, S^\star, U$. There are written with an expectation, or with the test dataset as a proxy.

Note that we have $T^{-1}\widehat{S}^\star = \widehat{S}^\star (K/n)^{-1}$. We denote σ_i, e_i the spectrum of T and u_i the eigenvectors of K/n (not K), with the same spectrum. The eigenvectors in $\mathcal{H}$ and $\mathbb{R}^n$ are related with

$$\forall i \in \{1, \dots, n\},\ u_i = \frac{1}{\sqrt{\sigma_i}} \widehat{S} e_i,\ e_i = \frac{1}{\sqrt{\sigma_i}} \widehat{S}^\star u_i.$$

Finally, an estimator $\theta \in \mathcal{H}$ can be represented with a vector $\alpha \in \mathbb{R}^n$. Specifically, we have the relation

$$\theta = \sqrt{n}\widehat{S}^\star \alpha \iff \sqrt{n}\widehat{S}\theta = K\alpha \iff \alpha = \sqrt{n} K^{-1} \widehat{S}\theta. \tag{59}$$

B.4.2 Gradient descent on the Hilbert norm is possible

Different spectrum between $\mathcal{H}$ and $\mathbb{R}^n$. We denote the training loss with F and the Hilbert norm with $L_{\mathcal{H}}$. Given the relation of eq. (59), we have that

$$\begin{aligned} F(\theta) &= \frac{1}{2} \|\theta - \theta^\star\|_T^2 = \frac{1}{2n} \|K(\alpha - \alpha^\star)\|_{\mathbb{R}^n}^2, \\ L_{\mathcal{H}}(\theta) &= \frac{1}{2} \|\theta - \theta^\star\|_{\mathcal{H}}^2 = \frac{1}{2} \|\alpha - \alpha^\star\|_K^2, \end{aligned} \tag{60}$$

where we overloaded F to be a function of $\mathcal{H}_n$ *and* $\mathbb{R}^n$. Specifically, we used $F(\alpha) = F \circ \sqrt{n}\widehat{S}^\star(\alpha)$. Recall that K/n and T share the same spectrum. Thus, F is a quadratic whose spectrum is $(\sigma_1, \dots, \sigma_n)$ w.r.t the variable θ, but spectrum $(n\sigma_1^2, \dots, n\sigma_n^2)$ w.r.t the variable α. Likewise, $L_{\mathcal{H}}$ is a quadratic with spectrum $(1, \dots, 1)$ w.r.t the variable θ, but a spectrum $(n\sigma_1, \dots, n\sigma_n)$ w.r.t the variable α.

Do we care about this difference? The global picture behind what follows is that α is isomorphic to $T^{-1/2}\mathcal{H}$. If we expressed the estimator θ as a combination of eigenbasis vector, that is $\theta = \sum_i \beta_i e_i$, then β is isormorphic to $\mathcal{H}$ and the distinction does not hold. The fact that the estimator writes as a combination of $\phi(x_i)$ with α adds another level of geometric distortion.

Gradient descent in $\mathcal{H}$. In the Hilbert space $\mathcal{H}$, the updates are easy:

$$\begin{aligned}
\theta_{t+1} = \theta_t - \eta T(\theta_t - \theta^\star) &\iff \theta_t - \theta^\star &&= (\mathbf{I} - \eta T)^t (\theta_0 - \theta^\star), && \text{(GD on F)} \\
\theta_{t+1} = \theta_t - \eta(\theta_t - \theta^\star) &\iff \theta_t - \theta^\star &&= (1-\eta)^t (\theta_0 - \theta^\star), && \text{(GD on } L_{\mathcal{H}}\text{)}.
\end{aligned}$$

The big learning rate range is then $\eta_s < 2/(\sigma_1 + \sigma_n) < \eta_b < (2/\sigma_1)$.

Gradient descent in $\mathbb{R}^n$. In practice, we do not have access to $\alpha^\star$, or only though it's evaluation with K. Yet, we are still able to minimize these quadratic form through the gradient. *E.g.* when $\alpha^\star$ is defined through $K\alpha^\star = y$ in the unregulariz ed settings, or $(K + n\lambda)\alpha^\star = y$ in the Tikhonov-regularized case. The gradient descent updates on the train loss read:

$$\begin{aligned}
\alpha_{t+1} = \alpha_t - \eta \frac{K^2}{n}(\alpha_t - \alpha^\star) = \alpha_t - \eta \frac{K}{n}(K\alpha_t - y) \\
\iff \alpha_t - \alpha^\star = \left(\mathbf{I} - \eta \frac{K^2}{n}\right)^t (\alpha_0 - \alpha^\star) \qquad \text{(GD on F)}
\end{aligned} \tag{61}$$

Here, the range of learning rate is $\eta_s < 2/\left[n(\sigma_1^2 + \sigma_n^2)\right] < \eta_b < 2/\left[n\sigma_1^2\right]$. Interestingly, we can still do gradient descent on the Hilbert norm in closed form!

$$\begin{aligned}
\alpha_{t+1} = \alpha_t - \eta K(\alpha_t - \alpha^\star) = \alpha_t - \eta(K\alpha_t - y) \\
\iff \alpha_t - \alpha^\star = (\mathbf{I} - \eta K)^t (\alpha_0 - \alpha^\star) \qquad \text{(GD on } L_{\mathcal{H}}\text{)}
\end{aligned} \tag{62}$$

Here, the optimal learning rate is $1/[n\sigma_1]$. Interestingly, choosing a big learning rate in the range $1/\left[n(\sigma_1 + \sigma_n)\right] < \eta_b < 2/\left[n\sigma_1\right]$ results in an estimator which is closed in *euclidean* norm ($\mathbb{R}^n$) to $\alpha^\star$. Note that even though we can evaluate the gradient of $L_{\mathcal{H}}$, we *cannot* evaluate its value. Indeed, the objective function would read

$$\frac{1}{2}\|\alpha - \alpha^\star\|^2 = \frac{1}{2}\left\|\alpha - K^{-1}y\right\|^2$$

which is not accessible without inverting the (regularized) kernel matrix.

B.5 Additional details on the experiment

Setting the learning rate. We give additional details on the plot "test accuracy function of train loss α" in fig. 3. The plot is averaged over 10 initialization for θ_0. We used $\eta_s = 1/\sigma_1$ and $\eta_b = \tau \cdot 2/\sigma_1$, with $\tau = 1 - 10^{-5}$. We elaborate on these choices:

- The optimal learning rate for upper bounding for σ_1-smooth, σ_n-strongly convex function is $\eta_{opt} = 2/(\sigma_1 + \sigma_n)$, as explained in the discussion of Assumption 2. However, this requires a massive amount of steps to converge. This is due to the terms depending on the initialization in the lower bound for t_b, t_s in eqs. (31) and (42). Thus, we set $\eta_s = 1/\sigma_1$, which is the optimal rate for σ_1-smooth function, and we do observe fast convergence with this choice.
- Instead of choosing $\eta_b \in [2/(\sigma_1 + \sigma_n), 2/\sigma_1]$, we use $\eta_b = \tau \cdot 2/\sigma_1$, with τ chosen with the experiment on the test accuracy (fig. 3, left). Indeed, setting $\tau = 1$ can result in situation where there can't be convergence; and choosing $\eta_b > \eta_{opt}$, as we describe in the theory, results in very slow convergence.

This discrepancy between theory and practice is due to our proof which is very conservative in the error bound. A more refined analysis would rely on $\theta_b - \theta^\star$ (resp. $\theta_s - \theta^\star$) belonging to the span of the k-th first (resp. last) eigenvectors. Besides, in practical settings the learning rate is an hyperparameter to tune, which is exactly the approach we used to produce fig. 3.

References for part II

[1] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[2] J-Y Audibert. Classification under polynomial entropy and margin assumptions and randomized estimators. 2004.

[3] Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 35(2):608–633, 2007.

[4] Peter L Bartlett, Philip M Long, and Olivier Bousquet. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *Journal of Machine Learning Research*, 24(316):1–36, 2023.

[5] F. Bauer, S. Pereverzev, and L. Rosasco. On regularization algorithms in learning theory. *Journal of complexity*, 23(1):52–72, 2007.

[6] Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. On the Benefits of Large Learning Rates for Kernel Methods. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 254–282. PMLR, June 2022.

[7] G. Blanchard and N. Mücke. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18, 2016. doi: 10.1007/s10208-017-9359-7.

[8] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[9] Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[10] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2021.

[11] Alex Damian, Tengyu Ma, and Jason D. Lee. Label Noise SGD Provably Prefers Flat Global Minimizers, December 2021.

[12] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp Minima Can Generalize For Deep Nets, May 2017.

[13] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A. Hamprecht. Essentially No Barriers in Neural Network Energy Landscape, February 2019.

[14] Jonas Geiping, Micah Goldblum, Phillip E. Pope, Michael Moeller, and Tom Goldstein. Stochastic training is not necessary for generalization. *arXiv preprint arXiv:2109.14119*, 2021.

[15] L. Lo Gerfo, L. Rosasco, F. Odone, E. De Vito, and A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008. doi: 10.1162/neco.2008.05-07-517. URL `https://app.dimensions.ai/details/publication/pub.1045202542andhttp://www.dima.unige.it/~devito/pub_files/spectral_finale.pdf`.

[16] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging Weights Leads to Wider Optima and Better Generalization, February 2019.

[17] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[18] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[19] Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof Geras. Catastrophic Fisher Explosion: Early Phase Fisher Matrix Impacts Generalization, June 2021.

[20] Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic Fisher explosion: Early phase Fisher matrix impacts generalization. In *International Conference on Machine Learning (ICML)*, 2021.

[21] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[22] Julien Mairal. End-to-End Kernel Learning with Supervised Convolutional Kernel Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[23] Preetum Nakkiran. Learning rate annealing can provably help generalization, even for convex problems. *arXiv preprint arXiv:2005.07360*, 2020.

[24] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[25] Yurii Nesterov. *Lectures on Convex Optimization*. Springer, 2018.

[26] Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative Flatness and Generalization, November 2021.

[27] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical Optimality of Stochastic Gradient Descent on Hard Learning Problems through Multiple Passes. In *Neural Information Processing Systems (NeurIPS)*, 2018.

[28] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Exponential convergence of testing error for stochastic gradient methods. In *Conference On Learning Theory (COLT)*, 2018.

[29] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[30] Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations (ICLR)*, 2021.

[31] E. De Vito, L. Rosasco, A. Caponnetto, U. De Giovannini, and F. Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research (JMLR)*, 6(30):883–904, 2005. URL `http://jmlr.org/papers/v6/devito05a.html`.

[32] Jingfeng Wu, Difan Zou, Vladimir Braverman, and Quanquan Gu. Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate. In *International Conference on Learning Representations (ICLR)*, 2021.

[33] Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with SGD. *arXiv preprint arXiv:1802.08770*, 2018.

[34] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Part III
Kernel Methods for Global Optimization

Section 9 in this part is based on our third article [4],

> Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. GloptiNets: Scalable Non-Convex Optimization with Certificates. In Advances in Neural Information Processing Systems, November 2023.

In section 6 we first cover polynomial optimization, a closely related research fields. Indeed, it is concerned too with providing certificates to general, non-convex polynomials. We use this detour to propose a new greedy algorithms for polynomial optimization. Then, section 7 introduces kernel sum-of-squares. We discuss the different point of views to achieve non-convex optimization with k-SoS and notably provide hindsight on the dual of K-SoS problems. With those hindight, section 8 is dedicated to outlining open challenges in this field, with open problems 1 to 6. Finally, section 9 introduce the main contribution of this thesis in non-convex optimization, which is the GloptiNets algorithm.

6 Polynomial Optimization

Kernel Sum of Squares (K-SoS), a fundamental component of our GloptiNets algorithm presented in section 9, represents a relatively new field introduced in 2020 with [22]. K-SoS is primarily used to model positive functions and has been applied in global optimization contexts [27, 38]. In contrast, the field of polynomial optimization, underpinned by Polynomial Sum of Squares (P-SoS) in polynomial hierarchies, is well-established and mature with two decades of development [17]. Importantly, P-SoS can be viewed as a specific subset of K-SoS. This relationship naturally prompts an in-depth investigation into how these two areas, K-SoS and P-SoS, are interconnected, to identify the most effective way to leverage K-SoS for global optimization..

Many works delve into the link between K-SoS and P-SoS: see chap. 8 in [21] or [1]. We do not aim for a comprehensive literature review in this section. Instead, our attention is directed towards three specific objectives: *(i)* Concentrating on Complex Polynomial Optimization (C-POP), an area of application of our GloptiNets algorithm presented in section 9; *(ii)* Emphasizing how sparsity is exploited in C-POP, an aspect that might be underappreciated from a K-SoS perspective; and *(iii)* Introducing an innovative approach for C-POP, which involves adapting the Matching Pursuit algorithm to facilitate the generation of sparse P-SoS solutions.

6.1 A brief overview of polynomial hierarchies

Remark 1 (Exhaustive litterature review.)**.** *For an exhaustive treatment of real polynomial hierarchies, see [13]. Complex polynomial hierarchies can be reframed as a real polynomial optimization problem. Yet, specific tools, such as* Hermitian Sum-of-Squares*, which we detail here, are introduced in [16] and enhanced in [35]. Finally, we only tackle unconstrained minimization and do not mention localization matrices, a crucial aspect of POP.*

This part will be useful nonetheless to introduce notations used in sections 6.2 and 6.3.

We want to minimize a trigonometric polynomial f defined on the torus in dimension d, as[2]

$$f_\star = \min_{z\in\mathbb{T}^d} \left\{ f(z) = \sum_{\omega\in\mathbb{Z}^d} f_\omega z^\omega \right\}. \tag{1}$$

We note $f_\omega = 0$ for $\|\omega\|_1 \geqslant \deg f$. Similarly to how we introduced GloptiNets in eq. (7) (section 3.3.2 in the introduction), the minimum can be written

$$f_\star = \sup_{c,g\geqslant 0} \{c \text{ s.t. } f - c = g\}. \tag{2}$$

We restrict the set of positive function to *Hermitian Sum-of-Squares* (H-SoS), introduced in [16]. For some $p \in \mathbb{N}$, we define the embedding

$$\forall z \in \mathbb{T}^d, \;\; \varphi_p(z) = (z^\omega)_{\omega\in\mathbb{Z}^d, \|\omega\|_1\leqslant p} \tag{3}$$

and a H-SoS is a polynomial g of degree at most $2p$ defined with a Hermitian matrix $G \succeq 0$ s.t.

$$\forall z \in \mathbb{T}^d, \;\; g(z) = \varphi_p(z)\cdot G\varphi_p(z) = \sum_{\gamma\in\mathbb{Z}^d} \left(\sum_{\substack{\|\alpha\|_1,\|\beta\|_1\leqslant p \\ \beta-\alpha=\gamma}} G_{\alpha\beta} \right) z^\gamma \overset{\text{def.}}{=} \sum_{\|\gamma\|\leqslant 2p} G\cdot B_\gamma z^\gamma. \tag{4}$$

Equation (4) clearly shows that g is a SoS hence positive (thanks to $G \succeq 0$). The coefficients of g are given with $G\cdot B_\gamma$, where $\cdot$ is the Frobenius inner product, and B_γ is a non-Hermitian, antisymmetric boolean matrix, defined with

$$\forall \|\alpha\|, \|\beta\| \leqslant p, \;\; (B_\gamma)_{\alpha\beta} = \mathbf{1}_{\beta-\alpha=\gamma}.$$

[2]The POP literature use n for the dimension and d for the degree; we keep the notations from the kernel literature, and use d for the dimension and p for the degree.

Hence, a lower bound on $f_\star$ can be obtained in using a H-SoS of eq. (4) in eq. (2), which yields

$$\begin{aligned} f_\star &\geqslant \max_{c, G \succeq 0} c \ \text{ s.t. } \forall z, f(z) - c = \varphi_p(z) \cdot G\varphi_p(z) \\ &= \max_{c, G \succeq 0} c \ \text{ s.t. } \forall \gamma, f_\gamma - c\mathbf{1}_{\gamma=0} = G \cdot B_\gamma. \end{aligned} \tag{POP-D}$$

Problem (POP-D) is a SDP, whose dual is

$$\begin{aligned} \min_{y \in \mathbb{C}^q} \ & \sum_\gamma f_\gamma \overline{y}_\gamma \\ \text{s.t. } & y_0 = 1 \\ & M(y) = \sum_\gamma y_\gamma B_\gamma \succeq 0, \end{aligned} \tag{POP-P}$$

q is the cardinal of $\{\gamma \in \mathbb{Z}^d; \|\gamma\| \leqslant 2p\}$ and $M(y)$ is the *moment matrix* of a distribution μ on $\mathbb{T}^d$. Indeed, the optimization problem (POP-P) embodies the *moment point of view* of polynomial optimization. y can be seen as the moments of a signed measure μ on $\mathbb{C}^d$, with $y_\gamma = \int z^\gamma d\mu(z)$. The condition $y_0 = 1$ becomes $\int d\mu = 1$, and the condition on $M(y)$ becomes

$$\forall h, \deg h \leqslant p, \ \int_{\mathbb{T}^d} |h|^2 \, d\mu \geqslant 0.$$

This illustrate that the *strengthening* of looking for g as a Hermitian SoS (going from eq. (2) to problem (POP-D) amounts to a relaxation in the dual, going from

$$f_\star = \inf_{\mu \in \mathcal{M}(\mathbb{T}^d)} \int_{\mathbb{T}^d} f d\mu, \ \text{ s.t. } \ \int d\mu = 1, \ \ \mu \geqslant 0$$

to

$$f_\star \geqslant \inf_{\mu \in \mathcal{M}(\mathbb{T}^d)} \int_{\mathbb{T}^d} f d\mu, \ \text{ s.t. } \ \int d\mu = 1, \ \ \forall h, \deg h \leqslant p, \int |h|^2 \, d\mu \geqslant 0.$$

In other words, we relaxed μ from being a positive measure to being positive when measuring sum-of-squares.

6.2 Previous works on exploiting sparsity in POP

In the field of certificate-based polynomial optimization, Lasserre's hierarchy plays a pivotal role [17, 18]. This hierarchy employs a sequence of SDP relaxations with increasing dimensions that ultimately converge to the optimal solution. While Lasserre's hierarchy is primarily associated with polynomial optimization, its applicability extends beyond this domain. It offers a specific formulation for the more general moment problem, enabling a wide range of applications; see [13] for an introduction. For polynomial optimization problems such as in eq. (1), a significant amount of research has been dedicated to leveraging problem structure to improve the scalability of the hierarchy. This research has predominantly focused on exploiting sparsity in the matrix G, enabling the handling of problem instances ranging from a few variables to even thousands of variables.

We give a brief overview of these approaches. The first approach that has been explored is "correlative sparsity", initially introduced by [32]. This approach focuses on exploiting sparsity patterns among the variables of the problem. However, it tends to be less effective when the variables are interconnected. Typically, the inclusion of a ball constraint, such as $\|x\|^2 \leqslant 1$, eliminates this type of sparsity.

Another approach, "term sparsity" was introduced by [36]. It targets sparsity between the monomials in the decomposition of $f - f_\star$ as a sum-of-squares, which amounts to finding a *block structure* in the matrix Q in problem (POP-D): then, we solve multiple small SDP problems instead of a single, larger one. As solving a SDP has cubic complexity in the size of the problem, this results in a significant speedup.

Term sparsity encompasses correlative sparsity and provides a hierarchy of lower bounds that converges to the optimal value $f_\star$. Finally, an extension of term sparsity, known as "chordal

sparsity" was proposed in [35]. This variant scales better in practical applications but no longer guarantees convergence to $f_\star$ (counterexamples exist).

Complex polynomial optimization, a specific branch within the field, is particularly interesting due to its relevance in solving industrial problems, such as the optimal power flow (OPF) problem [5]. In this context, various works have aimed to exploit the special structure inherent to complex polynomial optimization. One notable contribution is the introduction of "Hermitian Sum of Squares" by [16], which leverages this specific structure. More recent research has explored the combined utilization of term sparsity and correlative sparsity, as demonstrated by [34]. These approaches have successfully provided certificates for OPF instances involving a significant number of complex variables (2869).

Furthermore, alternative approaches exist that exploit different types of structure, such as the constant trace property, as exemplified in the work by [20].

6.3 Greedy-POP

This section contains unpublished material done in collaboration with Francis Bach.

6.3.1 Main Result

Following section 6, we proceed to finding a lower bound on $f_\star$ defined in eq. (1). Polynomial hierarchies rely on Hermitian SoS, as defined in eq. (4). We follow a slightly different approach, by allowing a flexible choice for the frequencies appearing in the SoS.

Definition 1 (Subset of SoS)**.** *Let $t \geqslant 0$. For a set of $t+1$ unique frequencies $\Omega^{(t)} = 0, \omega_1, \dots, \omega_t \subset \mathbb{Z}^d$ and a Hermitian matrix $G \succeq 0$ in $\mathbb{C}^{(t+1)\times(t+1)}$, we define the SoS $g_{\Omega^{(t)},G}$ as*

$$\forall z \in \mathbb{T}^d, \;\; g_{\Omega^{(t)},G}(z) = \varphi^{(t)}(z) \cdot G\varphi^{(t)}(z). \tag{5}$$

$g_{\Omega^{(t)},G}$ is a SoS and we denote $\mathcal{G}^{(t)}$ the set $\{g_{\Omega^{(t)},G}; \Omega^{(t)}, G\}$. We further define

$$\Delta^{(t)} = \{\omega_j - \omega_i; 1 \leqslant i \leqslant j \leqslant t\}, \;\; B_\omega \in \mathbb{R}^{(t+1)\times(t+1)}, (B_\omega)_{ij} = \mathbf{1}_{\omega_j - \omega_i = \omega} \tag{6}$$

which are antisymmetric boolean matrices, so that

$$g_{\Omega^{(t)},G}(z) = \sum_{\omega \in \pm\Delta^{(t)}} (G \cdot B_\omega) z^\omega. \tag{7}$$

Remark 2 ($\Omega^{(t)}$ for Lasserre's hierarchy)**.** *In the classical case of section 6, we have $\Omega^{(t)} = \{\omega; \|\omega\|_1 \leqslant t\}$, for which $\left|\Omega^{(t)}\right| = O(\binom{t+d}{t})$.*

Second, instead of using the formulation of problem (POP-D) (or its dual in problem (POP-P)), we use a penalized objective with the F-norm. The F-norm, taken from [38] and introduced in eq. (9) in the introduction, is the ℓ_1 norm of the Fourier coefficient of a periodic function. It upper bounds the L_∞ norm, hence the following result.

Proposition 1 (Certificate with the F-norm)**.** *Let $t \geqslant 0$ and Θ the set of frequencies of f. With $\Omega^{(t)} \subset \mathbb{Z}^d$ and $G \succeq 0$ as introduced in definition 1, the following bound on $f_\star$ holds*

$$f_\star \geqslant \sup_{\Omega^{(t)},G} L(G, \Omega^{(t)}) = (\widehat{f}_0 - G \cdot B_0) - 2 \sum_{\omega \in \Theta \cup \Delta^{(t)} \setminus \{0\}} \left|\widehat{f}_\omega - G \cdot B_\omega\right| \tag{8}$$

If $f - f_\star \in \mathcal{G}^{(t)}$, the bound is tight.

Proof. Let $\Omega^{(t)} \subset \mathbb{Z}^d, G \succeq 0$ as in definition 1. From theorem 2 in section 9, we have that

$$\forall c, g_{\Omega^{(t)},G} \in \mathcal{G}^{(t)}, \;\; f_\star \geqslant c - \|f - c - g_{\Omega^{(t)},G}\|_F, \;\; \text{where} \;\; \|u\|_F = \sum_{\omega \in \mathbb{Z}^d} |\widehat{u}_\omega|. \tag{9}$$

Using the expression of the coefficients of $g_{\Omega^{(t)},G}$ in definition 1 we have

$$c - \|f - c - g_{\Omega^{(t)},G}\|_F = c - \sum_{\omega \in \Theta \cup \pm\Delta^{(t)}} \left|\widehat{f}_\omega - c - G \cdot B_\omega\right|. \tag{10}$$

Replacing eq. (10) into eq. (9) and optimizing c out, we get the result in eq. (8). □

Solving eq. (8) requires finding an optimal set of frequencies along with the p.d. matrix G. This is an NP-hard problem, so we update $\Omega^{(t)}$ in a greedy fashion, in the spirit of matching pursuit (see *e.g.* Chap 7 in [2]).

We will look for ω_{t+1} in the neighborhood of $\Omega^{(t)}$, as next definition shows.

Definition 2 (Set of candidates)**.** *Given $\Omega^{(t)}$, we define the* candidate set *at step t with*

$$\mathcal{C}^{(t)} = \left\{ \omega \in \mathbb{Z}^d : \omega \notin \pm\Omega^{(t)}, \exists \nu \in \Omega^{(t)} \text{ s.t. } \|\omega - \nu\|_1 = 1 \right\}. \tag{11}$$

Note that $\left|\mathcal{C}^{(t)}\right| \leqslant 2d(t+1)$. Now, assume we are given $\Omega^{(t)}, G^{(t)}$. Given a candidate $\omega \in \mathcal{C}^{(t)}$, we evaluate the improvement in retaining it in $\Omega^{(t+1)}$ with

$$S^{(t)}(\omega) = \sup_{G' \in \mathcal{S}_+(\mathbb{C}^{t+2})} L(G', \Omega^{(t)} \cup \{\omega\}) - L(G^{(t)}, \Omega^{(t)}) \geqslant 0. \tag{12}$$

Computing $S^{(t)}(\omega)$ exactly requires solving a SDP, which is costly. We opt for a proxy of the improvement, which is cheaper to compute.

Definition 3 (Proxy for the improvement)**.** *Given $\Omega^{(t)}, G^{(t)}$, we define the function $\tilde{S}^{(t)} : \mathbb{Z}^d \to \mathbb{R}^+$ with*

$$\tilde{S}^{(t)}(\omega) = \sup_{\alpha, \nu \in \mathbb{C}^{t+1}} L\left(\begin{pmatrix} G^{(t)} & \nu \\ \nu^* & \alpha \end{pmatrix}, \Omega^{(t)} \cup \{\omega\}\right) - L(G^{(t)}, \Omega^{(t)}) \tag{13}$$
$$\text{s.t. } \nu^* G^{(t),-1} \nu \leqslant \alpha.$$

This is a much simpler problem, which turns out being a Lasso problem, as next Theorem shows.

Theorem 1 (Lower-bounding the improvement with a Lasso problem)**.** *Let $T^\top T = G^{(t),-1} =$ be the Cholesky factorization of $G^{(t),-1}$, assumed positive definite. For a candidate $\omega \in \mathcal{C}^{(t)}$, define $\delta_i = \omega - \omega_i$ for $i \in [0, t]$, and $y = T(g^{(t)}_{\delta_i} - f_{\delta_i})_{0 \leqslant i \leqslant t} \in \mathbb{C}^{t+1}$. The proxy for the improvement $\tilde{S}^{(t)}$ in definition 3 satisfies*

$$\tilde{S}^{(t)}(\omega) = 2\left\|T^{-1}y\right\|_1 - 2\textsc{Lasso}(T, y), \quad \textit{with } \textsc{Lasso}(A, y) = \min_{x \in \mathbb{C}^{t+1}} \frac{1}{2}\|Ax - y\|_2^2 + \|x\|_1. \tag{14}$$

It is a lower bound on $S^{(t)}(\omega)$, as

$$0 \leqslant \tilde{S}^{(t)}(\omega) \leqslant S^{(t)}(\omega). \tag{15}$$

The optimal (α, ν) in eq. (13) *can be retrieved from a solution $x_\star$ of the Lasso problem in eq.* (14) *with*

$$\nu = x_\star - T^{-1}y, \quad \alpha = \|T\nu\|_2^2. \tag{16}$$

Remark 3 (Necessary condition for non-null improvement)**.** *If $\bar{x} = (g^{(t)}_{\delta_i} - f_{\delta_i})_{0 \leqslant i \leqslant t}$ is a solution of the Lasso problem in eq.* (14)*, then the improvement $\tilde{S}^{(t)}(\omega)$ is 0. $\bar{x}$ is the solution of the Lasso problem if it is 0. This gives a necessary condition for being a potential candidate.*

Proof. Start with the definition of $\tilde{S}^{(t)}$ in eq. (13). Denoting

$$\tilde{G}(\alpha, \nu) = \begin{pmatrix} G^{(t)} & \nu \\ \nu^* & \alpha \end{pmatrix},$$

we see that the constraint $\nu^* G^{(t),-1} \nu \leqslant \alpha$ ensures that $\tilde{G}(\alpha, \nu) \succeq 0$, hence

$$\sup_{G'} L(G', \Omega_t \cup \{\omega\}) \geqslant \sup_{\alpha, \nu} L(\tilde{G}(\alpha, \nu), \Omega_t \cup \{\omega\}) \tag{17}$$
$$\text{s.t. } \nu^* G^{(t),-1} \nu \leqslant \alpha,$$

and the bound $S^{(t)}(\omega) \geqslant \tilde{S}^{(t)}(\omega)$ in eq. (15) follows.

We now proceed to simplifying the optimization problem. To do that, it is useful to keep in mind that the Fourier coefficients of $g^{(t)}$ are given with the entries of $G^{(t)}$ corresponding to the frequencies in

$$(\omega_j - \omega_i)_{1\leqslant i,j\leqslant t+1} = \begin{pmatrix} 0 & \omega_1 & \dots & \omega_t & \delta_0 = \omega_{t+1} \\ -\omega_1 & 0 & \dots & \omega_t - \omega_1 & \delta_1 = \omega_{t+1} - \omega_1 \\ -\omega_2 & -(\omega_2 - \omega_1) & \ddots & \omega_t - \omega_2 & \delta_2 = \omega_{t+1} - \omega_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -\omega_{t+1} & -(\omega_{t+1} - \omega_1) & \dots & -(\omega_{t+1} - \omega_t) & 0 \end{pmatrix}.$$

We will simplify the definition of L in eq. (8) in proposition 1 with three observations. *First*, recall that the Fourier frequencies of $g^{(t)}$ are given with $\pm\Delta^{(t)}$. Assume we chose ω_{t+1}. Then,

$$\Delta^{(t+1)} = \Delta^{(t)} \cup \left(\bigcup_{i=0}^{t} \delta_i \right), \text{ where } \delta_i = \omega_{t+1} - \omega_i. \tag{18}$$

Hence, $g^{(t+1)}$ has at most $(t+1)$ different Fourier frequencies from $g^{(t)}$. *Second*, since $\omega_{t+1} \neq \pm\omega_i, 0 \leqslant i \leqslant t$, with the definition of the candidate set in eq. (11), we have that the offset of $g^{(t)}$ is given with the trace of $G^{(t)}$, as

$$g_0^{(t)} = \text{Tr } G^{(t)}, \text{ and in particular } g_0^{(t+1)} = g_0^{(t)} + \alpha. \tag{19}$$

Finally, with the same reasoning we have that for any δ_i

$$g_{\delta_i}^{(t+1)} = g_{\delta_i}^{(t)} + \nu_i. \tag{20}$$

Taking the definition of L in eq. (8), splitting the sum over $\Delta^{(t+1)}$ and using the results of eqs. (18) to (20), we get that

$$\begin{aligned} L(\tilde{G}(\alpha,\nu), \Omega_t \cup \{\omega\}) = (f_0 - g_0^{(t)} - \alpha) \\ - 2\sum_{i=0}^{t} \left| f_{\delta_i} - g_{\delta_i}^{(t)} - \nu_i \right| \\ - 2 \sum_{\omega \in \Theta \cup \Delta^{(t)} \setminus \{0\} \setminus \{\pm\delta_i\}_{0\leqslant i\leqslant t}} \left| f_\omega - g_\omega^{(t)} \right| \end{aligned}$$

or put more simply,

$$L(\tilde{G}(\alpha,\nu), \Omega_t \cup \{\omega\}) = L(G^{(t)}, \Omega_t) - \alpha - 2\sum_{i=0}^{t} \left| f_{\delta_i} - \widehat{g}_{\delta_i}^{(t)} - \nu_i \right| - \left| f_{\delta_i} - \widehat{g}_{\delta_i}^{(t)} \right|. \tag{21}$$

To conclude, define $C = 2\sum_{i=0}^{t} \left| f_{\delta_i} - \widehat{g}_{\delta_i}^{(t)} \right|$ and inject the expression of L in eq. (21), into the definition of $\tilde{S}^{(t)}$ in eq. (13). We obtain

$$\tilde{S}^{(t)} = C + \sup_{\alpha,\nu} -\alpha - 2\sum_{i=0}^{t} \left| f_{\delta_i} - \widehat{g}_{\delta_i}^{(t)} - \nu_i \right| \text{ s.t. } \nu^* G^{(t),-1} \nu \leqslant \alpha.$$

Optimizing out α by saturating the constraint $\nu^* G^{(t),-1}\nu \leqslant \alpha$ and denoting $\overline{y} = (\widehat{f}_{\delta_i} - \widehat{g}_{\delta_i}^{(t)})_{0\leqslant i\leqslant t}$, we obtain

$$\tilde{S}^{(t)} = C - 2 \inf_{\nu} \frac{1}{2} \nu^* G^{(t),-1} \nu + \|\nu - \overline{y}\|_1 .$$

Assuming $G^{(t),-1} = T^\top T$ is the Cholesky factorization of $G^{(t),-1}$ and with the change of variables $x = \nu - \overline{y}$, $y = -T\overline{y}$, the definition for $\tilde{S}^{(t)}$ becomes

$$\tilde{S}^{(t)} = C - 2 \inf_{x} \frac{1}{2} \|Tx - y\|_2^2 + \|x\|_1 \tag{22}$$

which completes the proof. □

6.3.2 Greedy-POP in practice

Open Problem 1 (Testing Greedy-POP in practice)**.** *Compare alg. 1 on random trigonometric polynomials to test whether it manages to uncover sparse basis for decomposing a polynomial* f *as a Sum-of-Squares.*

All in all, a procedure for using Greedy-POP is summarized in alg. 1. A precise implementation and extensive testing across random polynomial would help decide if the heuristic of theorem 1 is useful. Note that:

- A backfitting step is costly but certainly necessary in practice.
- It would require experiments to find the fastest solver for the Lasso step. The homotopy method is likely to be the fastest if the solution is very sparse.
- Greedy-POP could easily incorporate previously mentioned sparsity inducing methods.
- Finally, solving multiple instances of the Lasso would be a great speed up. This can be achieved either with ADMM (solving the ridge problem is fast as we have the Cholesky decomposition of G^t), or by considering other proxy for the improvement, *e.g.* using the ℓ_2 norm rather than the F-norm.

Algorithm 1: Greedy-POP

Data: A trigonometric polynomial f
Result: A lower bound on $f_\star$
Initialize $\Omega_0 = \{0\}$, $G^{(0)} = f_0$, $T = f_0^{-1/2}$, $\Delta^{(0)} = \{0\}$;
for $t \in \{1, \dots, T\}$ **do**
  Compute candidate set $\mathcal{C}^{(t)}$ with definition 2;
  /* Best improvement and associated candidate */
  Initialize $\tilde{S}^* = 0$, $\omega_\star = \emptyset$.
  for $\omega \in \mathcal{C}^{(t)}$ **do**
    /* Necessary condition for pos. improvement (see remark 3) */
    $(\overline{x} = g^{(t)}_{\omega-\omega_i} - f_{\omega-\omega_i} = 0) \wedge$ continue;
    Compute $y = T(g^{(t)}_{\omega-\omega_i} - f_{\omega-\omega_i})$;
    /* Proxy for the improvement when using candidate ω */
    $\tilde{S} = 2\left\|T^{-1}y\right\|_1 - 2\textsc{Lasso}(T, y)$;
    /* Update if improvement is better than before */
    $(\tilde{S} \geqslant \tilde{S}^*) \wedge (\tilde{S}^* \leftarrow \tilde{S}) \wedge (\omega_\star \leftarrow \omega)$;
  $\Omega^{(t+1)} = \Omega^{(t)} \cup \{\omega_\star\}$;
  /* Backfitting - solves an SDP of size $(t+1)$ */
  $G^{(t+1)} = \arg\max_{G' \in \mathcal{S}_+(\mathbb{C}^{t+2})} L(G', \Omega^{(t+1)})$
Returns $L(G^{(t+1)}, \Omega^{(t+1)})$.

7 Global Optimization with K-SoS

We now consider how kernel Sum-of-Squares could be useful for global optimization, in manners different from GloptiNets (section 9).

7.1 The quest for certificates

In the following we place significant emphasis on providing methods that output an *a posteriori* certificate of optimality. For a given function f to minimize, we are not satisfied in providing a candidate z for f's minimum with $f(z) \approx f_\star$. As, by definition of the minimum, $f(z) \geqslant f_\star$ for any z, a meaningful certificate for our algorithm should manifest as a *lower bound* on $f_\star$.

The rationale behind seeking such certificates lies in the difference between *a priori* and *a posteriori* guarantees. *A priori* guarantees in global optimization algorithms typically involve convergence rates that, while possibly explicit, depend on the problem's parameters. These parameters are often not practically accessible. In contrast, *a posteriori* guarantees provide a clear, explicit metric to evaluate the algorithm's output quality (even though *e.g.* the algorithm may never terminate). For an in-depth discussion on global optimization in this context, refer to the introduction of [38].

Our focus on *a posteriori* certificates is driven by the inherent challenges of global optimization, a field traditionally plagued by the curse of dimensionality. Indeed, without assumptions on the function to be optimized, random sampling becomes an optimal procedure, yielding an error rate of $O(n^{-1/d})$. However, for smooth functions, we can exploit the objective's smoothness to achieve improved rates of $O(n^{-m/d})$, where m is the number of derivatives of the function. Despite the initial appeal of these rates, as detailed in [27], they conceal a constant factor that is exponential in dimension. This factor is inevitable, as illustrated by a hypothetical function that is 1 almost everywhere, drops to 0 within a small ball, while being infinitely differentiable. For instance, in [27] this constant arise from the uniform inequality from scattered constraints (thm 4).

Therefore, we hope to surpass these exponential constants in practice with *a posteriori* guarantees. We draw inspiration from the success of neural networks in finding effective minima in high-dimensional spaces. Our aspiration is to harness the capabilities of overparameterized models to provide robust *a posteriori* certificates for these findings.

Finally, note that any certificate on f inherently requires some global knowledge of f. A minima, this knowledge is a bound on the norm of the function in an appropriate space; for instance, the worst-case function we described (flat everywhere except on a small region) would have a very large norm. We will typically focus on optimizing trigonometric polynomials or kernel mixtures, with access to their coefficients, which represents a challenging task already.

7.2 The case of periodic functions

We also focus on optimizing periodic functions. There are multiple reasons to that. First of all, they provide an easy analysis with Fourier series. Second, we have polynomial hierarchies as a competitor. Lastly, interesting problems such as the Optimal Power Flow (OPF) [30] or Phase Retrieval [33] involve optimizing such functions.

This was not our first approach though. A failed attempt involved optimizing a function on $\mathbb{R}^d$ with a prior on the localization of the minimum $x_\star$ in the L_2 ball. Then, we would define a mask function μ on $\mathbb{R}^d$ s.t

$$\forall x \in \mathcal{B}_2(0,1),\ \ \mu(x) \geqslant 1,\ \ |\widehat{\mu}(\omega)| \underset{\|\omega\|\to\infty}{=} o(e^{-\|\omega\|}),$$

with the condition on the Fourier Transform ensuring sufficiently smoothness. A certificate would follow from a bound on the L_∞ norm, as first noted in [38, Thm. 2]; applying the mask, we would obtain another bound, but this time from a function whose Fourier Transform would

be well defined. This is illustrated in the following chain of inequality

$$\begin{aligned} f_\star &\geqslant c - \|f - c - g\|_{L_\infty(\mathcal{B}(0,1))} && \text{[38, Thm. 2]} \\ &\geqslant c - \|(f - c - g)\mu\|_{L_\infty(\mathcal{B}(0,1))} && \mu \geqslant 1 \text{ on } \mathcal{B} \\ &\geqslant c - \|(f - c - g)\mu\|_{L_\infty(\mathbb{R}^d)} \\ &\geqslant c - \|(f - c - g)\mu\|_F && \|\cdot\|_\infty \leqslant \|\cdot\|_F. \end{aligned}$$

Unfortunately, this approach was abandoned due to the burden of computing the Fourier Transform of the quantity $(f - g)\mu$, even for simple functions. With the benefit of hindsight, it might prove useful in optimizing Gaussian Mixtures, with a Gaussian Mask $\mu(x) = e^{-\|x\|^2+1}$.

7.3 Space of operators in K-SoS

7.3.1 Diagonal kernel and tensor space

Let $\mathcal{H}$ be a RKHS on a space $\mathcal{X}$, with reproducing kernel K and feature map $\varphi : \mathcal{X} \to \mathcal{H}$. We consider the space of Hermitian operators on $\mathcal{H}$, denoted $\mathcal{S}(\mathcal{H})$. It is a Hilbert space endowed with the Frobenius inner product, with $F \cdot G = \operatorname{Tr} F^* G$, for $F, G \in \mathcal{S}(\mathcal{H})$.

To introduce the concept of Kernel Sum of Squares (K-SoS) to a new audience, it's often stated that K-SoS extends linear models to model positive functions. This extension involves utilizing a positive definite (p.d.) operator G in $\mathcal{S}_+(\mathcal{H})$, and the K-SoS model g is then defined as

$$g : x \mapsto \varphi(x) \cdot G\varphi(x). \tag{23}$$

As highlighted in the introduction (eq. (8)), computing the L_∞ norm is a crucial step in enabling global optimization with certification. This necessity leads to the question: what is the precise Hilbert space in which the function g resides?

The formal answer is that g lives in the RKHS of the product space of $K \times K$, which is the pullback of $\mathcal{H} \otimes \mathcal{H}$ along the diagonal map; see [25], Sec. 5.5. We provide an intuitive way to illustrate this fact. In the following, we identify an operator G in $\mathcal{S}(\mathcal{H})$ with the function $g_G : \mathcal{X} \to \mathbb{R}$ for which $g_G(x) = G \cdot (\varphi(x)\varphi(x)^*)$.

Product space. By rewriting the definition of g in eq. (23), we rewrite the evaluation of g with $g(x) = \langle G, \varphi(x) \otimes \varphi(x) \rangle_{\mathcal{S}(\mathcal{H})}$. This motivates considering a subspace of $\mathcal{S}(\mathcal{H})$, which we denote $\overline{\mathcal{H}}$, defined as

$$\overline{\mathcal{H}} = \overline{\operatorname{Span}\{\varphi(x) \otimes \varphi(x); x \in \mathcal{X}\}}. \tag{24}$$

Let $x, y \in \mathbb{T}^d$. Consider $\varphi(x)\varphi(x)^*, \varphi(y)\varphi(y)^* \in \overline{\mathcal{H}}$. They inherit the Frobenius product of $\mathcal{S}(\mathcal{H})$. Hence, we have

$$\langle \varphi(x)\varphi(x)^*, \varphi(y)\varphi(y)^* \rangle_{\mathcal{S}(\mathcal{H})} = (\varphi(x)^*\varphi(y))^2 = K(x, y)^2,$$

which implies that $\overline{\mathcal{H}}$ is the RKHS associated to the reproducing kernel K^2.

Definition 4 (RKHS of K^2)**.** *We denote $\overline{\mathcal{H}}$the RKHS associated to the reproducing kernel K^2. It is a subspace of $\mathcal{S}(\mathcal{H})$, so we can decompose $\mathcal{S}(\mathcal{H})$ with*

$$\mathcal{S}(\mathcal{H}) = \overline{\mathcal{H}} \oplus \overline{\mathcal{H}}^\perp \tag{25}$$

and use $\Pi_{\overline{\mathcal{H}}}$ (resp. $\Pi_{\overline{\mathcal{H}}^\perp}$) the orthogonal projection on $\overline{\mathcal{H}}$ (resp. $\overline{\mathcal{H}}^\perp$). We will denote with a bar the quantities related to $\overline{\mathcal{H}}$, i.e. *$\overline{\varphi}(x) = \varphi(x)\varphi(x)^*$ and $\overline{K} = K^2$.*

It turns out that from the definition of a K-SoS model in eq. (23), what matters is the projection of G on $\overline{\mathcal{H}}$, as next proposition shows.

Proposition 2 (Norm on $\overline{\mathcal{H}}$ vs. norm on $\mathcal{S}(\mathcal{H})$)**.** *Consider a family of model defined on $\overline{\mathcal{H}}$ with*

$$\forall A \in \mathcal{S}(\mathcal{H}), x \in \mathcal{X}, \quad f_A(x) = \varphi(x) \cdot A\varphi(x) = A \cdot \overline{\varphi}(x),$$

where the first dot product is in $\mathcal{H}$ while the second is in $\mathcal{S}(\mathcal{H})$. For a given A, the representation of f is not unique. Finally, we have

$$\|f_A\|_{\overline{\mathcal{H}}}^2 = \|\Pi_{\overline{\mathcal{H}}} A\|_{\mathcal{S}(\mathcal{H})}^2 = \inf_{\substack{Y \in \mathcal{S}(\mathcal{H}) \\ \Pi_{\overline{\mathcal{H}}} A = \Pi_{\overline{\mathcal{H}}} Y}} \|Y\|_{\mathcal{S}(\mathcal{H})}^2 \leqslant \|A\|_{\mathcal{S}(\mathcal{H})}^2 . \tag{26}$$

Proof. First of all, as $f_A(x) = A \cdot \overline{\varphi}(x)$ (with the inner product in $\overline{\mathcal{H}}$), so f_A belongs to $\overline{\mathcal{H}}$. Secondly,

$$\forall Y \in \overline{\mathcal{H}}^\perp, \;\; f_{A+Y}(x) = (A+Y) \cdot \overline{\varphi}(x) = A \cdot \overline{\varphi}(x) = f_A(x),$$

hence the representation of f_A with $A \in \mathcal{S}(\mathcal{H})$ is not unique. Finally, using the decomposition of eq. (25), we can rewrite $A = U + V$, with $U \in \overline{\mathcal{H}}$ and $V \in \overline{\mathcal{H}}^\perp$. Then,

$$\|f_A\|_{\overline{\mathcal{H}}}^2 = \|U\|_{\overline{\mathcal{H}}}^2 = \inf_{Z \in \overline{\mathcal{H}}^\perp} \|U + Z\|_{\mathcal{S}(\mathcal{H})}^2 \leqslant \|U + V\|_{\mathcal{S}(\mathcal{H})}^2 = \|A\|_{\mathcal{S}(\mathcal{H})}^2 \tag{27}$$

which completes the proof. □

7.3.2 Understanding the dual of K-SoS problems

Following definition 4, we have two spaces $\overline{\mathcal{H}}$ and $\mathcal{S}_+(\mathcal{H})$. Take n anchor points $(z_1, \dots, z_n) \in \mathcal{X}$ and denote $\mathcal{H}_z$) (resp. $\overline{\mathcal{H}}_z$) the span of $\varphi_z = (\varphi(z_1), \dots, \varphi(z_n))$ (resp. $\overline{\varphi}_z = (\overline{\varphi}(z_1), \dots, \overline{\varphi}(z_n))$).

Characterization of $\overline{\mathcal{H}}_z$ and $\mathcal{S}_+(\mathcal{H}_z)$. The space $\overline{\mathcal{H}}_z$ contains mixture of the kernel K^2. $\mathcal{S}_+(\mathcal{H}_z)$ are K-SoS model projected on $\mathcal{H}_z$. That is,

$$\overline{\mathcal{H}}_z = \{\varphi_z \operatorname{Diag} \alpha \varphi_z^*; \alpha \in \mathbb{C}^n\},$$

$$\textit{i.e.} \;\; g \in \overline{\mathcal{H}}_z \iff \exists \alpha \in \mathbb{C}^n, \forall x, g(x) = \sum_{i=1}^n \alpha_i K(z_i, x)^2, \tag{28}$$

$$\text{and} \;\; \mathcal{S}_+(\mathcal{H}_z) = \{\varphi_z G \varphi_z^*; G \in \mathcal{S}_+(\mathbb{C}^n)\},$$

$$\textit{i.e.} \;\; g \in \mathcal{S}_+(\mathcal{H}_z) \iff \exists G \in \mathcal{S}_+(\mathbb{C}^n), \forall x, g(x) = \sum_{ij=1}^n G_{ij} K(z_i, x) K(z_j, x). \tag{29}$$

Both spaces $\overline{\mathcal{H}}$ and $\mathcal{S}_+(\mathcal{H})$ have interesting properties. The former has a smaller HS norm than the latter (see proposition 2), while the latter enforces positivity.

Example 1 (Intersection of $\overline{\mathcal{H}}_z$ and $\mathcal{S}_+(\mathcal{H}_z)$)**.** *We might be interested in having a model with low norm, hence in $\overline{\mathcal{H}}_z$, and positive, hence in $\mathcal{S}(\mathcal{H}_z)$. Unfortunately, the intersection are kernel mixtures in $\overline{\mathcal{H}}$ with positive coefficients,*

$$g \in \overline{\mathcal{H}}_z \cap \mathcal{S}_+(\mathcal{H}_z) \iff \exists \alpha \succeq 0, \;\; g(x) = \sum_{i=1}^n \alpha_i K(z_i, x)^2,$$

which have bad approximation properties [22].

Furthermore, we can project on each of these space, as next proposition shows.

Proposition 3 (Projection on $\overline{\mathcal{H}}_z$ and $\mathcal{S}_+(\mathcal{H}_z)$)**.** *Let $\alpha \in \mathbb{C}^n$ and $G \in \mathcal{S}_+(\mathbb{C}^n)$. Then,*

$$\operatorname{Proj}_{\mathcal{S}_+(\mathcal{H}_z)}(\varphi_z \operatorname{Diag} \alpha \varphi_z^*) = \varphi_z \left[T^{-1} (T \operatorname{Diag} \alpha T^\top)_+ T^{-\top}\right] \varphi_z^* \tag{30}$$

$$\operatorname{Proj}_{\overline{\mathcal{H}}_z}(\varphi_z G \varphi_z^*) = \varphi_z \operatorname{Diag}\left[\overline{K}_z^{-1} \operatorname{Diag}(K_z G K_z)\right] \varphi_z^* \tag{31}$$

with K_z (resp. $\overline{K}_z$) the kernel matrix of $(z_i)_{1 \leqslant i \leqslant n}$ for the kernel K (resp K^2), assumed non-singular, and $T^\top T = K_z$ is the Cholesky decomposition of K_z.

Proof.

Projection on $\mathcal{S}_+(\mathcal{H}_z)$. We want to solve

$$\min_{G \succeq 0} \|\varphi_z \operatorname{Diag} \alpha \varphi_z^* - \varphi_z G \varphi_z^*\|_F^2 .$$

If $T^\top T = K_z$ is the Cholesky factorization of K_z, then $E_z = \varphi_z T^{-1}$ is an orthonormal basis of Span φ_z, hence

$$\|\varphi_z G \varphi_z^* - \varphi_z \operatorname{Diag} \alpha \varphi_z^*\|_F^2 = \left\|E_z(TGT^\top)E_z^* - E_z(T \operatorname{Diag} \alpha T^\top)E_z^*\right\|_F^2 = \left\|TGT^\top - T \operatorname{Diag} \alpha T^\top\right\|_F^2$$

and the solution is $TGT^\top = (T \operatorname{Diag} \alpha T^\top)_+$ hence the result

Projection on $\overline{\mathcal{H}_z}$. We want to solve

$$\min_{\alpha \in \mathbb{C}^n} \|\varphi_z G \varphi_z^* - \varphi_z \operatorname{Diag} \alpha \varphi_z^*\|_F^2 .$$

By expanding the norm, we obtain (as $\varphi_z^* \varphi_z = K_z$)

$$\alpha^* \overline{K}_z \alpha - 2(\operatorname{Diag} \alpha) \cdot (K_z G K_z) = \alpha^* \overline{K}_z \alpha - 2\alpha^* \operatorname{Diag}(K_z G K_z)$$

and setting the gradient to 0 yields the result. □

Reinterpreting dual operations. Those two spaces shed light on the dual of the K-SoS problem. Typically, a K-SoS problem reads

$$\rho_p = \min_{A \in \mathcal{S}_+(\mathbb{R}^n)} L((f_A(x_i))_{1 \leqslant i \leqslant n}) + \frac{\lambda}{2} \|A\|_F^2 , \tag{K-SoS: P}$$

where L would involve *e.g.* fitting some samples (x_i, y_i), with $y_i \geqslant 0$. Equation (K-SoS: P) is a SDP whose dual reads [22]

$$\rho_d = \max_{\alpha \in \mathbb{R}^n} -L^*(\alpha) - \frac{1}{\lambda} \left\|(T \operatorname{Diag} \alpha T^\top)_-\right\|_F^2 , \tag{K-SoS: D}$$

And $\rho_p \geqslant \rho_d$, with equality if we have constraint qualification, *e.g.* there exists one strictly feasible point.

Now, defining g_α as in eq. (28), $g_\alpha = (\varphi_z \operatorname{Diag} \alpha \varphi_z^*) \cdot \varphi(x)\varphi(x)^*$, we have that

$$\begin{aligned} \left\|(T \operatorname{Diag} \alpha T^\top)_-\right\|_F^2 &= \left\|(E_z T \operatorname{Diag} \alpha T^\top E_z)_-\right\|_F^2 && \text{(as } E_z \text{ is orthonormal)} \\ &= \left\|g_\alpha - (E_z T \operatorname{Diag} \alpha T^\top E_z)_+\right\|_F^2 \\ &= d_F(g_\alpha, \mathcal{S}_+(\mathcal{H}_z))^2, \end{aligned} \tag{32}$$

where the last term is the *distance from* g_α *in* $\overline{\mathcal{H}_z}$ *to* $\mathcal{S}_+(\mathcal{H}_z)$. Hence, the dual in eq. (K-SoS: D) can be rewritten

$$\rho_d = \min_{\alpha \in \mathbb{R}^n} L^*(\alpha) \ \text{ s.t. } \ d_F(g_\alpha, \mathcal{S}_+(\mathcal{H}_z)) \leqslant \mu, \tag{33}$$

with μ the dual variable associated to the penalty on the distance. Thus, the dual problem to eq. (K-SoS: P) amounts to finding a mixture in $\overline{\mathcal{H}_z}$ which is not too far from the set of positive operators. A representation is shown on fig. 1.

Example 2 (Interpolation under constraint)**.** *Let* $c \in \mathbb{R}$ *and* $\epsilon > 0$ *some hyperparameters. Let* $x_1, \dots, x_n \in \mathcal{X}$ *and* $y_1, \dots, y_n \in \mathcal{Y}$*. Denote the set of* ϵ*-interpolants of* $y + c$ *with*

$$I_{\epsilon,c} = \left\{g_\alpha \in \overline{\mathcal{H}}; |g_\alpha(x_i) - y_i - c| \leqslant \epsilon\right\} .$$

With eq. (28)*, recall that as* $g_\alpha \in \overline{\mathcal{H}}$*, the requirement* $g_\alpha \in I_{\epsilon,c}$ *translates to*

$$\alpha = \overline{K}_x^{-1}(y + c + \epsilon u) \ \textit{ with } \ \|u\|_\infty \leqslant 1.$$

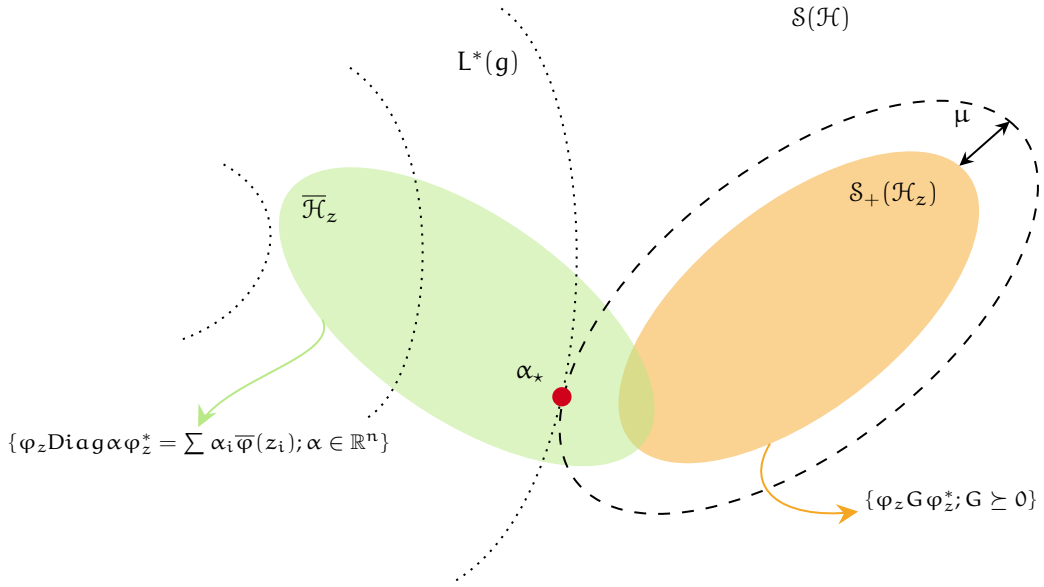


Figure 1: Representation of the dual problem in eq. (K-SoS: D). The green set are the mixtures in $\overline{\mathcal{H}}_z$, the RKHS of kernel K^2. The orange set are the K-SoS models in $\mathcal{H}_z$. The dotted lines are the level set of the loss function L^*. The solution is the minimum of L^* under the constraint of being in $\overline{\mathcal{H}}_z$, at most μ away from $\mathcal{S}_+(\mathcal{H}_z)$.

We are interested if there exists some (c, ϵ) *interpolants which have a positive certificate. This amounts to solve*

$$\delta_{\epsilon,c} = \min_{g \in I_{\epsilon,c}} d_{HS}(g, \mathcal{S}_+(\mathcal{H}_x)) = \min_{\alpha,u \in \mathbb{R}^n} \left\| (T \,\mathrm{Diag}\, \alpha T^\top)_- \right\|_F^2 \quad (34)$$
$$s.t.\ \alpha = \overline{K}_x^{-1}(y + c + \epsilon u),$$
$$\|u\|_\infty \leqslant 1$$

If $\delta_{\epsilon,c} = 0$*, this amounts to solving the interpolation problem in* $\overline{\mathcal{H}}$ *with the constraints* $\alpha \succeq 0$*. But we have additional hindsight here when the interpolation with positive coefficients is not possible.*

7.4 Global optimization with K-SoS

7.4.1 Different approaches for global optimization

This section delves into various approaches for conducting global optimization, linking back to the discussions in section 6.1 which focused on polynomial hierarchies. Our objective here is to delineate the dual problems clearly. By doing so, we aim to understand how applying strengthenings or relaxations can yield a lower bound on $f_\star$, thereby providing the sought-after certificate. Essentially, minimizing f can be recast as finding a probability measure supported on the minima of f, or finding the biggest lower bound of f. We denote $\mathcal{M}(\mathbb{T}^d)$ the set of signed measure on the torus $\mathbb{T}^d$.

Primal. The first possibility – finding a probability measure supported on the minima of f – reads

$$f_\star = \inf \int f d\mu$$
$$\text{s.t. } \mu \in \mathcal{M}(\mathbb{T}^d), \int d\mu = 1, d\mu \geqslant 0.$$

As we did with polynomial hierarchies in problem (POP-P), we relax the constraint on the positivity of μ and consider

$$\rho_p = \inf \int f d\mu \qquad \text{(GO-P)}$$
$$\text{s.t. } \mu \in \mathcal{M}(\mathbb{T}^d), \int d\mu \overset{\text{def.}}{=} \operatorname{Tr} M(\mu) = 1, M(\mu) \overset{\text{def.}}{=} \int \varphi(x)\varphi(x)^* d\mu \succeq 0.$$

As for polynomial, $M(\mu)$ is a p.d. operator of $\mathcal{H}$ containing the moments of the measure μ. Crucially, the relaxation in problem (GO-P) is *tight*, with $\rho_p = f_\star$ [27]. We recover the moment matrix of the polynomial hierarchies by using the polynomial embeddings, *i.e.* when we set $\varphi(x)_\omega = x^\omega$.

Dual. The second point of view –finding the biggest lower bound of f – amounts to writing

$$f_\star = \sup_c c \text{ s.t. } \forall x, f(x) - c \geqslant 0.$$

and again, the following tightening is tight, and dual of problem (GO-P):

$$\rho_d = \sup_{c, G \in \mathcal{S}_+(\mathcal{H})} c \qquad \text{(GO-D)}$$
$$\text{s.t. } \forall x, f(x) - c = \varphi(x)^* G \varphi(x).$$

Whereas in polynomial hierarchies we cannot always write $f - f_\star$ as a sum-of-squares, as the feature map $\varphi(x) = (x^\omega)_{\|\omega\| \leqslant p}$ is not rich enough, a universal kernel's feature map circumvent this issue.

Related work on Global Optimization. In [27], a *relaxation* of the dual – problem (GO-D) – is considered, which no longer gives a lower bound on $f_\star$ but rather an *upper bound*. Specifically, they sample $(x_1, \dots, x_n) \in \mathcal{X}$ and consider the problem

$$f_\star \leqslant \sup_{c, \overline{G} \in \mathcal{S}_+(\mathbb{R}^N)} c \qquad (35)$$
$$\text{s.t. } \forall i \in [1, N], f(x_i) - c = \varphi(x_i)^* G \varphi(x_i)$$

and the representer theorem $G = \varphi_X \overline{G} \varphi_X$ is a by product of the sampling of the inequality, where $\varphi_X = (\varphi(x_1), \dots, \varphi(x_N))$. To compensate for the relaxation of the constraint, they add a penalty term $-\lambda \operatorname{Tr} \overline{G}$ in the objective. This approach will give a poor certificate. Indeed, it requires to bound the norm of the projection on $\operatorname{Span} \overline{\varphi}(x_i)$, which is done with a uniform inequality from scattered constraints (Thm. 4, [27]). This can't escape a constant exponential in the dimension to account for the worst-case scenario which encompasses NP hard problems, as discussed in section 7.1.

In [38], another strategy is used, by replacing the constraint in problem (GO-D) with a penalty with the L_∞ norm. This yields an unconstrained objective, but with untractable L_∞ norm. This is circumvented by using an upper bound of the L_∞ norm, *e.g.* using the ℓ_1 norm of the Fourier coefficients, referred to as the "F-norm". Formally, this amounts to

$$f_\star = \sup_{c, G \in \mathcal{S}_+(\mathcal{H})} c - \|f - c - \varphi(\cdot)^* G \varphi(\cdot)\|_{L_\infty(\mathbb{T}^d)} \geqslant \sup_{c, G \in \mathcal{S}_+(\mathcal{H})} c - \|f - c - \varphi(\cdot)^* G \varphi(\cdot)\|_F \qquad (36)$$

While retaining the property of having a lower bound on $f_\star$ (hence a certificate), computing the certificate with target accuracy ϵ scale exponentially with the dimension, requiring $O(\epsilon^{-d})$ compute time.

New approaches to Global Optimization with K-SoS. In the next section, we discuss two technical problems which, once solved, could greatly streamline global optimization of polynomials with kernel Sum-of-Squares.

7.4.2 Computing the $\overline{\mathcal{H}}$ norm of a K-SoS Model

In light of eq. (36), a pertinent question arises: "Why not employ the RKHS norm of $\overline{\mathcal{H}}$ to upper bound the L_∞ norm?" Given that the kernel is bounded, the RKHS norm emerges as a logical upper bound for the L_∞ norm. The K-SoS model, as explained in section 7.3.2 and eq. (29), writes as

$$g(x) = \sum_{i,j=1}^{m} \overline{G}_{ij} K(z_i, x) K(z_j, x). \tag{37}$$

Here, the norm of $G = \varphi_z \overline{G} \varphi_z^*$ can be readily calculated within $\mathcal{S}(\mathcal{H})$, offering an upper bound on $\|G\|_{\overline{\mathcal{H}}}$ and consequently on $\|g\|_{\overline{\mathcal{H}}}$. However, there's often a significant disparity between these two norms.

Computing the norm of g in $\overline{\mathcal{H}}$ can be achieved with some kernel stable by multiplication. For instance, the Gaussian kernel allows the representation of a product of RKHS terms associated with $e^{-\cdot/2s}$ as elements within a larger RKHS linked to $e^{-\cdot/s}$. A challenge arises when the K-SoS model contains m anchor points, as in eq. (37): the resulting mixture comprises m^2 terms. Consequently, calculating the RKHS norm in $\overline{\mathcal{H}}$ becomes a process with a time complexity of $O(m^4)$, rendering it impractical for large-scale applications. While simple in appearance, this problem is one of the main reason for the design of the GloptiNets algorithm (remark 5 in section 9).

Open Problem 2 (Computing a norm in $\overline{\mathcal{H}}$ for a K-SoS model)**.** *Find a reproducing kernel on $\mathbb{T}^d$ for which the norm of a K-SoS model in $\overline{\mathcal{H}}$ can be efficiently computed. Then, optimize directly $\sup_{c,g} c - \|f - c - g\|_{\overline{\mathcal{H}}}$.*

7.4.3 Relaxation with Nyström projections

Despite being omnipresent in polynomial optimization, the moment point of view is barely used for performing global optimization with K-SoS. Here we consider another alternative relaxation for problem (GO-P) and describe its dual.

Notations. We use the same notations as in section 7.3.2. Given $X = (x_1, \dots, x_N) \in \mathbb{T}^{dn}$,

$$\begin{aligned} \varphi_X &= (\varphi(x_1), \dots, \varphi(x_N)), \\ \varphi_X^* \varphi_X &= K_X = T^\top T, \\ E_X &= \varphi_X T^{-1}, \\ P_X &= E_X E_X^* = \varphi_X K_X^{-1} \varphi_X^*, \\ \Pi_X A &= P_X A P_X, \ \ \forall A : \mathcal{H} \to \mathcal{H}. \end{aligned} \tag{38}$$

K is the kernel matrix, T its Cholesky decomposition, E_X contains in its column an orthonormal basis of Span φ_X, hence $E_X^* E_X = \mathbf{I}_N$ and $P_X : \mathcal{H} \to \mathcal{H}$ is the orthogonal projection operator on Span φ_X, *i.e.* $P_X^2 = P_X$ and $P_X \varphi(x_i) = \varphi(x_i)$ for all $i \in 1 : N$. Finally, Π_X is also a projector but this time on the space $\mathcal{S}(\mathcal{H})$.

We focus on polynomial optimization. A trigonometric polynomial f can be represented as a quadratic form in $\mathcal{H}$, *i.e.*

$$f(x) = \varphi(x) \cdot F\varphi(x) \tag{39}$$

for some Hermitian operator F of $\mathcal{S}(\mathcal{H})$. Such operator can be established in closed-form with an explicit characterization of φ. For instance, if K is a shift invariant kernel, $K(x, y) = q(x - y)$. We then take the Fourier Transform of q, which is positive everywhere because K is a r.k. Then, we write

$$K(x, y) = \sum_{\omega \in \mathbb{Z}^d} \widehat{q}_\omega e^{2\pi i \omega \cdot (x-y)} \overset{\text{def.}}{=} \varphi(x) \cdot \varphi(y) \ \text{ with } \ q(x) = (\sqrt{\widehat{q}_\omega} e^{2\pi i \omega \cdot x})_{\omega \in \mathbb{Z}^d}$$

and we obtain an explicit embedding for $\varphi : \mathbb{T}^d \to \mathcal{H}$. Armed with this embedding, we can build a $F \in \mathcal{S}(\mathcal{H})$ s.t eq. (39) holds. Again, recall proposition 2 which states that such definition is not unique. With eq. (39), applying a measure μ to f can be written

$$\int_{\mathbb{T}^d} f(x) d\mu(x) = F \cdot \int_{\mathbb{T}^d} \varphi(x) \varphi(x)^* d\mu(x) = F \cdot M(\mu) \tag{40}$$

by definition of the moment operator $M(\mu)$ given in problem (GO-P). This is a direct parallel with polynomial hierarchies, where the action of a measure μ on a polynomial f is seen through its moments $y_\omega = \int x^\omega d\mu$, as $\int f d\mu = \int \sum f_\omega x^\omega d\mu = (f_\omega) \cdot (y_\omega)$.

Relaxation of the primal. We relax problem (GO-P) with

$$\begin{aligned} f_\star \geqslant \rho_p = \inf \ & F \cdot M(\mu) \\ \text{s.t. } & \mu \in \mathcal{M}(\mathbb{T}^d), \operatorname{Tr} M(\mu) = 1, \Pi_X M(\mu) \succeq 0, \end{aligned} \tag{Nys-P}$$

where we relaxed the condition $M(\mu) \succeq 0$ to $\Pi_X M(\mu) \succeq 0$. It is equivalent to relaxing $d\mu \geqslant 0$ to

$$\forall \alpha \in \mathbb{C}^N, \ \int \left| \sum_{i=1}^N \alpha_i K(x_i, x) \right|^2 d\mu(x) \geqslant 0.$$

In other words, we relax the condition on μ being positive everywhere to being positive on sum-of-square of kernel mixtures.

Strengthening of the dual The dual of problem (Nys-P) is a *strengthening* of problem (GO-D) and reads

$$\begin{aligned} \rho_d = \sup_{c, \overline{G} \in \mathcal{S}_+(\mathbb{R}^N)} \ & c \\ \text{s.t. } & \forall x, f(x) - c = \varphi(x)^* G \varphi(x), \\ & G = \varphi_X^* \overline{G} \varphi_X. \end{aligned} \tag{Nys-D}$$

This expression shows that problem (Nys-D) might be unfeasible. To circumvent that, the constraint could be *relaxed* to $|f(x) - c - g(x)| \leqslant \lambda$ for some $\lambda > 0$. Then ρ is no longer a lower bound on $f_\star$, but $\rho - \lambda$ is.

Difficulties. The main difficulty lies in representing the space $\Pi_X M(\mu)$. It is obviously a subset of $\mathcal{S}_+(\mathbb{R}^n)$. Furthermore, we have *e.g.* that for all $M \in \Pi_X M(\mu)$,

$$|M_i j| = \left| \int K(x_i, x) K(x_j, x) d\mu \right| \leqslant 1$$

so that $\|M\|_{L_\infty} \leqslant 1$. However, a more precise characterization of $\Pi_X M(\mu)$ is necessary to obtain meaningful result from problem (Nys-P).

Open Problem 3. *Representing projection of moments Let $X = (x_1, \ldots, x_n) \in \mathbb{T}^d$. For a r.k K, represent the sets*

$$(b, S) \in \left\{ \left(\int K_N(x) d\mu, \int K_N(x) K_N(x)^\top d\mu(x) \right), \mu \in \mathcal{M}(\mathcal{X}) \right\}$$

where $K_N(x) = K(x_i, x)_{1 \leqslant i \leqslant N}$. They are subsets of e.g. $[\mathbb{R}^n, \mathcal{S}_+(\mathbb{R}^n)] \cap B_\infty$ *where B_∞ is the unit ball for the infinity norm. Then, solve problem* (Nys-P)*.*

8 Beyond K-SoS & Open problems

Before covering the GloptiNets algorithm extensively in section 9, we highlight three open problems on global optimization which we deem of particular interest.

K-SoS for Gaussian Mixture. Optimizing Gaussian Mixtures warrants special focus due to their widespread use in machine learning, such as in kernel ridge regression. An algorithm capable of robustly optimizing Gaussian Mixtures holds the potential to optimize any black box function, by optimizing the results of a kernel ridge regression fitted on samples drawn from the black box function. Furthermore, integrating the algorithm's certificate with statistical assumptions pertaining to the black box function can enhance the reliability and validity of the optimization results.

Furthermore, a natural domain of application would be to optimize the acquisition function in Bayesian Optimization (BO) [10]. In BO, a Gaussian Process is fitted to sample data from a black box function, effectively creating a Gaussian mixture. The aim is to maximize the acquisition function, a sum of the Gaussian Process and a function that encourages exploration. Often, this acquisition function takes the form of a Gaussian mixture and is non-convex. Current results in Bayesian Optimization typically rely on an oracle to determine the maximum of this acquisition function. An algorithm that can certify the attainment of this maximum would mark a significant breakthrough in the field.

Open Problem 4 (K-SoS for Bayesian optimization)**.** *Using the Gaussian kernel for a K-SoS model adapted to optimize Gaussian mixture, optimize globally the acquisition function of a Bayesian optimization problem.*

Certificates for faster optimization. Beyond merely lending credibility to the results and making the algorithm suitable for high-stakes applications, certificates can also accelerate the optimization of non-convex objectives. An algorithm that provides a certificate on the minimum of a function has to approximate uniformly the entire function across its domain. However, in the context of minimization, it's unnecessary to achieve precise optimization or uniform approximation in areas where the function's values are significantly high. Therefore, the domain can be divided into several subdomains, corresponding *e.g.* to the number of available processing cores. Each segment undergoes optimization with certified upper and lower bounds on the function values within that segment. Optimization efforts can be halted in segments where the lower bound exceeds the upper bound of another segment. This allows computational resources to be reallocated to refine other domain segments, thereby streamlining the entire optimization process. Figure 2 provides an example of this procedure, detailed in section C.1.2 with alg. 3.

Open Problem 5 (Certificates for faster optimization)**.** *Given an algorithm which provides a certificate on $f_\star$ which is gradually refined, implement a splitting scheme such as alg. 3 to accelerate optimization.*

Global optimization with normalizing flows. Modeling positive functions is crucial in global optimization, as highlighted in the dual problem (GO-D), which involves modeling $f - c$ as a positive function. Similarly, the primal problem (GO-P) seeks a probability distribution concentrated on the minima of f. While K-SoS serves as an effective positive model, exploring alternative methods like normalizing flows is a natural research direction.

Normalizing flows, a subset of Neural Ordinary Differential Equations (Neural ODEs), offer a novel approach to modeling positive functions [7]. They define the evolution of a state $z(t) \in \mathbb{R}^d$ through a neural network $h_\theta : \mathbb{R}^d \times t \to \mathbb{R}^d$ with parameters $\theta \in \mathbb{R}^p$:

$$z(0) = z_0 \in \mathbb{R}^d, \ \ \forall t \in (0,1), \ \partial_t z(t) = h_\theta(z(t), t). \tag{41}$$

Neural Ordinary Differential Equations (Neural ODEs) have been applied across a diverse range of fields, particularly in probability density modeling, which is our focus in this section. A key advantage of Neural ODEs is their ability to create a bijective mapping between inputs and outputs. This is possible as long as the function h_θ remains Lipschitz continuous in its

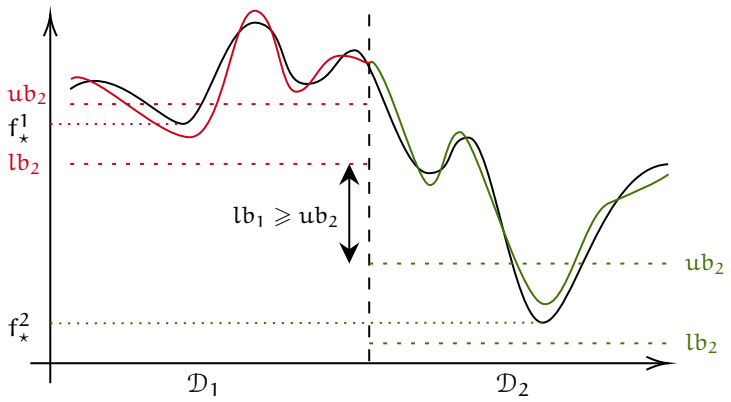


Figure 2: Certificates can be used to accelerate optimization, by discarding unimportant part of the domain. First, the domain is split in multiple parts $(\mathcal{D}_1, \mathcal{D}_2)$. Then, an algorithm is used to optimize each part, with lower and upper bound $(lb_i, ub_i)_i$ on the optimum $f^i_\star$ of the domain. This requires approximating uniformly the function *(red, green curve)*. As soon as $lb_i \geqslant ub_j$ for some j (here, $(i, j) = (1, 2)$), the domain $\mathcal{D}_i$ can be discarded, and the computational resource reallocated to splitting the domain $\mathcal{D}_j$.

first variable for all t, in accordance with the Cauchy-Lipshitz theorem or Picard's existence theorem[3]. Such properties make applying the change of variable formula feasible.

A significant technical advancement made in [7] is the introduction of a continuous backpropagation rule. For a given function $L : \mathbb{R}^d \to \mathbb{R}$, this allows for the computation of gradients of the function $(\theta, z_0) \mapsto L(z_1)$, with computational costs similar to evaluating L itself. Ultimately, Neural ODEs provide a mapping from $\mathbb{R}^d$ to $\mathbb{R}^d$. By initially sampling z_0 from a standard normal distribution $\mathcal{N}(0, \mathbf{I}_d)$ and examining the distribution of z_1, we can track the Jacobian's trace along the transformation Tr $\partial_z h_\theta(z_t)$. This process turns the distribution of z_1 into a probability distribution. The FFJORD algorithm, as cited in [12], employs a stochastic estimator of this trace, further enhancing the efficiency of Neural ODEs in modeling probability distributions.

For global optimization, where we seek a probability distribution μ supported on f's minimizer (problem (GO-P)), normalizing flows are an interesting candidate. We define μ as the distribution of z_1 given by eq. (41) with $z_0 \sim \mathcal{N}(0, \mathbf{I})$. The optimization objective is thus formulated as:

$$F(\theta) = \mathbb{E}_{z_0 \sim \mathcal{N}(0,\mathbf{I})}[h(z_1)], \tag{42}$$

and its gradient is estimated unbiasedly by

$$\nabla F(\theta) = \mathbb{E}_{z_0 \sim \mathcal{N}(0,\mathbf{I})}[\partial_\theta h(z_1(z_0, \theta))]. \tag{43}$$

This setup allows for black-box optimization, provided that h_θ's gradient is available for continuous backpropagation. The architecture of h_θ and the chosen optimization algorithm are the primary hyperparameters. While the architecture should permit a Dirac distribution for a definitive minimizer, obtaining the global minimizer in expectation is also feasible, with $x_\star = \mathbb{E}_{z_0}[z_1]$.

Open Problem 6 (Normalizing flows for Global Optimization)**.** *Explore the theoretical properties of normalizing flows for global optimization. Test whether there are competitive with Bayesian optimization for black-box optimization.*

[3]These theorems ensure the injectivity of the function surjectivity is achieved by reversing the flow of time and considering the evolution with $-h_\theta$, which also maintains Lipschitz continuity.

9 GloptiNets: Scalable Non-Convex Optimization with Certificates

Abstract

We present a novel approach to non-convex optimization with certificates, which handles smooth functions on the hypercube or on the torus. Unlike traditional methods that rely on algebraic properties, our algorithm exploits the regularity of the target function intrinsic in the decay of its Fourier spectrum. By defining a tractable family of models, we allow *at the same time* to obtain precise certificates and to leverage the advanced and powerful computational techniques developed to optimize neural networks. In this way the scalability of our approach is naturally enhanced by parallel computing with GPUs. Our approach, when applied to the case of polynomials of moderate dimensions but with thousands of coefficients, outperforms the state-of-the-art optimization methods with certificates, as the ones based on Lasserre's hierarchy, addressing problems intractable for the competitors.

9.1 Introduction

Non-convex optimization is a difficult and crucial task. In this section, we aim at optimizing globally a non-convex function defined on the hypercube, by providing a certificate of optimality on the resulting solution. Let h be a smooth function on $[-1,1]^d$. Here we provide an algorithm that given $\widehat{x}$, an estimate of the minimizer $x_\star$ of h

$$x_\star = \operatorname*{arg\,min}_{x\in[-1,1]^d} h(x),$$

produces an ϵ, that constitutes an explicit *certificate* for the quality of $\widehat{x}$, of the form

$$|h(x_\star) - h(\widehat{x})| \leqslant \epsilon_\delta,$$

with probability $1-\delta$. The literature abounds of algorithms to optimize non-convex functions. Typically they are either *(a)* heuristics, very smart, but with no guarantees of global convergence [23, 15] *(b)* variation of algorithms used in convex optimization, which can guarantee convergence only to *local* minima [6] *(c)* algorithms with only asymptotic guarantees of convergence to a global minimum, but no explicit certificates [31]. In general, the methods recalled above are quite fast to produce some solution, but don't provide guarantees on its quality, with the result that the produced point can be arbitrarily far from the optimum, so they are used typically where non-reliable results can be accepted.

On the contrary, there are contexts where an explicit quantification of the incurred error is crucial for the task at hand (finance, engineering, scientific validation, safety-critical scenarios [18]). In these cases, more expensive methods that provide certificates are used, such as *polynomial sum-of-squares* (poly-SoS) [17, 18]. These kinds of techniques are quite powerful since they provide certificates in the form above, often with machine-precision error. However, *(a)* they have reduced applicability since h must be a multivariate polynomial (possibly sparse, low-degree) and must be known in its analytical form *(b)* the resulting algorithm is a semi-definite programming optimization on matrices whose size grows very fast with the number of variables and the degree of the polynomial, becoming intractable already in moderate dimensions and degrees.

Our approach builds and extends the more recent line of works on *kernel sum-of-squares*, and in particular the work of Woodworth et al. [38] based on the Fourier analysis. It mitigates the limitations of poly-SoS methods in both aspects: *(a)* we can deal with any function h (not necessarily a polynomial) for which the Fourier transform is known and *(b)* the resulting algorithm leverages the smoothness properties of the objective function as [38] rather than relying on its algebraic structure leading to way more compact representations than poly-SoS. Contrary to [38], we fully leverage the power of the certificate allowing for a drastic reduction of the computational cost of the method. Indeed, we cast the minimization in terms of a way smaller problem, similar to the optimization of a small neural network that, albeit again non-convex, produces efficiently a good candidate on which we then compute the certificate.

Notably, our focus lies on a posteriori guarantees: we define a family of models that allow for efficient computation of certificates. Once the model structure is established, we have ample flexibility in training the model, offering various possibilities to achieve good certificates in practical scenarios, while still using well-established and effective techniques in the field of deep neural networks (DNN) [11] to reduce the computational burden of the approach.

Our contributions can be summarized as follows:

- We propose a new approach to global optimization *with certificates* which drastically extends the applicability domain allowed by the state of the art, since it can be applied to any function for which we can compute the Fourier transform (not just polynomials).
- The proposed approach is naturally tailored for GPU computations and provides a refined control of time and memory requirements of the proposed algorithm, contrary to poly-SoS methods (whose complexity scales dramatically and in a rigid way with dimension and degree of the polynomial).
- From a technical viewpoint, we improve the results in [38], by developing a fast stochastic approach to recover the certificate in high probability (theorem 4), and we generalize the formulation of the problem to allow the use of powerful techniques from DNN, still providing a certificate on the result (section 9.3, in particular alg. 2)
- In practical applications, we are able to provide certificates for functions in moderate dimensions, which surpasses the capabilities of current state-of-the-art techniques. Specifically, as shown in the experiments we can handle polynomials with thousands of coefficients. This achievement marks an important milestone towards utilizing these models to provide certificates for more challenging real-life problems.

9.1.1 Previous work

Polynomial SoS. In the field of certificate-based polynomial optimization, Lasserre's hierarchy plays a pivotal role [17, 18]. This hierarchy employs a sequence of SDP relaxations with increasing size proportional to $O(r^d)$ (where d is the dimension of the space and r is a parameter that upper bounds the degree of the polynomial) and that ultimately converges to the optimal solution when $r \to \infty$. While Lasserre's hierarchy is primarily associated with polynomial optimization, its applicability extends beyond this domain. It offers a specific formulation for the more general moment problem, enabling a wide range of applications; see [13] for an introduction. For polynomial optimization problems such as in eq. (44), a significant amount of research has been dedicated to leveraging problem structure to improve the scalability of the hierarchy. This research has predominantly focused on exploiting very specific sparsity patterns among the variables of the polynomial, enabling the handling in these restricted scenarios of instances ranging from a few variables to even thousands of variables [32, 36, 35]. There have been theoretical results regarding optimization on the hypercube [1, 19], but there are no algorithms handling them natively. Furthermore, alternative approaches exist that exploit different types of structure, such as the constant trace property [20].

Kernel SoS. Kernel Sum of Squares (K-SoS) is an emerging research field that originated from the introduction of a novel parametrization for positive functions in [22]. This approach has found application in various domains, including Optimal Control [3], Optimal Transport [24] and modeling probability distribution [26]. In the context of function optimization, two types of theoretical results have been explored: *a priori* guarantees [27] and *a posteriori* guarantees [38]. A priori guarantees offer insights into the convergence rate towards a global optimum of the function, giving a rate on the number of parameters and the complexity necessary to optimize a function up to a given error. For example, [27] proposes a general technique to achieve the global optimum, with error ϵ of a function that is s-times differentiable, by requiring a number of parameters essentially in the order of $O(\epsilon^{-s/d})$, allowing to avoid the curse of dimensionality in the rate, when the function is very regular, i.e., $s \geqslant d$, while typical black-box optimization algorithms have a complexity that scales as ϵ^{-d}. A-posteriori guarantees focus on providing a certificate for the minimum found by the algorithm. In particular, [38], provides both a-priori guarantee and a-posteriori certificates; however, the model considered makes it computationally infeasible to provide certificates in dimension larger than 2.

To conclude, approaches based on kernel-SoS allow to extend the applicability of global optimization with certificates methods to a wider family of functions and on exploiting finer regularity properties beyond just the number of variables and the degrees of a polynomial. By comparison, we focus on making the optimization amenable to high-performance GPU computation while retaining an a posteriori certificate of optimality.

9.2 Computing certificates with extended k-SoS

Without loss of generality (see next remark), with the goal of simplifying the analysis and using powerful tools from harmonic analysis, we cast the problem in terms of minimization of a *periodic* function f over the torus, $[0,1]^d$ (we will denote it also as $\mathbb{T}^d$). In particular, we are interested in minimizing periodic functions for which we know (or we can easily compute) the coefficients of its Fourier representation, i.e.

$$f_\star = \min_{z\in\mathbb{T}^d} f(z), \qquad f(z) = \sum_{\omega\in\mathbb{Z}^d} \widehat{f}_\omega e^{2\pi i\omega\cdot z}, \quad \forall z \in \mathbb{T}^d, \tag{44}$$

where $\mathbb{Z}$ is the set of integers. This setting is already interesting on its own, as it encompasses a large class of smooth functions. It includes notably trigonometric polynomials, *i.e.* functions which have only a finite number of non-zero Fourier coefficients $\widehat{f}_\omega$. Optimization of trigonometric polynomials arises in multiple research areas, such as the optimal power flow [30] or quantum mechanics [14]. Note that this problem is already NP-hard, as it encompasses for instance the Max-Cut problem [33]. Even so, we will consider the more general case where we can evaluate function values of f, along with its Fourier coefficient $\widehat{f}_\omega$, and we have access to its norm in a certain Hilbert space. This norm can be computed numerically for trigonometric polynomials, and more generally reflects the regularity (degree of differentiability) of the function, and thus the difficulty of the problem.

Remark 4 (No loss of generality in working on the torus)**.** *Given a (non-periodic) function* $h : [-1,1]^d \to \mathbb{R}$ *we can obtain a periodic function whose minimum is exactly* h_* *and from which we can recover* $x_\star$*. Indeed, following the classical Chebychev construction, define* $\cos(2\pi z)$ *as the componentwise application of* cos *to the elements of* $2\pi z$*, i.e.* $\cos(2\pi z) := (\cos(2\pi z_1), \dots, \cos(2\pi z_d))$ *and define* f *as* $f(z) := h(\cos(2\pi z))$ *for* $z \in [0,1]^d$*. It is immediate to see that* (a) f *is periodic, and,* (b) *since* $\cos(2\pi z)$ *is invertible on* $[0,1]^d$ *and its image is exactly* $[-1,1]^d$*, we have* $h_* = h(x_\star) = f(z_\star)$ *where*

$$x_\star = \cos(2\pi z_\star), \quad \textit{and} \quad z_\star = \min_{z\in\mathbb{T}^d} f(z).$$

We discuss an efficient representation of these problems in section 9.3.3.

9.2.1 Certificates for global optimization and k-SoS

A general "recipe" for obtaining a certificates was developed in [38] where, in particular, it was derived the following bound [38, see Thm. 2]

$$f_\star \geqslant \sup_{c\in\mathbb{R},\ g\in\mathcal{G}_+} c - \|f - c - g\|_F \ , \tag{45}$$

where $\|u\|_F$ is the ℓ_1 norm of the Fourier coefficients of a periodic function u, i.e.

$$\|u\|_F := \sum_{\omega\in\mathbb{Z}^d} |\widehat{u}_\omega| , \tag{46}$$

and the sup is taken over $\mathcal{G}_+$ that is a class of non-negative functions. The paper [38] then chooses $\mathcal{G}_+$ to be the set of *positive semidefinite models*, leading to a possibly expensive convex SDP problem. Our approach instead starts from the following two observations: *(a)* the lower bound in eq. (45) holds for any set $\mathcal{G}_+$ of non-negative functions, *not necessarily convex*, moreover *(b)* any candidate solution (g, c) of the supremum in eq. (45) would constitute a lower bound for $f_\star$, so there is no need to solve eq. (45) exactly. This yields the following theorem

Theorem 2. *Given a point* $\widehat{x} \in \mathbb{T}^d$ *and a non-negative and periodic function* $g_0 : \mathbb{T}^d \to \mathbb{R}_+$*, we have*

$$|f(\widehat{x}) - f(x_\star)| \leqslant \|f - f(\widehat{x}) - g_0\|_F \tag{47}$$

Proof. Since $x_\star$ is the minimizer of f, then $f(x_\star) \leqslant f(\widehat{x})$. Moreover, since $c_0 = f(\widehat{x})$ and g_0 are feasible solutions for the r.h.s. of eq. (45), we have

$$f(\widehat{x}) \geqslant f(x_\star) \geqslant \sup_{c\in\mathbb{R},\ g\in\mathcal{G}_+} c - \|f - c - g\|_F \geqslant c_0 - \|f - c_0 - g_0\|_F\,,$$

from which we derive that $0 \leqslant f(\widehat{x}) - f(x_\star) \leqslant \|f - f(\widehat{x}) - g_0\|_F$. □

In particular, since any good candidate g_0 is enough to produce a certificate, we consider the following class of non-negative functions that can be seen as a *two-layer neural network*.

Definition 5 (extended K-SoS model on the torus)**.** *Let $K : \mathbb{T}^d \times \mathbb{T}^d \to \mathbb{R}$ be a periodic function in the first variable and let $m, r \in \mathbb{N}$. Given a set of anchors $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_m) \subset \mathbb{T}^d$ and a matrix $R \in \mathbb{R}^{r\times m}$, we define the K-SoS model g with*

$$\forall \mathbf{x} \in \mathbb{T}^d, \quad g(\mathbf{x}) = \|RK_{\mathbf{Z}}(\mathbf{x})\|_2^2, \quad \textit{and} \quad K_{\mathbf{Z}}(\mathbf{x}) = (K(\mathbf{x}, \mathbf{z}_1), \dots, K(\mathbf{x}, \mathbf{z}_m)) \in \mathbb{R}^m. \tag{48}$$

The functions represented by the model above are non-negative and periodic. The model is an extension of the k-SoS model presented in [22], where the points $(\mathbf{z}_1, \dots, \mathbf{z}_m)$ cannot be optimized. Moreover it has the following benefits at the expense of the convexity in the parameters:

1. The extended k-SoS models benefit of the good approximation properties of k-SoS models described in [22] and especially [26], since they are a super-set of the k-SoS, that have optimal approximation properties for non-negative functions.
2. The extended model can have a *reduced number of parameters*, by choosing a matrix R with $r = 1$ or $r \ll m$. This will drastically improve the cost of the optimization, while not impacting the approximation properties of the model, since a good approximation is still possible with already r proportional to d [27, see Thm. 3].
3. The extended model *does not require any positive semidefinite constraint* on the matrix (contrary to the base model) that is typically a well-known bottleneck to scale up the optimization in the number of parameters [22]. In the extended model we trade the positive semidefinite constraint with non-convexity. However this allows us to use all the advanced and effective techniques we know for unconstrained (or box-constrained) non-convex optimization for (two-layers) neural networks [11].

To conclude the picture on the k-SoS models, a critical aspect of the model is the choice of K, since it must guarantee good approximation properties and at the same time we need to compute easily its Fourier coefficients since we need to evaluate $\|f - c - g\|_F$. To this aim, a good candidate for K are the *reproducing kernels* defined on the torus [29]. We use shift-invariant kernels, enabling a convenient analysis of the associated RKHS through their Fourier Transform.

Definition 6 (Reproducing kernel on the torus)**.** *Let q be a real function on $\mathbb{T}^d$, with positive Fourier Transform and $q(0) = 1$. Let K be the kernel defined with*

$$\forall x, y \in \mathbb{T}^d, \ \ K(x, y) = q(x - y) = \sum_{\omega\in\mathbb{Z}^d} \widehat{q}_\omega e^{2\pi i\omega\cdot(x-y)}. \tag{49}$$

Then, K is a r.k bounded by 1. We denote $\mathcal{H}$ its Reproducing kernel Hilbert Space (RKHS) and by $\|\cdot\|_{\mathcal{H}}$ the associated RKHS norm

$$\|f\|_{\mathcal{H}}^2 = \sum_{\omega\in\mathbb{Z}^d} |\widehat{f}_\omega|^2/\widehat{q}_\omega.$$

Define $\lambda(x) = q(x)^2$. We assume that we can compute *(and* sample *from, see next section) $\widehat{\lambda}_\omega$, i.e., the Fourier transform of λ, corresponding to $(\widehat{q} \star \widehat{q})_\omega$, for all $\omega \in \mathbb{Z}^d$.*

By choosing such a K, the models inherit the good approximation properties derived in [22, 26]. We conclude by recalling that shift-invariant r.k kernel have a positive Fourier transform due to Bochner's theorem [28]. The fact that K is bounded by 1 can be seen with $|K(x, x)| = |q(0)| = \sum_\omega \widehat{q}_\omega = 1$. Finally, note that the Fourier coefficients of an extended k-SoS model can be computed exactly, as in shown *e.g.* later in lemma 1.

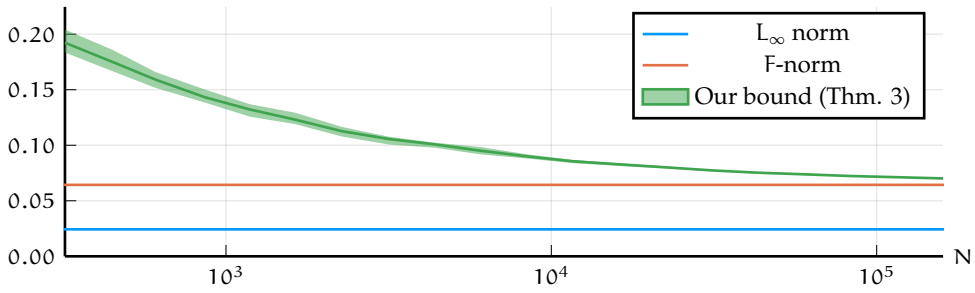


Figure 3: $f - f_\star$ is a trigonometric polynomial approximated by an extended k-SoS model g. The L_∞ norm of the difference *(blue)* is upper-bounded by the F-norm *(red)*, which is itself upper bounded by the MoM inequality in theorem 4, with probability 98%, here showed with respect to the number N of sampled frequencies. Shaded area shows min/max values across 10 estimations.

9.2.2 Providing certificates with the F-norm

As discussed in the previous section our approach for providing a certificate on f relies on first obtaining $\widehat{x}$ using a fast algorithm without guarantees and solving approximately eq. (45) to obtain the certificate (see theorem 2). With this aim, now we need an efficient way to compute the norm $\|\cdot\|_F$. We use here a stochastic approach. Introducing a probability $\widehat{\lambda}_\omega$ (that later will be chosen as a rescaled version of $\widehat{\lambda}_\omega$ in definition 6) on $\mathbb{Z}^d$ we rewrite the F-norm

$$\|u\|_F = \sum_{\omega\in\mathbb{Z}^d} \widehat{\lambda}_\omega \cdot \frac{|\widehat{u}_\omega|}{\widehat{\lambda}_\omega} = \mathbb{E}_{\omega\sim\widehat{\lambda}_\omega}\left[\frac{|\widehat{u}_\omega|}{\widehat{\lambda}_\omega}\right] \tag{50}$$

which yields an objective that is amenable to stochastic optimization. From there, [38] computes a certificate by truncating the sum to a hypercube $\{\omega; \|\omega\|_\infty \leqslant N\}$ of size N^d and bounding the remaining terms with a smoothness assumption on $u = f - c - g$, which enables to control the decay of $\widehat{u}_\omega$. We want to avoid this cost exponential in the dimension so we proceed differently.

Probabilistic estimates with the $\mathcal{H}$ norm. Given that the F-norm can be written as an expectation in eq. (50), we approximate it with an empirical mean $\widehat{S}$ given with N i.i.d samples $(\omega_i)_{1\leqslant i\leqslant n} \sim \widehat{\lambda}_\omega$. Now, note that the variance of $\widehat{S}$ can be upper bounded by a Hilbert norm, as

$$\widehat{S} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\widehat{u}_{\omega_i}|}{\lambda_{\omega_i}}, \quad \text{so that} \quad \operatorname{Var}\widehat{S} \leqslant \frac{1}{N}\mathbb{E}\left(\frac{|\widehat{u}_\omega|}{\widehat{\lambda}_\omega}\right)^2 = \frac{1}{N}\sum_{\omega\in\mathbb{Z}^d}\frac{|\widehat{u}_\omega|^2}{\widehat{\lambda}_\omega} = \frac{1}{N}\|u\|^2_{\mathcal{H}_\lambda}, \tag{51}$$

with $\mathcal{H}_\lambda$ the RKHS from definition 6 with kernel $K(x,x') = \sum_{\omega\in\mathbb{Z}^d}\widehat{\lambda}_\omega e^{2\pi i\omega\cdot(x-x')}$. This allows to quantify the deviation of $\widehat{S}$ from $\mathbb{E}[\widehat{S}] = \|u\|_F$, with *e.g.* Chebychev's inequality, as shown in next theorem.

Theorem 3 (Certificate with Chebychev Inequality)**.** *Let $(\widehat{\lambda}_\omega)_\omega$ be a probability distribution on $\mathbb{Z}^d$, $\delta \in (0,1)$ and g a positive function. Let $N > 0$ and $\widehat{S}$ be the empirical mean of $|\widehat{f}_\omega - c - \widehat{g}_\omega|/\widehat{\lambda}_\omega$ obtained with N samples $\omega_i \sim \widehat{\lambda}_\omega$. Then, a certificate with probability $1-\delta$ is given with*

$$f_\star \geqslant c - \widehat{S} - \frac{\|f - c - g\|_{\mathcal{H}_\lambda}}{\sqrt{N\delta}}, \quad \widehat{S} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\widehat{f}_{\omega_i} - c\mathbf{1}_{\omega_i=0} - \widehat{g}_{\omega_i}\right|}{\widehat{\lambda}_{\omega_i}}. \tag{52}$$

Proof. From its definition in eq. (50), we see that an unbiased estimator of the F-norm is given by $\widehat{S}$. Then, Chebychev's inequality states that $|\widehat{S} - \|u\|_F|^2 \leqslant \operatorname{Var}\widehat{S}/\delta$ with probability at least $1-\delta$. Using the computation of the variance in eq. (51), it follows that $\|u\|_F \leqslant \widehat{S} + \|f - c - g\|_{\mathcal{H}_\lambda}/\sqrt{N\delta}$ with probability at least $1-\delta$. Plugging this expression into eq. (45), we obtain the result. □

Note that the norm in $\mathcal{H}_\lambda$ can be developed with (assuming for conciseness that $(-c)$ is comprised in the 0-frequency of f)

$$\|u\|^2_{\mathcal{H}_\lambda} = \sum_{\omega\in\mathbb{Z}^d} \frac{\widehat{f}^*_\omega(\widehat{f}_\omega - 2\widehat{g}_\omega)}{\widehat{\lambda}_\omega} + \|g\|^2_{\mathcal{H}_\lambda} \leqslant (\|f\|_{\mathcal{H}_\lambda} + \|g\|_{\mathcal{H}_\lambda})^2. \tag{53}$$

Thus, theorem 3 provides a certificate of $f_\star$ as long as we can *(i)* evaluate the Fourier transform $\widehat{g}_\omega$ of g and *(ii)* compute its Hilbert norm in some r.k $\mathcal{H}_\lambda$ induced by $\widehat{\lambda}_\omega$. In next section, we detail the choice we make to achieve this efficiently, with kernels amenable to GPU computation, scaling to thousands of coefficients.

Remark 5 (Using a RKHS norm instead of the F-norm)**.** *Note that since $(\widehat{\lambda}_\omega)_\omega$ sums to 1, the associated kernel is bounded by 1. Hence $\|u\|_{L_\infty} \leqslant \|u\|_{\mathcal{H}_\lambda}$, and the latter could be used instead of the F-norm in eq.* (45)*. There are two reasons for taking our approach instead. Firstly, the F-norm is always tighter that a RKHS norm (see* e.g.*[38, Lem. 4]); secondly, we cannot compute $\|u\|_{\mathcal{H}_\lambda}$ efficiently and have to rely instead on another upper bound. However, taking the number of samples $N = O(\|u\|^2_{\mathcal{H}_\lambda})$ alleviates this issue.*

Exponential concentration bounds with MoM. The scaling in $1/\sqrt{\delta}$ in theorem 3 can be prohibitive if one requires a high probability on the result ($\delta \ll 1$). Hopefully, alternative estimator exist for those cases. The Median-of-Mean estimator is an example, illustrated in theorem 4.

Theorem 4 (Certificate with MoM estimator)**.** *Let $(\widehat{\lambda}_\omega)_\omega$ be a probability distribution on $\mathbb{Z}^d$, and $\delta \in (0,1)$. Draw $N > 0$ frequencies $\omega_i \sim \widehat{\lambda}_\omega$. Define the MoM estimator with the following: for $K \in \mathbb{N}$ s.t. $\delta = e^{-K/8}$ and $N = Kb$, $b \geqslant 1$, write $B_1, \dots, B_K$ a partition of $[N]$; then*

$$\mathsf{MoM}_\delta(|\widehat{u}_{\omega_i}|/\lambda_{\omega_i}) = \text{median}\left\{ \frac{1}{b}\sum_{i\in B_j} \frac{|\widehat{f}_{\omega_i} - c\mathbf{1}_{\omega_i=0} - \widehat{g}_{\omega_i}|}{\lambda_{\omega_i}} \right\}_{1\leqslant j\leqslant K}. \tag{54}$$

A certificate on $f_\star$ with probability $1-\delta$ follows, with

$$f_\star \geqslant c - \mathsf{MoM}_\delta(|\widehat{u}_{\omega_i}|/\lambda_{\omega_i}) - 4\sqrt{2}\,\|f - c - g\|_{\mathcal{H}_\lambda}\sqrt{\frac{\log(1/\delta)}{N}}. \tag{55}$$

Proof. Using *e.g.* Theorem 4.1 from [8] we get that the deviation of the MoM estimator from the expectation is bounded by

$$|\|u\|_F - \mathsf{MoM}_\delta(|\widehat{u}_{\omega_i}|/\lambda_{\omega_i})| \leqslant 4\sqrt{2}\sqrt{\mathsf{Var}(|\widehat{u}_\omega|/\widehat{\lambda}_\omega)\frac{\log(1/\delta)}{N}} \text{ with proba. } 1-\delta. \tag{56}$$

Using the upper bound on the variance with the $\mathcal{H}_\lambda$ norm given in eq. (51) and plugging the resulting expression into eq. (45), we obtain the result. □

To conclude this section, bounding the L_∞ norm from above with the F-norm in eq. (46) enables to obtain a certificate on f, as shown in theorem 2. The F-norm requires an infinite number of computation in the general case, but can be bounded efficiently with a probabilistic estimate, given by theorem 3 or theorem 4. This is summed up in fig. 3. Note that the difference $\|\cdot\|_F - \|\cdot\|_{L_\infty}$ is a source of conservatism in the certificate which we do not quantify – yet, the F-norm is optimal for a class of norms, see [38, Lemma 3].

9.3 Algorithm and implementation

9.3.1 Bessel kernel

We now detail the specific choice of kernel we make in order to compute the certificate of theorem 3 or theorem 4 efficiently. Our first observation is to use a kernel stable by product, so

that we can easily characterize a Hilbert space the model g belongs to. This restricts the choice to the exponential family. That's why we define, for a parameter $s > 0$,

$$\forall x \in \mathbb{T}, \quad q_s(x) = e^{s(\cos(2\pi x)-1)} = \sum_{\omega\in\mathbb{Z}} e^{-s} I_{|\omega|}(s) e^{2\pi i \omega x}, \tag{57}$$

with $I_{|\omega|}(\cdot)$ the modified Bessel function of the first kind [37, p.181]. Then, define $K_s(x,y) = q_s(x-y)$ as in definition 6, and take a tensor product to extend the definition of K to multiple dimension, *i.e.* $K_{\mathbf{s}}(\mathbf{x},\mathbf{y}) = \prod_{\ell=1}^{d} K_{\mathbf{s}_\ell}(\mathbf{x}_\ell, \mathbf{y}_\ell)$ for any $\mathbf{x},\mathbf{y} \in \mathbb{T}^d$. We refer to this kernel as the *Bessel kernel*, and the associated RKHS as $\mathcal{H}_{\mathbf{s}}$. It is stable by product as $K_{\mathbf{s}}(x,y) = K_{\mathbf{s}/2}(x,y)^2$. This is key to compute the Fourier transform of the model g, and in contrast to previous approaches which used the exponential kernel with $\widehat{q}_\omega \propto e^{-s|\omega|}$ [38, 1].

In the following, g is a K-SoS model defined as in definition 5, with the Bessel kernel of parameter $\mathbf{s} \in \mathbb{R}_+^d$ defined in eq. (57).

Lemma 1 (Fourier coefficient of the Bessel kernel)**.** *For $\omega \in \mathbb{Z}^d$, the Fourier coefficient of g in ω can be computed in $O(drm^2)$ time with*

$$\widehat{g}_\omega = \sum_{i,j=1}^{m} R_i^\top R_j \prod_{\ell=1}^{d} e^{-2\mathbf{s}_\ell} I_{|\omega_\ell|}(2s\cos\pi(\mathbf{z}_{i\ell} - \mathbf{z}_{j\ell})) e^{-i\pi\omega_\ell(\mathbf{z}_{i\ell}+\mathbf{z}_{j\ell})}. \tag{58}$$

Proof. From its definition in eq. (48), we rewrite g as

$$g(x) = \sum_{i,j=1}^{m} R_i^\top R_j \prod_{\ell=1}^{d} K_s(x, \mathbf{z}_{i\ell}) K_s(x, \mathbf{z}_{j\ell}). \tag{59}$$

Now, from the definition of the Bessel kernel in eq. (57), we have that for any $(x,y,z) \in \mathbb{T}$, $K(x,y)K(x,z) = e^{-2s} e^{2s\cos(2\pi(y-z)/2)\cos 2\pi(x-(y+z)/2)}$. By definition of the modified Bessel function, the Fourier coefficient of this expression are given by $I_{|\omega|}(2s\cos(2\pi(y-z)/2))$. Using this into eq. (59), we get the result. □

The second necessary ingredient for using the certificate of theorem 3 is computing a RKHS norm for g. It relies on the inclusion of $\mathcal{H}_{2\mathbf{s}}$ into the bigger space of symmetric operator $\mathcal{S}(\mathcal{H}_{\mathbf{s}})$.

Lemma 2 (Bound on the RKHS norm of g)**.** *g belongs to $\mathcal{H}_{2\mathbf{s}}$, and $\|g\|_{\mathcal{H}_{2\mathbf{s}}}$ is bounded by the Hilbert-Schmidt norm of $\mathcal{S}(\mathcal{H}_{\mathbf{s}})$, which can be computed in $O(dm^2 + mr^2)$ time, with*

$$\|g\|^2_{\mathcal{H}_{2\mathbf{s}}} \leqslant \|g\|^2_{\mathcal{S}(\mathcal{H}_s)} = \operatorname{Tr}(RK_{s,\mathbf{z}}R^\top)^2. \tag{60}$$

Proof. Assume that $d = 1$; the reasoning can be extended to multiple dimensions with the tensor product. From the computation of the Fourier coefficient in lemma 1 and the fact that $I_{|\omega|}(2s\cos(2\pi\cdot)) \leqslant I_{|\omega|}(2s)$, we have that $\widehat{g}_\omega = O(I_{|\omega|}(2s))$ hence $g \in \mathcal{H}_{2s}$. Finally, since the kernel is stable by product, $\mathcal{H}_{2s} = \mathcal{H}_s \odot \mathcal{H}_s$, so we can use *e.g.* [25, Thm. 5.16], with $\mathcal{H}_1 = \mathcal{H}_2 = \mathcal{H}_s$ and $\mathcal{S}(\mathcal{H}_s) = \mathcal{H}_s \otimes \mathcal{H}_s$, with the operator $A = (\varphi(\mathbf{z}_1), \dots, \varphi(\mathbf{z}_m))R^\top R(\varphi(\mathbf{z}_1), \dots, \varphi(\mathbf{z}_m))^* \in \mathcal{S}(\mathcal{H}_s)$. □

With lemma 2, we have that the model g belongs to $\mathcal{H}_{2\mathbf{s}}$, so we will naturally use $\widehat{\lambda}_\omega = \prod_{\ell=1}^{d} e^{-2\mathbf{s}_\ell} I_\omega(2\mathbf{s}_\ell)$ in theorem 3; said differently, the space $\mathcal{H}_\lambda$ introduced in eq. (51) is simply $\mathcal{H}_{2\mathbf{s}}$ defined in eq. (57).

9.3.2 The algorithm: GloptiNets

We can now describe how GloptiNets yields a certificate on f. The key observation is that no matter how is obtained our model $g(R,\mathbf{z})$ from definition 5, we will always be able to compute a certificate with theorems 3 and 4. Thus, even though optimizing eq. (52) w.r.t $(c,R,\mathbf{z})$ is highly non-convex, we can use any optimization routine and check empirically its efficiency by looking at the certificate. Finally, thanks to its low-rank structure it is cheaper to evaluate g than evaluating its Fourier coefficient. This is formally shown in proposition 5 in section C.1, where a block-diagonal structure for the model is also introduced. That's why we first optimize

$\sup_{c,g} c - \|f - c - g\|_\star$, where $\|\cdot\|_*$ is a proxy for the L_∞ norm, *e.g.* the log-sum-exp on a random batch of N points[4]:

$$\|f - c - g\|_{L_\infty} \approx \max_{i\in[N]} |f(x_i) - c - g(x_i)| \approx \mathrm{LSE}(f(x_i) - c - g(x_i))_{i\in[N]}. \tag{61}$$

This optimization can be carried out by any deep learning libraries with automatic differentiation and any flavour of gradient ascent. Only afterwards do we compute the certificate with theorems 3 and 4. This procedure is summed up in alg. 2.

Algorithm 2: GloptiNets

Data: A trigonometric polynomial f, a candidate z s.t. $c = f(z)$, a model g, and a probability δ.
Result: A certificate $|f_\star - f(z)| \leqslant \epsilon_\delta$ with proba. $1-\delta$.
`/* Optimize g with function values */`
for *epoch = 1:nepochs* **do**
  Sample $x_1, \dots, x_N \in \mathbb{T}^d$;
  $L, \nabla L = \mathrm{autodiff}(\mathrm{LSE}(f(x_i) - c - g(x_i))_{i\in[N]})$;
  $\mathbf{z}, R \leftarrow \mathrm{optimizer}(\nabla L)$;
`/* Compute a certificate */`
$\widehat{\lambda}_\omega$: probability distribution on $\mathbb{Z}^d$ with $\widehat{\lambda}_\omega = \prod_{\ell=1}^d e^{-2s_\ell} I_\omega(2\mathbf{s}_\ell)$;
Sample $\Omega = (\omega_1, \dots, \omega_N) \sim \widehat{\lambda}_\omega$;
Compute $M = \mathsf{MoM}_\delta(|\widehat{f}_{\omega_i} - c\mathbf{1}_{\omega_i=0} - \widehat{g}_{\omega_i}|/\lambda_{\omega_i})_{i\in[n]}$ and $\overline{\sigma} = \|g\|_{S(\mathcal{H}_s)}$;
Returns $\epsilon_\delta = c - M - 4\sqrt{2}\overline{\sigma}\sqrt{\log(1/\delta)/N}$;

Remark 6 (Providing a candidate)**.** *In alg. 2, a candidate estimate* c *for the minimum value* $f(x_\star)$ *is necessary. However, it is possible to overcome this requirement by incorporating* c *as a learnable parameter within the training loop. Moreover,* $x_\star$ *can be learned using techniques similar to those in [27]: by replacing the lower bound* c *with a parabola centered at* z*,* z *becomes a candidate for* $x_\star$ *with precision corresponding to the tightness of the certificate. Note however that this method introduces additional hyperparameters.*

9.3.3 Specific implementation for the Chebychev basis

As already observed in [1], a result on trigonometric polynomial on $\mathbb{T}^d$ directly extends to a real polynomials on $[-1,1]^d$. The reason for that is that minimizing h on $[-1,1]^d$ amounts to minimizing the trigonometric polynomial $f = h((\cos 2\pi x_1, \dots, \cos 2\pi x_d))$ on $\mathbb{T}^d$. Note however that f is an even function in all dimension, as for any $x \in \mathbb{T}^d$, $f(x) = f(x_1, \dots, -x_i, \dots, x_d)$. Thus, approximating $f - f_\star$ with a K-SoS model of definition 5 is suboptimal, in the sense that we could approximate f only on $[0, 1/2]^d$, which is 2^{-d} smaller. Put differently, the Fourier coefficient of f are real by design: it would be convenient to enforce this structure in the model g. This is achieved with proposition 4.

Proposition 4 (Kernel defined on the Chebychev basis)**.** *Let* q *be a real, even function on the torus, bounded by* 1*, as in eq.* (49)*. Let* K *be the kernel defined on* $[-1,1]$ *by*

$$\forall (u,v) \in (0, 1/2), K(\cos 2\pi u, \cos 2\pi v) = \frac{1}{2}(q(u+v) + q(u-v)). \tag{62}$$

Then K *is a symmetric, p.d., hence reproducing kernel, bounded by* 1*, with explicit feature map given by*

$$\forall (x,y) \in [-1,1], K(x,y) = \widehat{q}_0 + \sum_{\omega\in\mathbb{N}} 2\widehat{q}_\omega H_\omega(x) H_\omega(y). \tag{63}$$

[4]Another detail of practical importance is that this loss can be efficiently backpropagated through; on the other hand, the certificate is not easily vectorized, and the Bessel function involved would require specific approximation to be efficiently backpropagated through.

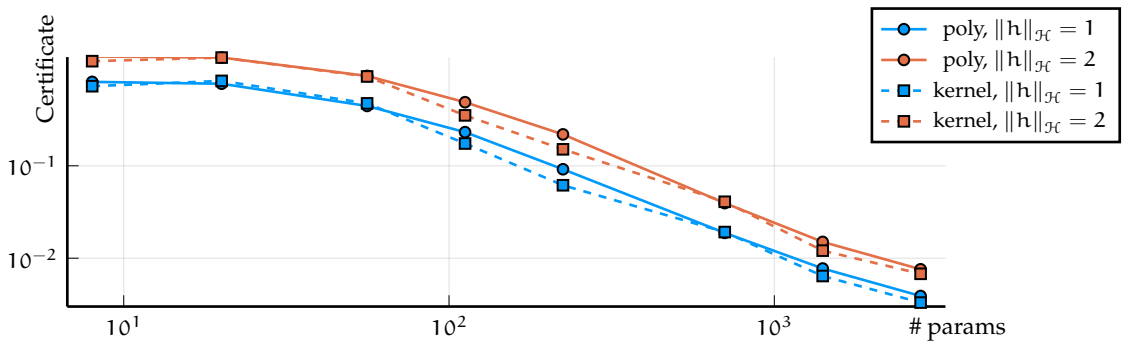


Figure 4: Certificate *vs.* number of parameters in g, for a given function h. The higher the RKHS norm of h, the more difficult it is to approximate uniformly and the looser the certificate, independently of the function type. The more parameters in the k-SoS model, the tighter the certificates obtained with theorem 4.

The proof is available in section C.2. It simply relies on expanding the definition of K in eq. (62). The resulting expression in eq. (63) exhibits only cosine terms (in the decomposition of $x \mapsto K(\cos 2\pi x, y)$). This enables to directly extend the PSD models from definition 5 with such kernels. Finally, when used with the Bessel kernel of eq. (57), we recover an easy computation of the Chebychev coefficient, as shown in lemma 3, in $O(drm^2)$ time. This enables to approximate any function expressed on the Chebychev basis. Note that polynomials expressed in other basis can be certified too, by first operating a change of basis.

9.4 Experiments

The code to reproduce these experiments is available at
`github.com/gaspardbb/GloptiNets.jl`

Settings. Given a function h, we compute a candidate $\widehat{x}$ with gradient descent and multiple initializations. The goal is then to certify that $\widehat{x}$ is indeed a global minimizer of h. This is a common setup in the Polynomial-SoS literature [34]. To illustrate the influence of the number of parameters, the positive model g defined in definition 5 for GloptiNets designates either a small model GN-small with 1792 parameters, or a bigger model GN-big with 22528 parameters. The latter should have higher expressivity and better interpolate positive functions, leading to tighter certificates. All results for GloptiNets are obtained with confidence $1-\delta = 1-e^{-4} \geqslant 98\%$. All other details regarding the experiments are reported in section C.3.

Polynomials. We first consider the case where h is a random trigonometric polynomial. Note that this is a restrictive analysis, as GloptiNets can handle any smooth functions (*i.e.* with infinite non-zero Fourier coefficients). Polynomials have various dimension d, degree p, number of coefficients n, but a constant RKHS norm $\mathcal{H}_{\mathbf{21}_d}$. We compare the performances of GloptiNets to TSSOS, in its complex polynomial variant [34]. The latter is used with parameters such that it executes the fastest, but without guarantees of convergence to the global minimum $f_\star$. Table 1 shows the certificates $h(x_\star) - h(\widehat{x})$ and the execution times (lower is better, t in seconds) for TSSOS, GN-small and GN-big. Figure 4 provides certificate on a random polynomial, function of the number of parameters in g.

Kernel mixtures. While polynomials provide ground for comparison with existing work, GloptiNets is not confined to this function class. This is evidenced by experiments on kernel mixtures, where our approach stands as the only viable alternative we are aware of. The function we certify are of the form $h(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$, where K is the Bessel kernel of eq. (57). Kernel mixtures are ubiquitous in machine learning and arise *e.g.* when performing

Table 1: GloptiNets and TSSOS on random trigonometric polynomials. While TSSOS provides machine-precision certificates, its running time grows exponentially with the problem size, and eventually fails on problems 3 and 6. On the other hand, GloptiNets has constant running time no matter the problem size, and its certificates can be tightened by increasing the model size.

d	p	n	TSSOS		GN-small		GN-big	
			Certif.	t	Certif.	t	Certif.	t
3	5	85	$5.3 \cdot 10^{-11}$	3	$8.35 \cdot 10^{-4}$	$6 \cdot 10^3$	$2.64 \cdot 10^{-4}$	$9 \cdot 10^3$
	7	231	$4.7 \cdot 10^{-13}$	120	$9.51 \cdot 10^{-4}$	$6 \cdot 10^3$	$2.90 \cdot 10^{-4}$	$9 \cdot 10^3$
	9	489	out of memory!	-	$1.18 \cdot 10^{-3}$	$6 \cdot 10^3$	$3.34 \cdot 10^{-4}$	$9 \cdot 10^3$
4	3	33	$3.1 \cdot 10^{-10}$	0.1	$2.46 \cdot 10^{-2}$	$1 \cdot 10^4$	$3.45 \cdot 10^{-3}$	$2 \cdot 10^4$
	5	225	$4.8 \cdot 10^{-12}$	53	$3.71 \cdot 10^{-2}$	$1 \cdot 10^4$	$3.59 \cdot 10^{-3}$	$2 \cdot 10^4$
	7	833	out of memory!	-	$4.76 \cdot 10^{-2}$	$1 \cdot 10^4$	$4.85 \cdot 10^{-3}$	$2 \cdot 10^4$

kernel ridge regression. Certificates obtained on mixtures are compared with those obtained on polynomials in fig. 4, function of the model size g.

Results. There are two key hindsight about the performances of GloptiNets. Firstly, its certificate *does not depend on the structure* of the function to optimize. Thus, although GloptiNets does not match the performances of TSSOS on small polynomials, it can tackle polynomials which cannot be handled by competitors, with arbitrarily as many coefficients ($n = \infty$). For instance, TSSOS cannot handle problems with $n \in \{489, 833\}$ in table 1. More importantly, GloptiNets can certify a richer class of functions than polynomials, among which kernel mixtures. The performances of GloptiNets mostly depends on the complexity of the function to certify, as measured with its RKHS norm.

Secondly, note that *a bigger model yields tighter certificate*. This is detailed in fig. 4, where the same function f is optimized with various models. The dependency of the certificate on the norm of f is shown in fig. 5 in section C.3, along with experiments with Chebychev polynomials.

9.5 Limitations

One limitation of GloptiNets is the trade-off resulting from its high flexibility for obtaining a certificate as in alg. 2. While this flexibility offers numerous advantages, it also introduces the need for an extensive hyperparameter search. Although we have identified a set of hyperparameters that align with deep learning practices – utilizing a Momentum optimizer with cosine decay and a large initial learning rate – the optimal settings may vary depending on the specific problem at hand.

In the same vein, the certificates given by GloptiNets are of moderate accuracy. While adding more parameters into the k-SoS model certainly helps (as shown in fig. 4), alternative optimization scheme to interpolate $h - h(\widehat{x})$ with g might provide easier improvement. For instance, we found that using approximate second-order scheme in alg. 2 is key to obtaining good certificates.

In the specific settings of polynomial optimization, we highlight that our model is not competitive on problems which exhibits some algebraic structure, as for instance term sparsity or the constant trace property. Typically, problems with coefficients of low degrees (less or equal than 2), which encompass notably the OPF problem, are really well handled by the family of solvers TSSOS belongs to. Finally, GloptiNets does not handle constraints yet.

9.6 Conclusion

The GloptiNets algorithm presented in this work lays the foundation for a new family of solvers which provide certificates to non-convex problems. While our approach does not aim to replace the well-established Lasserre's hierarchy for sparse polynomials, it offers a fresh perspective on tackling a new set of problems at scale. Through demonstrations on synthetic examples, we have showcased the potential of our approach. Further research directions include extensive

parameter tuning to obtain tighter certificates, with the possibility of leveraging second-order optimization schemes, along with warm-restart schemes for application which requires solving multiple similar problems sequentially.

C Appendix of GloptiNets: Scalable Non-Convex Optimization with Certificates

C.1 Extensions

We explore additional extensions of GloptiNets that further enhance its appeal. We first describe a block diagonal structure for the model for faster evaluation, a theoretical splitting scheme for optimization, and finally a warm-start scheme.

C.1.1 Block diagonal structure for efficient computation

Without any further assumption, we see that a model from definition 5 can be evaluated in $O(drm)$ time; its Fourier coefficient given by lemma 1 in $O(dm^2r)$; the bound on the RKHS norm is computed in $O(dm^2 + mr^2)$ time thanks to lemma 2; all that enables to compute a certificate, as stated in theorem 3, in $O(Ndm^2r + mr^2)$ time, where N is the number of frequencies sampled. If the function f to be minimized has big $\mathcal{H}_s$ norm, we might need a large model size m to have $f - f_\star \approx g$. Hence, we introduce specific structure on G which makes it *block-diagonal* and *better conditioned*.

Proposition 5 (Block-diagonal PSD model)**.** *Let g be a PSD model as in definition 5, with $m = bs$ anchors. Split them into b groups, denoting them $\mathbf{z}_{ij}$, $i \in [b]$ and $j \in [s]$. Compute the Cholesky factorization of each kernel matrix $T_i^\top T_i = K_{\mathbf{z}_i} \in \mathbb{R}^{s\times s}$. Then, define G as a block-diagonal matrix, with b blocks defined as $G_i = \tilde{R}_i\tilde{R}_i^\top$, $\tilde{R}_i = T^{-1}R_i$, and $R_i \in \mathbb{R}^{r\times s}$. Equivalently,*

$$G = \begin{pmatrix} \tilde{R}_1\tilde{R}_1^\top & & \\ & \ddots & \\ & & \tilde{R}_b\tilde{R}_b^\top \end{pmatrix}, \quad s.t. \quad g(x) = \sum_{i=1}^{b} \left\|\tilde{R}_i^\top K_{\mathbf{z}_i}(x)\right\|^2, \quad K_{\mathbf{z}_i}(x) = K(\mathbf{z}_{ij}, x)_{1\leqslant j\leqslant s}. \tag{64}$$

Then g can be evaluated in $O(rbs^3d)$ time, $\widehat{g}_\omega$ in $O(bs^2(dr+s))$ time, and $\|g\|^2_{\mathcal{S}(\mathcal{H}_s)}$ in $O(b^2(rs^2 + r^2s) + bs^3)$ time. The model has $(r+d)bs$ real parameters.

Proof. Having G defined as such, it is psd, of rank at most $rb \leqslant sb = m$. Written $g(x) = \sum_{i=1}^{b}\|\tilde{R}_i^\top K_{\mathbf{z}_i}(x)\|^2$, we can compute the Fourier coefficient by applying lemma 1 to each of the b component. Adding the cost of computing $G_i = \tilde{R}_i\tilde{R}_i^\top$ results in complexity of $O(bs^2(dr+s))$. Finally, note that $\|g\|^2_{\mathcal{S}(\mathcal{H}_s)} = \|A\|^2_{\mathcal{S}(\mathcal{H}_s)}$ where

$$A = ((\varphi(\mathbf{z}_{1j}))_{j\in[s]}, \dots, (\varphi(\mathbf{z}_{bj}))_{j\in[s]})(\operatorname{Diag} G_i)_{i\in[b]}((\varphi(\mathbf{z}_{1j}))_{j\in[s]}, \dots, (\varphi(\mathbf{z}_{bj}))_{j\in[s]})^*.$$

Then, defining Q the matrix of $b\times b$ blocks of size $s\times s$ s.t. for $j,k\in[b]$, $Q_{jk} = K(\mathbf{z}_j, \mathbf{z}_k) \in \mathbb{R}^{s\times s}$, we have

$$\|A\|^2_{\mathcal{S}(\mathcal{H}_s)} = \operatorname{Tr}\, Q(\operatorname{Diag} G_i)_{i\in[b]}Q(\operatorname{Diag} G_i)_{i\in[b]} = \sum_{j,k=1}^{b} \operatorname{Tr}\, G_jQ_{jk}G_kQ_{kj}, \tag{65}$$

and each term in the sum can be written $\operatorname{Tr}(\tilde{R}_j^\top Q_{jk}\tilde{R}_k)(\tilde{R}_k^\top Q_{kj}\tilde{R}_j^\top) = \|\tilde{R}_j^\top Q_{jk}\tilde{R}_k\|^2_{HS}$, which is computed in $O(rs^2 + r^2s)$ time, plus $O(bs^3)$ to compute the Cholesky factor. □

Denoting $\varphi_{\mathbf{z}_i} = (\varphi(\mathbf{z}_{ij}))_{1\leqslant j\leqslant s}$, note that

$$\varphi_{\mathbf{z}_i}G_i\varphi^*_{\mathbf{z}_i} = \varphi_{\mathbf{z}_i}T_i^{-1}R_iR_i^\top(\varphi_{\mathbf{z}_i}T_i^{-1})^* = E_iR_iR_i^\top E_i^*, \tag{66}$$

with $E_i = \varphi_{\mathbf{z}_i}T_i^{-1}$ an orthonormal basis of $\operatorname{Span}(\varphi_{\mathbf{z}_{ij}})_{1\leqslant j\leqslant s}$ as $E_i^*E_i = \mathbf{I}_s$. Thus, each model's coefficient is defined on an orthonormal basis, which makes the optimization easier. Of course, this comes at an added s^3 complexity, which could be alleviated by using *e.g.* an incomplete Cholesky factorization instead.

Remark 7 (Relation to Term Sparsity in POP)**.** *The successful application of polynomial hierarchies to problems with thousands of variables rely on making the moment matrix M having a block structure [36, 35]. If the monomial basis has size m, the constraint $M \succeq 0$ is replaced with $M = (\operatorname{Diag} M_i)_{i\in[b]}$ and $M_i \succeq 0$. This enables to solve b SDP of size at most s instead of one of size m. Our model in proposition 5 follows a similar route for having a lower computational budget.*

C.1.2 Global optimization with splitting scheme

While GloptiNets can provide certificates for functions, it falls behind local solvers in terms of competitiveness. The challenge lies in the fact that finding a certificate is considerably more difficult than finding a local minimum, as it necessitates the uniform approximation of the entire function. However, we present a novel algorithmic framework that has the potential to enhance the competitiveness of GloptiNets with local solvers while simultaneously delivering certificates. Our approach involves partitioning the search domain into multiple regions and computing lower bounds for each partition. By discarding portions of the domain where we can certify that the function exceeds a certain threshold, the algorithm progressively simplifies the optimization problem and removes areas from consideration. Moreover, such an approach is naturally well suited to parallel computation.

The algorithm relies on a divide-and-conquer mechanism. First, we split the hypercube $(-1,1)^d$ in N regions, where N is the number of core available. We compute an upper bound with a local solver. For each region, we run GloptiNets *in parallel*, computing a certificate at regular interval. As soon as the certificate is bigger than the upper bound, we stop the process: we know that the global minimum is not in the associated region. We can then reallocate the freed computing power by splitting the biggest current region, which yields an easier problem. We stop as soon as the region considered are small enough. This is summarized in alg. 3, where Ⓟ indicates the loop run in parallel.

Note that minimizing f on a hypercube of center μ and size σ amounts to minimizing $x \mapsto f((x-\mu)/\sigma)$ on $[-1,1]^d$, which is another Chebychev polynomial whose coefficients can be evaluated efficiently thanks to the order-2 relation every orthonormal polynomial satisfy. For Chebychev polynomials, that is $H_{\omega+1}(x) = 2xH_\omega(x) - H_{\omega-1}(x)$.

Algorithm 3: Splitting scheme with GloptiNets

Data: A Chebychev polynomial f with a unique global optimum, a probability δ, a number of cores N and a volume $\rho < 1/N$.
Result: A certificate on f: $f_\star \geqslant C_\delta(f)$ with proba. $1-\delta_\star$.
`/* Initialization: upper bound and partition */`
$\Pi = \mathsf{partition}([-1,1]^d, N)$, $\delta_\star = 0$;
Ⓟ $\mathsf{ub} = \min_{\pi\in\Pi}\{\mathsf{localsolver}_{x\in\pi} f(x)\}$;
`/* Iterate over the partition */`
Ⓟ **for** $\pi \in \Pi$, **While** $\mathsf{length}(\Pi) > 1$ **do**
  while $C_\delta(f_\pi) < \mathsf{ub}$ **do**
    Continue optimization;
  Split biggest part: $\pi_0 = \arg\max_{\pi\in\Pi} \mathrm{Vol}(\pi)$; $(\pi_1,\pi_2) = \mathsf{partition}(\pi_0, 2)$;
  If $\mathrm{Vol}(\pi_{1,2}) < \rho$: end this process ;
  Update upper bound: $\mathsf{ub} = \min\{\mathsf{ub}, \mathsf{localsolver}_{x\in\pi_{1,2}} f(x)\}$;
  Update search space and $\delta_\star$: $\Pi = \Pi \setminus \{\pi,\pi_0\} \cup \{\pi_1,\pi_2\}$, $\delta_\star = 1-(1-\delta_\star)(1-\delta)$;
`/* A single region in Π remains */`
Returns $\Pi = \{\pi\}$, $C_\delta(f_\pi)$, $\delta_\star$;

C.1.3 Warm restarts

Our model distinguishes itself by leveraging the analytical properties of the objective function, rather than relying solely on algebraic characteristics. This approach offers a notable advantage, as closely related functions can naturally benefit from a warm restart. For example, if we already have a certificate for a function f using a PSD model g, and we seek to compute a certificate for a similar function $\tilde{f} \approx f$, we can readily employ GloptiNets by initializing the PSD model with g. Indeed, if $f - f_\star \approx g$, we can expect $\tilde{f} - \tilde{f}_\star \approx g$, so we can expect the optimization to be faster.

In contrast, P-SoS methods, which rely on SDP programs, cannot directly adapt to new problems without significant effort. For instance, if a new component is introduced, an entirely new SDP must be solved. Our model's ability to accommodate related yet distinct problems could prove highly valuable in domains with a frequent need to certify different but closely related problems. In the industry, the Optimal Power Flow (OPF) problem requires periodic

solves every 5 minutes [30]. With GloptiNets, once the initial challenging solve is performed, subsequent solves become easier assuming minimal changes in supply and demand conditions.

C.1.4 Optimizing the certificate directly

As explained in section 9.3.2 where GloptiNets is introduced, we optimize a proxy of the L_∞ norm rather than the certificate of theorems 3 and 4. This proxy is the log-sum-exp on a random batch of N points. The reason for this is that evaluating an extended k-SoS model $g(x)$ on $x \in \mathbb{T}^d$ requires $O(drs)$ time, while evaluating $\widehat{g}_\omega$ on $\omega \in \mathbb{Z}^d$ requires $O(drs^2)$ time. Yet, optimizing the certificate directly could probably help obtaining higher-precision certificate. Lemma 4 in section C.4 sketches a method to reduce the computational cost of the Fourier computation from $O(s^2)$ to $O(s)$.

C.2 Kernel defined on the Chebychev basis

In this section we describe the approach we take to model functions written in the Chebychev basis. For h such a polynomial, a naive approach would simply model $f = h \circ \cos(2\pi\cdot)$ as a trigonometric polynomial. However, note that the decomposition of f only has cosine terms. Thus, approximating $f - f_\star$ efficiently requires a PSD model which has only cosine terms in its Fourier decomposition. This is achieved by using a kernel written in the Chebychev basis, as introduce in proposition 4, for which we now provide a proof.

Proof of proposition 4. Let $x, y \in [-1, 1]$ and $u, v \in [0, 1/2]$ s.t. $x, y = \cos(2\pi u), \cos(2\pi v)$, by bijectivity of the cosine function on $[0, \pi]$. From the definition of K in eq. (62) and the definition of q in eq. (49), we have that

$$\begin{aligned} K(x, y) &= \frac{1}{2} \sum_{\omega \in \mathbb{Z}} \widehat{q}_\omega \left(e^{2\pi i \omega (u+v)} + e^{2\pi i \omega (u-v)} \right) \\ &= \sum_{\omega \in \mathbb{Z}} \widehat{q}_\omega e^{2\pi i \omega u} \cos(2\pi\omega v) \\ &= \widehat{q}_0 + 2 \sum_{\omega \in \mathbb{N}} \widehat{q}_\omega \cos(2\pi\omega u) \cos(2\pi\omega v) \\ &= \widehat{q}_0 + 2 \sum_{\omega \in \mathbb{N}} \widehat{q}_\omega H_\omega(u) H_\omega(v). \end{aligned}$$

Since q has positive Fourier transform, this makes the feature map of K explicit with $K(x, y) = \varphi(u) \cdot \varphi(v)$, $\varphi(u)_\omega = \sqrt{(1 + \mathbf{1}_{\omega \neq 0})\widehat{q}_\omega} H_\omega(u)$, for $\omega \in \mathbb{N}$. Hence the kernel is a reproducing kernel. □

We now use this kernel with the Bessel function $x \mapsto e^{s(\cos(2\pi x) - 1)}$, *i.e.* we define the kernel K on $[-1, 1]$ to satisfy

$$\forall u, v \in (0, 1/2), \quad K(\cos(2\pi u), \cos(2\pi v)) = \frac{1}{2} \left(e^{s(\cos(2\pi(u+v))} + e^{s(\cos(2\pi(u-v))} \right). \tag{67}$$

As it was the case for the torus, this kernel enables an easy characterization of a RKHS in which an associated PSD model g lives.

Lemma 3 (Chebychev coefficient of the Bessel kernel)**.** *Let g be a PSD model as in definition 5, with the kernel K of eq.* (67)*. Then, the Chebychev coefficient $\omega \in \mathbb{N}^d$ of g can be computed in $O(rdm^2)$ time with*

$$g_\omega = \sum_{i,j=1}^{m} R_i^\top R_j \prod_{\ell=1}^{d} (1 + \mathbf{1}_{\omega \neq 0}) \frac{e^{-2\mathbf{s}_\ell}}{2} \Big[\begin{aligned} &I_{\omega_\ell}(2\mathbf{s}_\ell \sigma_{-\ell ij}) H_{\omega_\ell}(\sigma_{+\ell ij}) \\ &+ I_{\omega_\ell}(2\mathbf{s}_\ell \sigma_{+\ell ij}) H_{\omega_\ell}(\sigma_{-\ell ij}) \Big] \end{aligned} \tag{68}$$

where

$$\sigma_{\pm \ell ij} = \cos(2\pi m_{\pm \ell ij}), \quad m_{\pm \ell ij} = (\mathbf{u}_{\ell ij} \pm \mathbf{u}_{\ell ij})/2, \quad \textit{and} \quad \cos 2\pi \mathbf{u}_{\ell ij} = \mathbf{z}_{\ell ij}.$$

Proof.

Expanding g and definition of Chebychev coefficient. From the definition of g in eq. (48), we have

$$g(\mathbf{x}) = \sum_{i,j=1}^{m} R_i^\top R_j \prod_{\ell=1}^{d} K_{\mathbf{s}_\ell}(\mathbf{x}_\ell, \mathbf{z}_{\ell i}) K_{\mathbf{s}_\ell}(\mathbf{x}_\ell, \mathbf{z}_{\ell j}). \tag{69}$$

We consider $x, y, z \in (-1, 1)$ and $s > 0$. We denote $u, v, w \in (0, 1/2)$ s.t.

$$x, y, z = \cos 2\pi u, \cos 2\pi v, \cos 2\pi w$$

with the bijectivity of $x \mapsto \cos(2\pi x)$ on $(0, 1/2)$. We now compute the Chebychev coefficient of $x \mapsto K_s(x, y)K_s(x, z)$. Denoted p_ω, this is

$$\forall \omega \in \mathbb{N}, \;\; p_\omega = \frac{1 + \mathbf{1}_{\omega \neq 0}}{\pi} \int_{-1}^{1} K_s(x, y) K_s(x, z) T_\omega(x) \frac{dx}{\sqrt{1 - x^2}},$$

or equivalently

$$\forall \omega \in \mathbb{N}, \;\; p_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \int_0^1 K_s(\cos 2\pi u, \cos 2\pi v) K_s(\cos 2\pi u, \cos 2\pi w) \cos(2\pi \omega u) du. \tag{70}$$

Chebychev coefficient of kernel product. With the definition of the kernel in proposition 4, eq. (62), we have

$$\begin{aligned} K_s(x, y)K_s(x, z) &= \frac{1}{4}\left(p(u + v) + p(u - v)\right) \times \left(p(u + w) + p(u - w)\right) \\ &= \frac{e^{-2s}}{4}\left(e^{s \cos 2\pi(u+v)} + e^{s \cos 2\pi(u-v)}\right) \times \left(e^{s \cos 2\pi(u+w)} + e^{s \cos 2\pi(u-w)}\right) \end{aligned}$$

Now use the sum-to-product formula with the cosines to obtain

$$\begin{aligned} K_s(x, y)K_s(x, z) = \frac{e^{-2s}}{4}\Bigg(& e^{2s \cos 2\pi(\frac{v-w}{2}) \cos 2\pi(u + \frac{v+w}{2})} + e^{2s \cos 2\pi(\frac{v-w}{2}) \cos 2\pi(u - \frac{v+w}{2})} \\ & + e^{2s \cos 2\pi(\frac{v+w}{2}) \cos 2\pi(u + \frac{v-w}{2})} + e^{2s \cos 2\pi(\frac{v+w}{2}) \cos 2\pi(u - \frac{v-w}{2})} \Bigg), \end{aligned} \tag{71}$$

We simplify this expression by introducing

$$m_\pm = \frac{1}{2}(v \pm w) \;\text{ and }\; \sigma_\pm = \cos 2\pi m_\pm. \tag{72}$$

Then, eq. (71) becomes

$$\begin{aligned} K_s(x, y)K_s(x, z) = \frac{e^{-2s}}{4}\Bigg(& e^{2s\sigma_- \cos 2\pi(u + m_+)} + e^{2s\sigma_- \cos 2\pi(u - m_+)} \\ & + e^{2s\sigma_+ \cos 2\pi(u + m_-)} + e^{2s\sigma_+ \cos 2\pi(u - m_-)} \Bigg). \end{aligned} \tag{73}$$

We recognize the definition of the kernel (which is not a surprise as we chose the kernel to be stable by product). However, we need variables in $(0, 1/2)$ to retrieve the proper definition of the kernel. Instead, we use lemma 5 on eq. (73) combined with eq. (70), to obtain

$$\begin{aligned} p_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \frac{e^{-2s}}{4}\Bigg(& \cos(2\pi\omega m_+) I_\omega(2s\sigma_-) + \cos(2\pi\omega m_+) I_\omega(2s\sigma_-) \\ & + \cos(2\pi\omega m_-) I_\omega(2s\sigma_+) + \cos(2\pi\omega m_-) I_\omega(2s\sigma_+) \Bigg), \end{aligned}$$

which gives

$$p_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \frac{e^{-2s}}{2} (\cos(2\pi\omega m_+) I_\omega(2s\sigma_-) + \cos(2\pi\omega m_-) I_\omega(2s\sigma_+)). \tag{74}$$

Equation (74) contains the Chebychev coefficient of the product of two kernel function as defined in eq. (70). Plugging this result into the definition of g in eq. (69), and noting that $\cos(2\pi\omega m_\pm) = H_\omega(\cos 2\pi m_\pm) = H_\omega(\sigma_\pm)$, we obtain the result. □

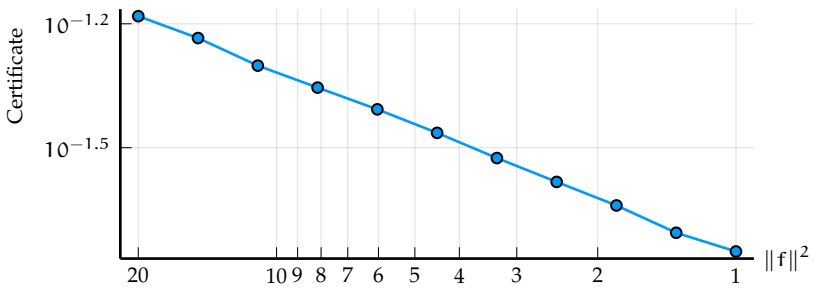


Figure 5: Certificate *vs.* RKHS norm of f, for a given model g with a fixed number of parameters. f has 1146 coefficients and g has 22528 parameters. Best certificate is kept among a set of optimization hyperparameters. As the norm of f decreases, fitting $f - f_\star$ with g is easier and the certificate becomes tighter.

Thanks to lemma 3, we see that a model g defined as in definition 5 with the Bessel kernel K_s of eq. (67) as its Chebychev coefficients decaying in $O(I_\omega(2s))$. Hence, it belongs to $\mathcal{H}_{2s}$, the RKHS associated to K_{2s}.

C.3 Additional details on the experiments

Tuning the hyperparameters. The time reported in section 9.4 does *not* take into account the experiments needed to find a good set of hyperparameters. The parameters tuned were the type of optimizer, the decay of learning rate, and the regularization on the Frobenius norm of G.

Regularization. Regularization is performed by approximating the HS norm with a proxy which is faster to compute. We use $\|R_j^\top R_k\|_{HS}^2$ instead of $\|\tilde{R}_j^\top Q_{jk} \tilde{R}_k\|_{HS}^2$ in eq. (65).

Hardware. GloptiNets was used with NVIDIA V100 GPUs for the interpolation part, and Intel Xeon CPU E5-2698 v4 @ 2.20GHz for computing the certificate. TSSOS was run on a Apple M1 chip with Mosek solver.

Configuration of TSSOS. We use the lowest possible relaxation order d (*i.e.* $\lceil \deg f/2 \rceil$), along with Chordal sparsity. We use the first relaxation step of the hierarchy. In these settings, TSSOS is not guaranteed to converge to $f_\star$ but will executes the fastest.

Certificate vs. number of parameter for a given function. In fig. 4, the target function is a random polynomial of norm 1 or 2, or a kernel mixture with 10 coefficients of norm 1 or 2. The models forming the blue line are defined as in proposition 5, with rank, block size and number of blocks equal to $(1, bs, 1)$ respectively, with bs the block size we vary. The number of frequencies sampled to compute the certificate is $1.6 \cdot 10^7$, and accounts for the fact that the bound on the variance becomes larger than the MOM estimator for large models.

Certificate vs. problem difficulty for a given model. We have 3 related parameters: the quality of the optimization (given by the certificate), the expressivity of the model (given by its number of parameters), and the difficulty of the optimization (given by the norm of the function). In fig. 5, we fix the latter and plot the relation between the first two. Here, we fix the model with parameters $(8, 16, 128)$, and we optimize a polynomial in 3d of degree 12, with RKHS norm ranging from 1 to 20. The certificates obtained are given in fig. 5. The resulting plot exhibits a clear polynomial relation between the certificate and the norm of the function, with a slope of -0.88. This suggest that the certificate behaves as $O(\|f\|_{\mathcal{H}_{2s}}^{1/2})$.

Comparison with TSSOS on the Fourier basis. In table 1, the polynomials f all have a RKHS norm of 1. The small model is defined as in proposition 5, with rank, block size and number of blocks equal to $4, 32, 8$ respectively. For the big models, those values are $8, 128, 16$. The certificate

Table 2: GloptiNets and TSSOS on random Chebychev polynomials. The same conclusion as in table 1 applies. While TSSOS is very efficient on small problems, its memory requirements grow exponentially with the problem size. GloptiNets has less accuracy, but a computational burden which does not increase with the problem size.

d	p	n	TSSOS		GN-small		GN-big	
			Certif.	t	Certif.	t	Certif.	t
4	3	255	$3.4 \cdot 10^{-7}$	6	$1.1 \cdot 10^{-2}$	$2 \cdot 10^2$	$4.1 \cdot 10^{-3}$	$1 \cdot 10^3$
	4	624	$2.1 \cdot 10^{-9}$	153	$2.5 \cdot 10^{-2}$	$2 \cdot 10^2$	$3.6 \cdot 10^{-3}$	$1 \cdot 10^3$
	5	1295	Out of memory!	-	$1.8 \cdot 10^{-2}$	$2 \cdot 10^2$	$4.2 \cdot 10^{-3}$	$2 \cdot 10^3$

is the maximum of the Chebychev bound of theorem 3 and the MoM bound of theorem 4. The number of frequencies sampled is $3.2 \cdot 10^7$.

Comparison with TSSOS on the Chebychev basis. We compare GloptiNets with TSSOS on random Chebychev polynomials in table 2, similarly to the comparison with trigonometric polynomials in table 1. Minimizing polynomials defined on the canonical basis is easier: contrary to trigonometric polynomials, there is no need to account for the imaginary part of the variable. If d is the dimension, complex polynomials are encoded in a variable of dimension $2d$ in TSSOS, following the definition of Hermitian Sum-of-Squares introduced in [16]. Hence, the random polynomials we consider are characterized by the dimension d and their number of coefficients n; instead of bounding the degree, we use all the basis elements $H_\omega(\mathbf{x}) = \prod_{\ell=1}^d H_{\omega_\ell}(\mathbf{x}_\ell)$ for which $\|\omega\|_\infty \leqslant p$. The maximum degree is then dp. The RKHS norm of f is fixed to 1. As with the comparison on Trigonometric polynomial table 1, we see that GloptiNets provides similar certificates no matter the number of coefficients in f. Even though it lags behind TSSOS for small polynomials, it handles large polynomials which are intractable to TSSOS. The "small" and "big" models have the same structure as for the trigonometric polynomials experiments.

Sampling from the Bessel distribution. The function $\omega \mapsto e^{-s} I_\omega(s)$ decays rapidly. In fact, with $s = 2$, which is the value used to generate the random polynomials, it falls under machine precision as soon as $\omega > 14$. Thus, we approximate the distribution with a discrete one with weights $I_\omega(s)$ for ω s.t. the result is above the machine precision. We then extend it to multiple dimension with a tensor product. Finally, we use a hash table to store the already sampled frequency, to make the evaluation of million of frequencies much faster. For instance in dimension 5, sampling 10^6 frequencies from the Bessel distribution of parameter $s = 2$ on $\mathbb{N}^5$ yields only $\approx 10^4$ unique frequencies. This allows for tighter certificates, as it makes the r.h.s of eq. (52), in $1/N$, much smaller. Note that the time to generate this hash table is *not* reported in tables 1 and 2, and of the order of a few seconds.

Optimizing a kernel mixture. As it is the case with polynomials, when optimizing a function of the form $h(x) = \sum_{i=1}^n \alpha_i K(x_i, x)$ the certificate provided by GloptiNets only depends on the function norm $\|h\|_{\mathcal{H}}^2$ and not on *e.g.* the number of coefficients n. This is illustrated in fig. 6.

C.4 Fourier coefficients in linear time

Lemma 4 (Fourier coefficient of the Bessel kernel in linear time)**.** *Let g be an extended k-SoS model as in definition 5. Then, its Fourier transform can be evaluated in* linear *time in m with*

$$\widehat{g}_\omega = \sum_{k=1}^r \sum_{n \in \mathbb{Z}^d} \left(\sum_{i=1}^m R_{ki} \prod_{\ell=1}^d \phi_{\ell,-}(z_{i\ell})_{n_\ell} \right) \cdot \left(\sum_{i=1}^m R_{ki} \prod_{\ell=1}^d \phi_{\ell,+}(z_{i\ell}) \right) \tag{75}$$

where

$$\forall n \in \mathbb{Z}, z \in \mathbb{T}, \ell \in [d], \quad \phi_{\ell,\pm}(z)_n = \sqrt{q_{\ell,n}} e^{\pi i (n \pm \omega_\ell) z}$$

and $q_{\cdot,\cdot}(s)$ is defined with lemma 6.

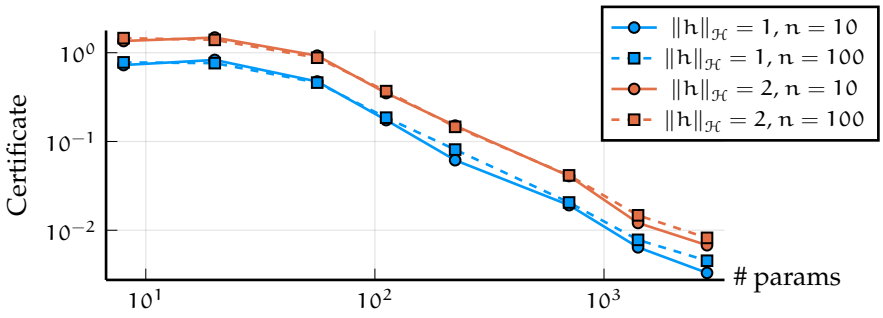


Figure 6: Certificate vs. number of parameters in g when certifying mixture of Bessel functions, characterized by their RKHS norm *(1 in blue, 2 in red)* and their number of coefficients *(10 in circles, 100 in rectangles)*. As with polynomials, this shows that GloptiNets is only sensible to the former, and not to the way the function is represented. We are not aware of other algorithms able to certify this class of functions.

Lemma 4 provides a formula for computing $\widehat{g}_\omega$ which is linear in m, but which still requires numerical approximation to compute the sum on $n \in \mathbb{Z}^d$. For instance, restraining the sum to the hyperbolic cross [9]

$$\mathrm{HC}(d, n) = \left\{ \omega \in \mathbb{Z}^d ; \prod_{\ell=1}^{d} \max\{1, |\omega_\ell|\} \leqslant n \right\}$$

would result in a complexity of $O(n(\log d)^n mr)$ and should produce reasonably accurate estimate of $\widehat{g}_\omega$ for low n.

Furthermore, since q is real-even w.r.t n, the inner-product in eq. (79) can be simplified by computing only half of the terms.

Proof. From lemma 1, we have that

$$\widehat{g}_\omega = \sum_{i,j=1}^{m} R_i^\top R_j \prod_{\ell=1}^{d} e^{-2s_\ell} I_{|\omega_\ell|}(2s_\ell \cos\pi(z_{i\ell} - z_{j\ell})) e^{-i\pi\omega_\ell(z_{i\ell}+z_{j\ell})}. \tag{76}$$

Introducing

$$f_\ell(x, y) = e^{-2s_\ell} I_{|\omega_\ell|}(2s_\ell \cos\pi(x - y)) e^{-i\pi\omega_\ell(x+y)}, \tag{77}$$

eq. (76) simplifies to

$$\widehat{g}_\omega = \sum_{i,j=1}^{m} R_i^\top R_j \prod_{\ell=1}^{d} f_\ell(z_{i\ell}, z_{j\ell}). \tag{78}$$

Using lemma 6, for any $x, y \in \mathbb{T}$,

$$e^{-2s_\ell} I_{|\omega_\ell|}(2s_\ell \cos\pi(x - y)) = \sum_{n\in\mathbb{Z}} q_{\ell,n} e^{\pi i n(x-y)}$$

($q_{\ell,n}$ depends on ω_ℓ) so that, f_ℓ defined in eq. (77) now writes

$$\begin{aligned} f_\ell(x, y) &= \sum_{n\in\mathbb{Z}} q_{\ell,n} e^{\pi i n(x-y)} e^{-\pi i \omega_\ell(x+y)} \\ &= \sum_{n\in\mathbb{Z}} q_{\ell,n} e^{\pi i(n-\omega_\ell)x} e^{-\pi i(n+\omega_\ell)y} \\ &= \phi_{\ell,-}(x) \cdot \phi_{\ell,+}(y) \end{aligned} \tag{79}$$

where, for any $\ell \in \{1, \ldots, d\}$ and $z \in \mathbb{T}$, we defined

$$\phi_{\ell,\pm}(z) = \left(\sqrt{q_{\ell,n}} e^{\pi i(n\pm\omega_\ell)z} \right)_{n\in\mathbb{Z}}. \tag{80}$$

We then define the embedding $\phi_{\pm} : \mathbb{T} \to (\mathbb{Z}^d \to \mathbb{C})$ be the tensor product of the $\phi_{\ell,\pm}$. Then, eq. (79), enables to write $\widehat{g}_\omega$ in eq. (78) as

$$\begin{aligned}
\widehat{g}_\omega &= \sum_{i,j=1}^{m} \sum_{k=1}^{r} R_{ki} R_{kj} \phi_-(\mathbf{z}_i) \cdot \phi_+(\mathbf{z}_j) \\
&= \sum_{k=1}^{r} \left[\sum_{i=1}^{m} R_{ki} \phi_-(\mathbf{z}_i) \right] \cdot \left[\sum_{i=1}^{m} R_{ki} \phi_+(\mathbf{z}_i) \right] \\
&= \sum_{k=1}^{r} \left[\sum_{i=1}^{m} R_{ki} \phi_{1,-}(\mathbf{z}_{i1}) \otimes \cdots \otimes \phi_{d,-}(\mathbf{z}_{id}) \right] \cdot \left[\sum_{i=1}^{m} R_{ki} \phi_{1,+}(\mathbf{z}_{i1}) \otimes \cdots \otimes \phi_{d,+}(\mathbf{z}_{id}) \right] \\
&= \sum_{k=1}^{r} \sum_{n \in \mathbb{Z}^d} \left(\sum_{i=1}^{m} R_{ki} \prod_{\ell=1}^{d} \phi_{\ell,-}(\mathbf{z}_{i\ell})_{n_\ell} \right) \cdot \left(\sum_{i=1}^{m} R_{ki} \prod_{\ell=1}^{d} \phi_{\ell,+}(\mathbf{z}_{i\ell}) \right)
\end{aligned}$$

which is the desired result. □

C.5 Other computation

Lemma 5. *Let f be the function defined on $(-1, 1)$ with*

$$\forall u \in (0, 1/2), \quad f(\cos 2\pi u) = e^{s \cos 2\pi(u-\nu)}. \tag{81}$$

Then, its Chebychev coefficient are given with

$$f_\omega = (1 + \mathbf{1}_{\omega \neq 0}) \cos(2\pi\omega\nu) I_\omega(s). \tag{82}$$

Proof. The $\omega \in \mathbb{N}_*$. The component ω of a function f on the Chebychev basis is given with

$$f_\omega = \frac{2}{\pi} \int_{-1}^{1} f(x) T_\omega(x) \frac{dx}{\sqrt{1-x^2}},$$

which we conveniently rewrite, with the classical change of variable $x = \cos 2\pi u$,

$$f_\omega = 2 \int_{I_1} f(\cos 2\pi u) \cos(2\pi\omega u) du \tag{83}$$

which is valid for any interval $I_1 \subset \mathbb{R}$ of length 1.

Now, for $s > 0$, consider the function f defined on $(-1, 1)$ with $x \mapsto e^{s \cos(\arccos(x) - 2\pi\nu)}$, or equivalently

$$\forall u \in (0, 1/2), \quad f(\cos 2\pi u) = e^{s \cos 2\pi(u-\nu)}. \tag{84}$$

Putting eq. (84) into eq. (83), we obtain

$$\begin{aligned}
f_\omega &= 2 \int_{I_1} e^{s \cos 2\pi(u-\nu)} \cos(2\pi\omega u) du \\
&= 2 \int_{I_1} e^{s \cos 2\pi u} \cos(2\pi\omega(u+\nu)) du \\
&= 2 \int_{I_1} e^{s \cos 2\pi u} \cos(2\pi\omega u) \cos(2\pi\omega\nu) du - 2 \int_{I_1} e^{s \cos 2\pi u} \sin(2\pi\omega u) \sin(2\pi\omega\nu) du.
\end{aligned}$$

The last term is odd, hence integrate to 0 on an interval centered around 0. Hence,

$$f_\omega = 2 \cos(2\pi\omega\nu) \int_{I_1} e^{s \cos 2\pi u} \cos(2\pi\omega u) du. \tag{85}$$

We recognize the definition of the modified Bessel function of the first kind, defined in eq. (57). Plugging this into eq. (85), we obtain

$$f_\omega = 2 \cos(2\pi\omega\nu) I_\omega(s) = 2 I_\omega(s) H_\omega(\cos(2\pi\nu)). \tag{86}$$

If $\omega = 0$, we add a factor $1/2$ into the definition in eq. (83), which yields

$$f_\omega = I_0(s). \tag{87}$$

□

Lemma 6 (Fourier decomposition of Bessel composed with cosine). *Let $s > 0$, $\omega \in \mathbb{N}$ and $z \in \mathbb{T}$. Then,*

$$\begin{aligned} e^{-2s} I_\omega(2s\cos 2\pi z) &= \sum_{n\in\mathbb{Z}} q_{\omega,n} e^{2\pi i n z}, \\ \text{where } \forall n \geqslant 0, q_{\omega,n} &= \begin{cases} e^{-2s} \sum_{p\geqslant (\frac{n-\omega}{2})_+} \frac{(s/2)^{2p+\omega}}{p!(p+\omega)!}\binom{2p+\omega}{p-\frac{n-\omega}{2}} & \text{if } n \equiv \omega, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{88}$$

and $q_{\omega,-n} = q_{\omega,n}$ by evenness of the coefficients.

Proof. From the definition of the modified Bessel function of the first kind [37, p.77, Eq. 2], we have

$$I_\omega(z) = \sum_{p\geqslant 0} \frac{(z/2)^{2p+\omega}}{p!(p+\omega)!},$$

so that

$$\begin{aligned} I_\omega(2s\cos 2\pi z) &= \sum_{p\geqslant 0} \frac{s^{2p+\omega}}{p!(p+\omega)!} \cos(2\pi z)^{2p+\omega} \\ &= \sum_{p\geqslant 0} \frac{(s/2)^{2p+\omega}}{p!(p+\omega)!} \left(e^{2\pi i z} + e^{-2\pi i z}\right)^{2p+\omega} \\ &= \sum_{p\geqslant 0} \frac{(s/2)^{2p+\omega}}{p!(p+\omega)!} \sum_{k=0}^{2p+\omega} \binom{2p+\omega}{k} e^{2\pi i (2(p-k)+\omega) z}. \end{aligned} \tag{89}$$

Using the change of variable $n = 2(p-k)+\omega$ into eq. (89), we see that n has the same parity as ω and

$$I_\omega(2s\cos 2\pi z) = \sum_{p\geqslant 0} \frac{(s/2)^{2p+\omega}}{p!(p+\omega)!} \sum_{\substack{n=-(2p-\omega) \\ n\equiv\omega}}^{2p+\omega} \binom{2p+\omega}{p-\frac{n-\omega}{2}} e^{2\pi i n z}. \tag{90}$$

Equation (90) can be rewritten

$$I_\omega(2s\cos 2\pi z) = \sum_{\substack{n\in\mathbb{Z} \\ n\equiv\omega}} e^{2\pi i n z} \sum_{p\geqslant 0} \frac{(s/2)^{2p+\omega}}{p!(p+\omega)!} \binom{2p+\omega}{p-\frac{n-\omega}{2}} \mathbf{1}_{-(2p+\omega)\leqslant n\leqslant 2p+\omega},$$

for which eq. (88) is a concise rewriting. □

References for part III

[1] Francis Bach and Alessandro Rudi. Exponential convergence of sum-of-squares hierarchies for trigonometric polynomials. *SIAM Journal on Optimization*, 33(3):2137–2159, 2023.

[2] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with Sparsity-Inducing Penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, January 2012. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000015.

[3] Eloïse Berthier, Justin Carpentier, Alessandro Rudi, and Francis Bach. Infinite-Dimensional Sums-of-Squares for Optimal Control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 577–582, December 2022. doi: 10.1109/CDC51059.2022.9992396.

[4] Gaspard Beugnot, Julien Mairal, and Alessandro Rudi. GloptiNets: Scalable Non-Convex Optimization with Certificates. In *Advances in Neural Information Processing Systems*, November 2023.

[5] Dan Bienstock, Mauro Escobar, Claudio Gentile, and Leo Liberti. Mathematical programming formulations for the alternating current optimal power flow problem. *4OR*, 18(3):249–292, September 2020. URL `https://ideas.repec.org/a/spr/aqjoor/v18y2020i3d10.1007_s10288-020-00455-w.html`.

[6] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations, December 2019.

[8] Luc Devroye, Matthieu Lerasle, Gabor Lugosi, and Roberto I. Oliveira. Sub-Gaussian Mean Estimators. *The Annals of Statistics*, 44(6):2695–2725, 2016.

[9] Dinh Dũng, Vladimir N. Temlyakov, and Tino Ullrich. Hyperbolic Cross Approximation. *arXiv:2211.04889*, April 2017.

[10] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023. doi: 10.1017/9781108348973.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[12] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, October 2018.

[13] Didier Henrion, Milan Korda, and Jean-Bernard Lasserre. *The Moment-SOS Hierarchy*, volume 4 of *Optimization and Its Applications*. World Scientific Publishing Europe Ltd., December 2020. doi: 10.1142/q0252.

[14] Joseph J. Hilling and Anthony Sudbery. The geometric measure of multipartite entanglement and the singular values of a hypermatrix. *Journal of Mathematical Physics*, 51(7):072102, July 2010.

[15] Reiner Horst and Panos M Pardalos. *Handbook of global optimization*, volume 2. Springer Science & Business Media, 2013.

[16] Cédric Josz and Daniel K. Molzahn. Lasserre Hierarchy for Large Scale Polynomial Optimization in Real and Complex Variables. *SIAM Journal on Optimization*, 28(2):1017–1048, January 2018.

[17] Jean B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, January 2001. doi: 10.1137/S1052623400366802.

[18] Jean Bernard Lasserre. *Moments, Positive Polynomials and Their Applications*, volume 1 of *Series on Optimization and Its Applications*. October 2009. doi: 10.1142/p665.

[19] Monique Laurent and Lucas Slot. An effective version of schmüdgen's positivstellensatz for the hypercube. *Optimization Letters*, September 2022. doi: 10.1007/s11590-022-01922-5.

[20] Ngoc Hoang Anh Mai, J. B. Lasserre, Victor Magron, and Jie Wang. Exploiting Constant Trace Property in Large-scale Polynomial Optimization. *ACM Transactions on Mathematical Software*, 48(4):40:1–40:39, December 2022.

[21] Ulysse Marteau-Ferey. *Modelling Functions with Kernels: From Logistic Regression to Global Optimization*. PhD thesis, PSL, September 2022.

[22] Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Non-parametric Models for Non-negative Functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12816–12826. Curran Associates, Inc., 2020.

[23] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826 (1989):37, 1989.

[24] Boris Muzellec, Adrien Vacher, Francis Bach, François-Xavier Vialard, and Alessandro Rudi. Near-optimal estimation of smooth transport maps with kernel sums-of-squares. *arXiv:2112.01907*, December 2021.

[25] Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 2016. doi: 10.1017/CBO9781316219232.

[26] Alessandro Rudi and Carlo Ciliberto. PSD Representations for Effective Probability Models. In *Advances in Neural Information Processing Systems*, volume 34, pages 19411–19422. Curran Associates, Inc., 2021.

[27] Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding Global Minima via Kernel Approximations. *arXiv:2012.11978*, December 2020.

[28] Walter Rudin. The Basic Theorems of Fourier Analysis. In *Fourier Analysis on Groups*, chapter 1, pages 1–34. John Wiley & Sons, Ltd, 1990. doi: 10.1002/9781118165621.ch1.

[29] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[30] Pascal Van Hentenryck. Machine Learning for Optimal Power Flows. *INFORMS Tutorials in Operations Research*, October 18.

[31] Peter JM Van Laarhoven, Emile HL Aarts, Peter JM van Laarhoven, and Emile HL Aarts. *Simulated annealing*. Springer, 1987.

[32] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of Squares and Semidefinite Program Relaxations for Polynomial Optimization Problems with Structured Sparsity. *SIAM Journal on Optimization*, 17(1):218–242, January 2006. doi: 10.1137/050623802.

[33] Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. Phase Recovery, Max-Cut and Complex Semidefinite Programming, July 2013.

[34] Jie Wang and Victor Magron. Exploiting Sparsity in Complex Polynomial Optimization. *Journal of Optimization Theory and Applications*, 192(1):335–359, January 2022.

[35] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Chordal-TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity with Chordal Extension. *SIAM Journal on Optimization*, 31(1):114–141, January 2021. doi: 10.1137/20M1323564.

[36] Jie Wang, Victor Magron, and Jean-Bernard Lasserre. TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity. *SIAM Journal on Optimization*, 31(1):30–58, January 2021. doi: 10.1137/19M1307871.

[37] G. N. Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press, 1922.

[38] Blake Woodworth, Francis Bach, and Alessandro Rudi. Non-Convex Optimization with Certificates and Fast Rates Through Kernel Sums of Squares. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 4620–4642. PMLR, June 2022.

Conclusion

In this thesis, we explored how established kernel method tools can help understand the fundamental aspects of machine learning: generalization and optimization.

In section 4, we revealed that not all forms of regularization contribute equally to generalization. Regularization forms a critical bridge between generalization and optimization in traditional convex machine learning. We discovered that certain optimization procedures, specifically the proximal point algorithm, possess advantageous statistical properties when learning with kernels. Moving to section 5, our focus shifted to the non-convex domain of neural networks, which exhibit a complex relationship between optimization and generalization. Here, we found that improved optimization doesn't necessarily lead to better generalization—in fact, often the opposite in practice. We introduced a simpler convex model in Hilbert spaces that mirrors neural network behavior, providing insights into the more complex dynamics of neural networks. section 9 then delved into optimizing non-convex functions with certificates, utilizing the recent advancements in kernel sum of squares (K-SoS). This approach successfully optimizes functions that are challenging for other methods, particularly those with a low RKHS norm.

Future work directions, as outlined at the end of section 4, include extending spectral filter theory to self-concordant loss functions. This extension could enhance our understanding of the statistical capabilities of various machine learning algorithms. From section 5, we suggest pursuing the development of new optimization methods that prioritize generalization. The challenge lies in the difficulty of analyzing gradient descent paths in neural networks. However, insights from convex models might offer valuable heuristics. Lastly, part III presents open problems in global optimization, a field ripe for innovative contributions. There remains much to explore in effectively applying K-SoS to global optimization. The open problems listed, particularly in open problems 1 to 6, are especially promising. They offer the potential to integrate K-SoS into the existing toolkit for global optimization, heralding new possibilities in this area.

RÉSUMÉ

Cette thèse explore le lien entre la généralisation et l'optimisation dans l'apprentissage machine via les méthodes à noyaux reproduisant. Nous améliorons les bornes d'excès de risque pour les fonctions auto-concordantes, étendant ainsi la théorie du filtrage spectral à une plus grande famille de fonctions, et obtenons des taux d'apprentissage plus rapides. Un phénomène de généralisation supérieure avec de grands pas de gradients dans l'entraînement de réseaux de neurones est également étudié, avec un modèle intuitif dans un espace de Hilbert. Enfin, nous présentons un algorithme pour certifier l'optimalité sur des minima de fonctions non-convexes, utilisant les sommes de carrés de noyaux reproduisant pour traiter des problèmes inaccessibles par les méthodes existantes.

MOTS CLÉS

Noyaux reproduisants, Optimisation, Apprentissage statistique

ABSTRACT

This thesis delves into the relationship between generalization and optimization in machine learning, focusing on kernel methods. We enhance excess risk bounds for generalized self-concordant functions, including logistic loss, expanding the application of spectral filter theory beyond traditional uses. This results in estimators with faster, optimal learning rates for a broader problem set. We also explore the counterintuitive benefit of high learning rates in neural network training with square loss, demonstrating through a convex function model in Hilbert space that these step sizes improve generalization in various tasks. Lastly, we introduce an algorithm to certify optimality in minimizing non-convex functions on tori or hypercubes, using kernel sum-of-squares to address problems beyond the reach of current methods.

KEYWORDS

Kernel methods, Optimisation, Statistical Learning