Betting on Sparsity: Leveraging Hidden Linear Features through Regularisation for Supervised Learning

Bertille Follain

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure et INRIA

Betting on Sparsity: Leveraging Hidden Linear Features through Regularisation for Supervised Learning

Miser sur la Parcimonie : Exploitation des Dépendances Linéaires Cachées via la Régularisation pour l'Apprentissage Supervisé

Soutenue par

Bertille Follain

Le 21 Novembre 2024

École doctorale n°386

École doctorale de Sciences Mathématiques de Paris Centre

Spécialité

Informatique

Préparée au

DIENS, Équipe-projet SIERRA

Composition du jury :

Claire BOYER Professeure, Université Paris-Saclay	*Présidente du jury* *Examinatrice*
Christophe GIRAUD Professeur, Université Paris-Saclay	*Rapporteur*
Matthieu LERASLE Professeur, ENSAE	*Rapporteur*
Gabriel PEYRÉ Directeur de recherche, CNRS	*Examinateur*
Francis BACH Directeur de recherche, INRIA	*Directeur de thèse*

ENS | PSL

Inria

Remerciements

Quelle émotion de contempler aujourd'hui le chemin parcouru depuis le début de cette aventure qu'est la thèse, chemin que l'on n'accomplit jamais seul. C'est donc le cœur empli de grâce et de reconnaissance que je me tourne maintenant vers celles et ceux qui m'ont accompagnée.

D'abord, je remercie bien évidemment Matthieu Lerasle et Christophe Giraud, qui se sont attelés à la délicate tâche de rapporteur, et Claire Boyer et Gabriel Peyré qui ont accepté d'être membres de mon jury. Sans eux, le précieux sésame ne pourrait m'être délivré!

Un immense merci à Francis, qui m'a guidée, épaulée, poussée dans mes retranchements..., et qui restera définitivement le personnage le plus mémorable de ce chapitre de ma vie. Comment exprimer l'admiration que l'on peut avoir pour un tel monument d'intelligence, d'humour, mais surtout de cyclisme ?

Je tiens également à remercier tous ceux qui ont fait des équipes Willow et Sierra un environnement si agréable où travailler. Merci particulièrement à mes charmants co-bureaux Antoine, Gaspard et Ziad. Merci à l'équipe Argo, notamment Mathieu et Luca, dont les visites intempestives ont égayé mes journées. Merci à Céline pour nos joyeuses conversations et à Théophile de m'avoir fait rire si souvent. Merci à tout le personnel de l'INRIA et de l'ENS qui rendent notre travail de recherche possible.

Je remercie aussi ceux qui ont cru en moi à différentes étapes de ma scolarité, alors que je n'avais pas toujours les meilleures cartes en main. Merci à Mathias Weislinger d'avoir partagé avec moi son amour des mathématiques, à Jeremie Llodra-Perez et Yves Duval d'avoir eu confiance en mes capacités, à Éric Moulines de m'avoir fait découvrir les statistiques et présenté Francis. Un immense merci à Richard Samworth et Tengyao Wang d'avoir été mes premiers co-auteurs et de m'avoir montré à quel point la recherche pouvait être belle. Je garde un superbe souvenir de cette collaboration si *British*.

Qu'ils viennent de ma Normandie natale, de la HX3, de l'ENS, de la X17 ou d'ailleurs, j'ai la chance d'être entourée par des amis incroyables. Merci donc à chacun d'entre eux de constituer l'une des plus grandes fiertés de ma vie. Merci en particulier à Alice, Émile et Titouan, pour leur soutien sans faille dans cette aventure et dans toutes les autres.

Je remercie évidemment de tout cœur ma famille pour son amour et son soutien et ma belle-famille pour son accueil chaleureux. Merci à mon père de m'avoir transmis sa soif de connaissance. Merci à ma mère d'avoir œuvré pour moi contre vents et marées. Merci à mon beau-père pour ses encouragements, à ma sœur pour sa sérénité. Merci à mon frère pour son sens du devoir. Merci à mes grands-parents, qui m'ont tendrement aimée.

Enfin, merci à Antonin, qui a bouleversé ma vie en faisant de chaque jour une fête.

Résumé

Nous considérons le problème de l'apprentissage supervisé lorsqu'il existe des structures de données cachées, en nous concentrant sur les cas où quelques caractéristiques linéaires pertinentes expliquent la relation entre la réponse et les covariables, comme dans le modèle "multi-index". Notre objectif est de développer des méthodes qui exploitent ces structures cachées pour améliorer l'apprentissage. De nombreuses approches existantes reposent sur des hypothèses fortes concernant la génération de données et se heurtent à la malédiction de la dimensionnalité, présentant souvent une dépendance exponentielle en la dimension des données. Nous explorons les modèles "multi-index" en utilisant la minimisation du risque empirique régularisé (ERM), car ce cadre flexible est applicable à tout problème pour lequel un risque peut être défini. Tout au long de cette thèse, nous explorons trois méthodes innovantes pour simultanément apprendre les caractéristiques et estimer la fonction de prédiction dans un contexte non-paramétrique. Chaque méthode intègre des éléments des espaces de Hilbert à noyau reproduisant (RKHS), contient des pénalités d'apprentissage des caractéristiques qui sont adaptables à la sélection de variables, utilise des procédures d'optimisation basées sur la repondération pour un calcul efficace et s'appuie sur des hypothèses limitées sur le mécanisme de génération des données. Nous avons veillé à la facilité d'utilisation du code développé et à la reproductibilité des expériences. La première méthode, KTNGrad, considère l'ERM dans un RKHS avec une pénalité de norme nucléaire sur la matrice empirique des gradients. L'analyse théorique montre que KTNGrad a des taux de convergence pour le risque attendu dans les contextes bien spécifiés qui ne dépendent pas exponentiellement de la dimension, tout en estimant l'espace des caractéristiques pertinentes d'une manière sûre. La deuxième méthode, RegFeaL, exploite les propriétés d'orthogonalité et d'invariance par rotation des polynômes de Hermite. Cette méthode fait pivoter les données de manière itérative pour les aligner avec les caractériques. Le risque attendu converge vers le risque minimal avec des taux explicites, sans hypothèses fortes sur la véritable fonction de régression. Enfin, la troisième méthode, BKerNN, introduit un nouveau modèle qui combine les méthodes à noyaux et les réseaux de neurones. Cette méthode optimise les poids de la première couche par descente de gradient tout en ajustant explicitement la non-linéarité et les poids de la deuxième couche. L'optimisation tire parti de l'homogénéité positive du noyau Brownien, et l'analyse de la complexité de Rademacher montre que le risque attendu de BKerNN atteint des taux de convergence favorables qui sont indépendants de la dimension, sans hypothèses fortes sur la véritable fonction de régression ou sur les données.

Mots clés : apprentissage de caractéristiques, noyau reproduisant, minimisation du risque empirique régularisé, parcimonie, apprentissage supervisé, réseaux de neurones

Abstract

We tackle the challenge of supervised learning with hidden data structures, focusing on cases where a few relevant linear features explain the relationship between response and covariates, as in the multi-index model. We aim to develop methods that leverage these hidden structures to improve learning. Many existing approaches rely on strong assumptions about data generation and struggle with the curse of dimensionality, often exhibiting exponential dependency on data dimension. We explore multi-index models through regularised empirical risk minimisation (ERM), as this flexible framework is applicable to any problem where a risk can be defined. Throughout this thesis, we explore three innovative methods for joint feature learning and function estimation in nonparametric learning. Each method integrates elements from reproducing kernel Hilbert spaces (RKHS), contains sparsity-inducing penalties for feature learning which are adaptable to the variable selection setting, uses optimisation procedures based on reweighting for efficient computation and relies on limited assumptions on the data-generating mechanism. We ensured the usability of the developed code and the reproducibility of the experiments. The first method, KTNGrad, considers ERM within an RKHS, augmented by a trace norm penalty on the sample matrix of gradients. Theoretical analysis shows that KTNGrad achieves convergence rates that do not depend exponentially on the dimension for the expected risk in well-specified settings while recovering the underlying feature space in a safe-filter manner. The second method, RegFeaL, leverages Hermite polynomials' orthogonality and rotation invariance properties. This method iteratively rotates the data to align with leading directions. The expected risk converges to the minimal risk with explicit rates without strong assumptions on the true regression function. Finally, the third method, BKerNN, introduces a novel framework that combines kernel methods and neural networks. This method optimises the first layer's weights via gradient descent while explicitly adjusting the non-linearity and weights of the second layer. The optimisation leverages the positive homogeneity of the Brownian kernel, and Rademacher complexity analysis shows that BKerNN achieves favourable convergence rates that are dimension-independent without strong assumptions on the true regression function or the data.

Keywords : feature learning, reproducing kernel, regularised empirical risk minimisation, sparsity, supervised learning, neural networks

Contents

CHAPTER 1

General Introduction

Goal

The aim of this chapter is to establish the theoretical framework and motivation for this thesis, situating our work within the broader context of current research. We begin with a review of key concepts in supervised learning and learning theory, with a particular emphasis on finite-sample generalisation guarantees, which are essential for evaluating the performance of the proposed algorithms. The chapter then explores the challenges posed by high-dimensional data, particularly focusing on the use of sparsity assumptions to manage and reduce the complexity of learning problems. Subsequently, we turn our attention to the multi-index model, providing a comprehensive review of existing methodologies in this domain. The chapter concludes with a summary and analysis of the three main contributions of this thesis.

Contents

1 Background

In an era characterised by the exponential growth of data and its increasing complexity, the ability to extract meaningful insights has become critical. High-dimensional data, which is prevalent across various fields such as genomics, finance, and image processing, presents unique challenges that often overwhelm traditional learning methods. The curse of dimensionality, a phenomenon where the volume of the feature space grows exponentially with the number of dimensions, often impedes the effectiveness of these methods. This can lead to models that are either overly complex and prone to overfitting or too simplistic to accurately capture the underlying structure of the data.

This thesis, titled *Betting on Sparsity: Leveraging Hidden Linear Features through Regularisation for Supervised Learning*, investigates innovative methods that harness the power of sparsity and regularisation to enhance the performance of learning algorithms. By focusing on sparsity-driven techniques, we aim to exploit hidden linear structures within high-dimensional data, such as those found in the multi-index model, ultimately leading to more effective and interpretable models. Regularisation techniques, such as the LASSO, play a critical role in this context by promoting sparsity in the model parameters, thereby mitigating the risks associated with high-dimensional data. While the LASSO is for linear prediction, we tackle non-linear settings and, more specifically, non-parametric prediction. We also consider variable selection as a by-product of the feature learning framework.

The introduction begins with a review of relevant concepts in supervised learning, followed by a discussion of the challenges posed by high-dimensional data and the motivation for adopting sparsity-based approaches. We then examine existing methods in the literature, setting the stage for the novel contributions presented in this work, which are summarised in the final section of the introduction.

1.1 Supervised Learning Methods

Supervised learning is a fundamental paradigm in machine learning where the objective is to learn a function that maps inputs to outputs using training data. Given a dataset $(x_i, y_i)_{i\in[n]}$, where $[n] := \{1, \dots, n\}$ and each $x_i \in \mathcal{X} \subset \mathbb{R}^d$ represents an input vector and $y_i \in \mathcal{Y} \subset \mathbb{R}$ is the corresponding output or label, the goal is to find a function $f : \mathcal{X} \to \mathcal{Y}$ that accurately predicts the output y for a new input x.

Supervised learning methods can be broadly categorised into parametric and non-parametric approaches, both of which rely heavily on the principle of empirical risk minimisation (ERM) [Vapnik, 1991].

Empirical risk minimisation and regularisation. In supervised learning, the ultimate objective can be formulated as finding a function f that minimises the *expected risk* $\mathcal{R}(f)$, defined as the expected value of the loss function over the joint distribution of the data

$$\mathcal{R}(f) = \mathbb{E}_{(X,Y)}\left(\ell(Y, f(X))\right),$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a loss function that quantifies the discrepancy between the predicted value $\hat{y} = f(x)$ and the true label y. However, since the joint distribution is generally unknown, direct minimisation of the expected risk is infeasible.

Instead, the expected risk is approximated by the *empirical risk*, which is the average

loss over the training dataset $(x_i, y_i)_{i\in[n]}$

$$\widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)).$$

The principle of empirical risk minimisation (ERM) is to find a function $\hat{f}$ that minimises this empirical risk:

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f),$$

where $\mathcal{F}$ represents the hypothesis space, i.e., the set of candidate functions that the learning algorithm can explore [Vapnik, 2013].

ERM provides a framework for approximating the best predictor within a given function class, but it does not guarantee optimal performance on unseen data, leading to a *generalisation gap*, the difference between the empirical risk and the true risk. The choice of hypothesis space $\mathcal{F}$ is critical: a complex space may lead to overfitting, while a simpler one might result in underfitting. Additionally, the choice of loss function ℓ has a significant impact on the optimisation process, sensitivity to outliers, and the interpretability of the results.

Regularisation techniques are essential in mitigating the generalisation gap by penalising model complexity, thereby preventing overfitting while ensuring that the model remains sufficiently flexible to capture the underlying data patterns. In the context of empirical risk minimisation, regularisation is incorporated directly into the optimisation problem, leading to the formulation of *regularised empirical risk minimisation* (RERM). The RERM approach modifies the original ERM objective by adding a regularisation term to the empirical risk, which penalises the complexity of the function f. This can be expressed as

$$\hat{f}_\lambda = \arg\min_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) + \lambda\Omega(f),$$

where $\lambda > 0$ is a regularisation parameter that controls the trade-off between the empirical risk and the regularisation term, and $\Omega(f)$ is the regularisation penalty that typically enforces smoothness or sparsity in the function f.

The choice of λ is pivotal: an excessively large λ can impose too stringent a constraint on the model, leading to underfitting, whereas a very small λ might fail to adequately penalise model complexity, thereby increasing the risk of overfitting. To address this, methods such as cross-validation are frequently employed to determine the optimal value of λ [Arlot and Celisse, 2010]. Regularisation proves to be particularly powerful in high-dimensional settings, where the potential for overfitting is amplified due to the vast number of parameters relative to the available data [Bishop, 2006].

(Regularised) empirical risk minimisation is a powerful framework as it can be applied to a large variety of problems beyond typical i.i.d covariates/response pairs as long as a risk can be defined.

Parametric methods. Parametric methods assume that the function f can be parameterised by a finite set of parameters $\theta \in \mathbb{R}^p$, with f typically written as $f(x) = f(x;\theta)$, with $x \in \mathbb{R}^d$. The task is to estimate the parameters θ that minimise the empirical risk [Bishop, 2006].

A classic example of a parametric method is linear regression, where the model is defined as

$$f(x;\theta) = \theta^\top x,$$

and the parameters θ are estimated by minimising the empirical risk with respect to the square loss

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta^\top x_i)^2.$$

It is common to regularise by the ℓ_2 norm of θ, yielding ridge regression, which penalises large weights. This encourages the model to distribute the weight more evenly, reducing the risk of overfitting by preventing any single feature from having an excessively large influence on the predictions [Gruber, 1998].

Another powerful class of parametric methods includes neural networks, particularly feedforward networks, which consist of layers of linear transformations followed by non-linear activation functions [Goodfellow et al., 2016]. A one-hidden-layer neural network (also known as a single-layer perceptron) can be represented as

$$f(x; \theta) = \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j),$$

where $w_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ are parameters for the j-th neuron in the hidden layer, $\eta_j \in \mathbb{R}$ are the weights for the output layer, σ is a non-linear activation function (e.g., ReLU for rectified linear unit, $\sigma(z) = \max(0, z)$), and m is the number of neurons in the hidden layer.

The parameters $\theta = (w_j, b_j, \eta_j)_{j=1}^m$ are typically learned through back-propagation and stochastic gradient descent (SGD), where the empirical risk is minimised iteratively by updating the parameters in the direction of the negative gradient of the loss function with respect to the parameters [Rumelhart et al., 1986].

Neural networks have demonstrated exceptional success in capturing complex, non-linear relationships in data, as they are capable of approximating any continuous function given a sufficient number of neurons and layers (as per the universal approximation theorem) [Hornik et al., 1989]. However, this expressiveness often comes at the cost of interpretability and computational efficiency. Moreover, training neural networks requires large amounts of data to avoid overfitting, and tuning hyperparameters such as the learning rate, number of neurons, and network architecture can be challenging. Regularisation techniques and careful architecture design are crucial for controlling model complexity and maintaining generalisation to unseen data [Krogh and Hertz, 1991].

Non-parametric methods. Non-parametric methods differ from parametric methods in that they do not assume a fixed form for the function f. Instead, they allow the model complexity to grow with the size of the training data, enabling these methods to adapt to a broader range of functions [Hastie et al., 2001]. This flexibility makes non-parametric methods particularly useful when the underlying relationship between x and y is highly non-linear or unknown.

A classic example of a non-parametric method is the k-nearest neighbours (k-NN) algorithm, which, for a given input x, predicts the output by averaging the outputs of the k closest points in the training set [Cover and Hart, 1967]. While simple and effective for small datasets, k-NN suffers from the curse of dimensionality, where the concept of "closeness" becomes less meaningful as the number of dimensions increases.

A more sophisticated and widely used non-parametric approach, which plays a central role in this thesis, involves reproducing kernel Hilbert spaces (RKHS) [Aronszajn, 1950]. An RKHS is a Hilbert space of functions where the evaluation of a function at any point can be represented as an inner product with a fixed function known as the kernel. The

key feature of an RKHS is the reproducing property

$$f(x) = \langle f, k_x \rangle_{\mathcal{H}},$$

where $k_x = k(x, \cdot)$ with k the reproducing kernel associated with the space $\mathcal{H}$, and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product in $\mathcal{H}$. Popular choices for the kernel function include the Gaussian (RBF) kernel, $k(x, x') = \exp(-\|x - x'\|^2/2)$.

The strength of the RKHS framework lies in the representer theorem [Aronszajn, 1950, Schölkopf and Smola, 2002], which elegantly simplifies the solution to a regularised risk minimisation problem. According to this theorem, any solution to such a problem can be represented as a finite linear combination of kernel functions evaluated at the training points. In practical terms, this implies that instead of searching over an infinite-dimensional space of functions, the optimisation can be reduced to finding the optimal coefficients $\alpha \in \mathbb{R}^n$ in the expression $\hat{f}(x) = \sum_{i=1}^n \alpha_i k_{x_i}$. This idea can even be extended to more general settings with convex regularisation [Boyer et al., 2019]. The reduction induced by the representer theorem not only makes the computation of the estimator more tractable but also leads to an efficient implementation of the kernel ridge regression (KRR) problem, which uses the features induced by k instead of the original features as in ridge regression

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \widehat{\mathcal{R}}(f) + \lambda \|f\|_{\mathcal{H}}^2,$$

where $\lambda > 0$ is the regularisation parameter. This method is inherently non-parametric, as the complexity of the model, determined by the number of effective parameters, naturally increases with the size of the dataset [Wahba, 1990].

Neural networks can also be interpreted as non-parametric methods, as the number of neurons in the hidden layer, and consequently the number of parameters, can be scaled with the sample size. In the infinite-width limit, where the number of neurons is considered infinite, which consists in replacing the sum of the output layer with an integral, the method becomes explicitly non-parametric.

1.2 Learning Theory

Learning theory provides a formal framework for evaluating the generalisation capabilities of supervised learning algorithms, therefore assessing their overall performance. The key question is how well a model trained on a finite dataset can predict outcomes on unseen data, typically quantified by the generalisation error, i.e., the difference between the expected risk (true risk) and the minimal expected risk achievable by any function within the hypothesis class. To analyse this error, researchers often examine the generalisation gap, the discrepancy between empirical and expected risk, using tools such as Rademacher complexity.

Generalisation bounds. Generalisation bounds provide high-probability guarantees on a model's performance on unseen data. This can be done by quantifying the generalisation gap, i.e., the difference between the expected risk and the empirical risk. A common form of these bounds is then given by

$$\left| \mathcal{R}(f) - \widehat{\mathcal{R}}(f) \right| \leq \epsilon(n, d, \delta) \quad \text{with probability } \geq 1 - \delta,$$

where $\mathcal{R}(f)$ denotes the expected risk, $\widehat{\mathcal{R}}(f)$ represents the empirical risk, and $\epsilon(n, d, \delta)$ depends on the sample size n, the dimensionality d, and the confidence level $1 - \delta$.

Another important type of generalisation bound focuses on providing high-probability guarantees for the difference between the expected risk of an estimator $\hat{f}$ and the minimal expected risk achievable within the hypothesis class $\mathcal{F}$, expressed as $\mathcal{R}(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}(f)$. These bounds are particularly valuable because they offer insights into how closely the learned model approximates the best possible model within the given class, thus directly linking the estimator's performance to the inherent limitations of the hypothesis space.

The strength of both of these types of bounds lies in their explicit dependency on the key parameters of the learning problem, such as the dimensionality d and the sample size n. Unlike asymptotic convergence results, which, for example, assure that the empirical risk converges to the expected risk only as the sample size approaches infinity, these bounds offer concrete, quantitative insights into how the generalisation error behaves in finite-sample settings. This detailed understanding helps elucidate the interplay between model complexity, data dimensionality, and sample size, thereby enabling a more informed analysis of how these factors influence the model's performance on unseen data in practical cases [Reid, 2010].

Rademacher complexity. Rademacher complexity is a powerful tool for deriving generalisation bounds by measuring the capacity of a function class $\mathcal{G}$ to fit random noise or binary labels, thereby indicating its potential to overfit. Given a dataset of covariates only $(x_i)_{i \in [n]}$, the empirical Rademacher complexity is defined as the following expectation

$$\widehat{R}_n(\mathcal{G}) = \mathbb{E}_\varepsilon \left(\sup_{f \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i f(x_i) \right),$$

where ε_i are independent Rademacher random variables taking values in $\{-1, +1\}$ with equal probability [Bartlett and Mendelson, 2002]. This complexity can also be extended to include the dataset and the loss function. For the dataset, the Rademacher complexity $R_n(\mathcal{G})$ is defined as the expectation over the dataset of $\widehat{R}_n(\mathcal{G})$. For the loss, the presented form can be retrieved via the contraction principle for Lipschitz-continuous losses [Bach, 2024, Section 4.5.2], see Geoffrey et al. [2020] for a study of high-dimensional learning with convex and Lipschitz losses using extensions of Rademacher complexities.

The Rademacher complexity of $\mathcal{G}$ is crucial for deriving data-dependent generalisation bounds that adapt to the hypothesis class's complexity relative to the data. As an example, for binary error functions, and for any function class $\mathcal{G}$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following bound holds

$$\sup_{f \in \mathcal{G}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) \leq 2R_n(\mathcal{G}) + 4\sqrt{\frac{2\log(4/\delta)}{n}},$$

as shown by Shalev-Shwartz and Ben-David [2014, Chapter 26], see also Boucheron et al. [2005]. This result highlights that the generalisation gap depends both on the empirical risk and the complexity of the considered function class.

Curse of dimensionality. One of the major challenges in supervised learning is dealing with high-dimensional data. As the number of features or dimensions increases, the data points become sparse in the feature space, making it increasingly difficult for learning algorithms to generalise from training data to unseen examples. This phenomenon affects various supervised learning methods differently and is known as the curse of dimensionality [Bellman, 1966, Giraud, 2014].

The curse is evident in the dependency of generalisation bounds on the dimensionality, where, for instance, the rates may take the form $O(n^{-1/d})$. This implies an exponential growth in the required number of samples as the dimension increases in order to maintain comparable performance. This is typically the case of k-nearest neighbours and other local-averaging methods [Bach, 2024, Chapter 6]. In practice, this means that some methods fail to perform effectively in high-dimensional settings, in addition to the increased computational complexity they entail.

Well-specified models and adaptivity. In well-specified models, it is assumed that the true underlying prediction function, f^*, which minimises the expected risk over all possible functions, is contained within the hypothesis space $\mathcal{F}$. Under this assumption, methods like kernel ridge regression can achieve convergence rates that are independent of the data dimension, effectively circumventing the curse of dimensionality. However, this benefit relies on the strong assumption that f^* is indeed within $\mathcal{F}$. For instance, when using a Sobolev kernel, this assumption implies that the function must be highly regular, with square-integrable derivatives up to at least the order $d/2$ [Bach, 2024, Chapter 7].

In contrast, when dealing with misspecified models where f^* does not belong to $\mathcal{F}$, no function within the hypothesis space can attain the true minimal expected risk. Instead, the learning algorithm converges to the best possible approximation within $\mathcal{F}$, but an unavoidable gap known as the *approximation error* persists between the achievable risk and the true minimal risk. This error reflects the inherent limitations of the chosen hypothesis class and cannot be closed by merely increasing the sample size or adjusting regularisation parameters [Bach, 2024, Section 4.3].

Nonetheless, even in the face of misspecification, certain methods can exhibit adaptivity, meaning that the rates improve depending on properties of f^*. For example, kernel ridge regression is adaptive to intermediate regularity of the underlying function f^* [Bach, 2024, Chapter 7], while neural networks can even be adaptive to the presence of hidden linear features, with rates depending on the number of relevant linear features instead of the original dimension of the data [Bach, 2024, Chapter 9].

1.3 Sparsity Assumptions

In high-dimensional settings, sparsity assumptions are practical because not all the information contained in the covariates is likely relevant to the prediction task. By focusing on a small subset of key variables or linear features, sparsity reduces the problem's dimensionality, preventing overfitting, enhancing computational efficiency, and improving the model's generalisation ability while identifying important predictors [Hastie et al., 2015].

Dependency on a few variables. It is often reasonable to assume that the response variable depends on only a small subset of the covariates. Within the linear regression framework, this assumption translates to the parameter vector θ having many zero entries, meaning that $\|\theta\|_0$ is small, where $\|\theta\|_0$ counts the number of non-zero elements in θ.

However, directly minimising the $\|\theta\|_0$ norm is computationally infeasible due to its combinatorial nature, making the optimisation problem NP-hard [Natarajan, 1995]. To circumvent this, a common approach is to employ a convex relaxation of the $\|\theta\|_0$ norm by using the ℓ_1 norm $\|\theta\|_1 = \sum_{a=1}^{d} |\theta_a|$ instead. This relaxation leads to the formulation of the LASSO (Least Absolute Shrinkage and Selection Operator) [Tibshirani, 1996]

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \theta^\top x_i)^2 + \lambda \|\theta\|_1 \right).$$

The LASSO is particularly effective in high-dimensional settings where the number of variables d is large, as it simultaneously performs parameter estimation and variable selection. By introducing an ℓ_1-norm penalty, the LASSO encourages many of the coefficients to shrink towards zero, effectively excluding irrelevant features from the model, which improves both prediction performance and interpretability. However, the model remains inherently linear, which limits its ability to capture more complex patterns in the data.

Similarly, in the context of high-dimensional changepoint estimation with heterogeneous missing data, one can assume that the changepoint only occurs in a subset of the coordinates. This sparsity assumption is incorporated into the model through an ℓ_1 penalty in the optimisation problem, as demonstrated in Follain et al. [2022].

The multi-index model. The multi-index model takes sparsity one step further by assuming in the context of supervised learning that the relationship between the response and covariates can be explained by a limited number of linear combinations of the input variables. Formally, a multi-index model is given by

$$\forall x \in \mathcal{X}, \quad f^*(x) = g^*(P^\top x)$$

where f^* is the true regression function, $P \in \mathbb{R}^{d \times s}$ is a matrix with $s < d$ (typically $s \ll d$) where s is the number of relevant linear features and g^* is an unknown potentially non-linear link function that maps these linear combinations to the response variable. This model, formally presented by Li [1991], provides a flexible framework for capturing complex relationships in data while maintaining a focus on a reduced, interpretable set of features. The estimation of multi-index models can be challenging due to the non-linearity and non-parametric nature of the link function g^*, as well as the difficulty of potentially estimating the number of relevant features s.

2 Existing Methods for the Multi-Index Model

This thesis focuses on the exploitation of hidden linear features within the context of supervised learning, as in the multi-index model. To lay a solid groundwork for the contributions presented here, we first review the extensive body of literature that has emerged on the topic. It is important to note that the objectives of the methods reviewed here may differ from the ones we propose. While the existing methods primarily focus on recovering the subspace spanned by the matrix P and sometimes estimating its dimension, our approach is more comprehensive, aiming to simultaneously estimate both the function f^* and the matrix P, as well as its dimension, with a theoretical emphasis on the theoretical behaviour of the expected risk. Broadly speaking, the existing methods for estimating multi-index models can be classified into two main categories: moment-based methods and optimisation-based methods.

2.1 Moment-Based Methods

One of the foundational approaches to estimating multi-index models is rooted in moment-based methods, where specific moments of the data are exploited to eliminate the influence of the unknown link function g^*. The seminal work by Brillinger [2012] in the context of a single linear feature, also known as the single-index model, introduced the idea that for centred Gaussian data with an identity covariance matrix, the expectation $\mathbb{E}(YX)$ is pro-

portional to the feature vector. This insight laid the groundwork for further developments in the estimation of multi-index models.

To extend these ideas beyond Gaussian data, Stoker [1986] proposed the use of the score function, defined as the gradient of the log-likelihood with respect to the parameters in a parametric family of distributions, in place of the covariate vector X in the expectation. This innovation broadened the applicability of the method to distributions with a differentiable log density. However, a significant challenge remains: the score function must be known or estimated, which is itself a non-trivial task. Moreover, this approach does not generalise from the single-index to the multi-index model.

Sliced Inverse Regression (SIR), introduced by Li [1991], represents a key advancement in extending moment-based methods to multi-index models. SIR inverts the regression problem, investigating how the predictors X relate to the response Y rather than the reverse. By dividing the response variable into slices and examining the conditional expectation of the predictors within each slice, SIR identifies the directions that explain the most variation in these expectations as estimates of the matrix P. However, SIR relies heavily on the assumption of Gaussian-distributed data, and while it can be extended to elliptically-contoured distribution (where probability density contours are ellipsoids, which is also the case of Gaussian data), this remains a strong limitation that restricts its applicability. Furthermore, its consistency is guaranteed only when the ratio of the dimension to the sample size tends to zero, which can be a significant limitation in high-dimensional settings [Qian Lin and Liu, 2019].

Principal Hessian directions (PHD), introduced by Li [1992], is another significant moment-based approach that extends beyond single-index models. Unlike SIR, PHD utilises higher-order moments, such as $\mathbb{E}(YXX^\top)$, to uncover the linear features. By analysing the Hessian matrix of the predictors relative to the response, PHD can capture more complex, non-linear relationships. However, this method is computationally demanding due to the need for second derivative calculations and assumes that the covariates follow an elliptically contoured distribution, which is also restrictive.

There have been interesting extensions to these methods, for instance, Babichev and Bach [2018] extended SIR by integrating score functions, thereby enabling its use for more general distributions. However, this approach still requires prior knowledge or estimation of the score function, which introduces additional complexity. The problem of learning the score function is a non-parametric one and is subject to the curse of dimensionality. To address this, Babichev and Bach [2018] proposed a method that learns both the score function and the feature space simultaneously using trace norm regularisation, though the method lacks theoretical guarantees on its performance. Further advancements include Qian Lin and Liu [2019], which combined the SIR approach with LASSO-type regularisation to address the consistency issue in high-dimensional settings but still only considers elliptically-contoured distributions.

These methods primarily aim to identify the hidden linear features, while the link function can then be estimated in a subsequent step using conventional non-parametric techniques on the data projected onto the estimated feature space. While these approaches offer valuable insights and tools for handling multi-index models, their reliance on strong assumptions about the underlying data-generating process can significantly limit their practical applicability. Additionally, they focus exclusively on additive noise models, which effectively restricts their scope to scenarios where the square loss is used in the expected risk minimisation framework discussed earlier. These limitations underscore the need for more flexible and robust methods that can adapt to a broader range of data distributions and loss functions.

2.2 Optimisation-Based Methods

Optimisation-based methods represent a significant line of research in estimating multi-index models. Unlike moment-based methods, which focus on extracting linear features through statistical moments, optimisation-based approaches directly seek to optimise an objective function that captures the structure of the multi-index model.

One of the most significant contributions in this area is the minimum average variance estimation (MAVE) method, introduced by Xia et al. [2002]. MAVE tackles the estimation of the feature space in multi-index models by recasting it as an optimisation problem, which, while intractable in its original form, is estimated through an efficient approximation. A notable feature of MAVE is its cross-validation technique for determining the dimension of the feature space, which has been proven to converge in probability. Although the conditions imposed by MAVE on the data-generating mechanism are less restrictive than those required by moments-based methods, they are more technically intricate. However, like many other methods, MAVE is subject to the curse of dimensionality, with the rate of estimation deteriorating exponentially as the dimensionality of the data increases. Importantly, MAVE does not focus on estimating the link function and implicitly assumes a squared loss framework, although it is versatile enough to handle time series data as well.

The structural adaptation via maximum minimisation (SAMM) method, introduced by Dalalyan et al. [2008], is another optimisation-based approach designed to learn the feature space in multi-index models with mild technical assumptions on the data distribution. Unlike traditional methods that typically sum discrepancies across data points, SAMM focuses on minimising the maximum discrepancy between observed data and model predictions. Additionally, SAMM exploits gradient information to enhance the accuracy of the feature space estimation, a concept similarly employed by Xia et al. [2002] in the context of the OPG method, which is distinct from MAVE. One of SAMM's key strengths is its ability to achieve $\sqrt{n}$-consistency up to a logarithmic factor when the structural dimension is small ($s \leq 4$).

The SEAS (subspace estimation with automatic dimension and variable selection) method by Jing Zeng and Zhang [2024] effectively bypasses the curse of dimensionality by simultaneously accounting for both a small number of hidden linear features and a limited set of relevant variables. This is achieved through the application of nuclear norm and group sparsity penalties. However, the method assumes the linearity condition, which holds, for instance, when the data distribution is elliptically contoured.

The application of reproducing kernel Hilbert spaces (RKHS) to multi-index models has been explored by Fukumizu et al. [2009] and Fukumizu et al. [2004]. These methods approach dimensionality reduction by identifying a low-dimensional subspace of the input space that retains the statistical relationship between the input X and the output Y without requiring assumptions about the marginal distribution of X or a parametric model for the conditional distribution of Y. More recently, neural networks have also been applied to this problem, as demonstrated by Mousavi-Hosseini et al. [2024] and Bietti et al. [2022] (for the single-index model), who both focused on the continuous limit of the optimisation process. Furthermore, Bach [2024, Chapter 9] has shown that one-hidden-layer neural networks of infinite width with ReLU activation are adaptive to hidden linear features. Unlike earlier methods, these neural network approaches simultaneously learn both the feature space and the prediction function, a key difference from the other presented methods.

2.3 Limitations of Existing Methods and Goals

As we have seen, the estimation of multi-index models has been approached through a variety of methods, each designed to address specific assumptions and practical considerations. However, several limitations persist across these approaches. A significant number of these methods depend on strong assumptions about the data-generating process, such as the requirement for covariates to follow an elliptically contoured distribution or the need for the distribution of covariates to be known or pre-estimated. Moreover, many existing methods are susceptible to the curse of dimensionality, often exhibiting an exponential dependency on the data dimension, which hampers their applicability in high-dimensional settings. Another notable limitation is the lack of joint estimation of the link function and the underlying feature space, as most methods focus solely on recovering the features. Furthermore, these approaches are typically confined to additive noise models, which effectively limits them to scenarios that align with the use of the square loss.

In this thesis, we therefore aim to develop methods that operate under minimal assumptions about the data-generating mechanism and avoid overly restrictive assumptions on the true regression function, which minimises the expected risk for a given loss. We attempt to move beyond the square loss and to simultaneously estimate the feature space, its dimension, and the prediction function. Additionally, we strive to design approaches that mitigate the curse of dimensionality, enhancing their applicability to high-dimensional datasets. These objectives will be pursued within the framework of regularised empirical risk minimisation, providing a flexible approach for supervised learning with hidden linear features.

3 Overview of Contributions

To address the challenges highlighted in the previous subsections, we propose three distinct methods for non-parametric learning with hidden linear features using regularised empirical risk minimisation. Before delving into the contributions of each chapter, we provide a brief overview of the thesis structure. Each chapter introduces a different method, complete with its own notations (with the majority of them being common) and results, allowing them to be read independently. The chapters are presented in the order of their development during the course of this thesis.

In Chapter 2, we explore a novel approach as an extension of the work of Rosasco et al. [2013] on variable selection. The method incorporates a trace norm penalty on the sample matrix of gradients within reproducing kernel Hilbert spaces (RKHS) that include the partial derivatives of their kernel functions. The key idea is to leverage the gradients of the function as a means to capture the underlying linear structure within the data.

In Chapter 3, we extend the empirical risk minimisation framework by introducing a derivative-based overlapping group LASSO penalty, applied to functions represented in a basis of multivariate orthonormal Hermite polynomials. By using the orthogonality and rotation invariance properties of Hermite polynomials, we iteratively rotate the data, aligning it with the most informative directions.

In Chapter 4, we introduce a novel method by using averages of Sobolev spaces over one-dimensional projections of the data. The method combines kernel methods with infinite-width one-hidden layer neural networks. Our approach, centred around the Brownian kernel, substitutes the non-linearity of ReLU activations in neural networks with a kernel-based method. The positive homogeneity of the Brownian kernel is pivotal in steering the optimisation process.

Finally, the conclusion will address several key unresolved research questions. We now proceed to a more detailed examination of each contribution, starting with a discussion of the methods, followed by the main results, and concluding with an analysis of the strengths and weaknesses of each approach. Relevant references for the complete contributions will be provided in context.

3.1 Trace Norm Penalty on Sample Matrix of Gradients

Here, we present the yet unpublished work of Chapter 2, while the corresponding code is available at `https://github.com/BertilleFollain/KTNGrad`.

Method. In Chapter 2, building on the work of Rosasco et al. [2013] on variable selection, we introduce a novel method called KTNGrad. The key idea behind KTNGrad is to exploit the information about the underlying feature space that is contained in the gradients while operating within reproducing kernel Hilbert spaces that are sufficiently regular. This allows us to avoid computing gradients through finite differences and instead compute them directly through the intrinsic properties of RKHS.

We begin by noting that if the minimiser of the expected risk (here considered for the square loss), f^*, adheres to the multi-index model, then there exist a function g^* and a matrix $P \in \mathbb{R}^{d\times s}$ such that $f^* = g^*(P^\top \cdot)$. Assuming all relevant quantities are well-defined, this implies that for any $x \in \mathcal{X}$, the gradient satisfies $\nabla f^*(x) = P\nabla g^*(P^\top x)$. Consequently, the gradient at any given point contains information about the underlying feature space. More formally, for any function f belonging to the Sobolev space $H^1(\rho_X)$ (with ρ_X the distribution of the covariates), defined as $H^1(\rho_X) := \{f \in L^2(\rho_X) \mid \forall a \in [d],\ \partial f(x)/\partial x^{(a)} \in L^2(\rho_X)\}$, we can express the covariance matrix of the gradients of f as

$$\operatorname{cov}(\nabla f) := \operatorname{cov}(\nabla f(X)) = \mathbb{E}_{\rho_X}\left(\nabla f(X)\nabla f(X)^T\right) \in \mathbb{R}^{d\times d}.$$

Within this framework, the covariance matrix of the gradients of the true function f^* satisfies $\operatorname{cov}(\nabla f^*) = P \operatorname{cov}(\nabla g^*(P^T X))P^T$. Assuming that the rank of $\operatorname{cov}(\nabla g^*(P^T X))$ is equal to s, the number of linear features, it follows that the rank of $\operatorname{cov}(\nabla f^*)$ is also s, which is typically much smaller than d.

However, because the rank is both non-continuous and non-convex, it presents significant challenges as an optimisation penalty. Additionally, the direct computation of the covariance matrix is not feasible since ρ_X is unknown. To overcome these issues and following classical extensions of ℓ_1 norm regularisation, we employ a convex relaxation by using the trace norm ($\|\cdot\|_*$) of the sample matrix of gradients

$$\nabla_n f := (\nabla f(x_1)^T, \nabla f(x_2)^T, \dots, \nabla f(x_n)^T)^T/\sqrt{n} \in \mathbb{R}^{n\times d},$$

which estimates $\operatorname{tr}\left(\sqrt{\operatorname{cov}(\nabla f)}\right)$, providing a convex alternative to the rank of $\operatorname{cov}(\nabla f)$.

There are still two challenges to address: how to compute the gradients at the data points, given that finite differences are often unreliable and unstable in the context of random covariates, and how to effectively compute the minimiser of the regularised empirical risk minimisation problem. This is where reproducing kernel Hilbert spaces (RKHS) become advantageous. Let $\mathcal{H}$ denote an RKHS associated with a reproducing kernel k. If we assume that the kernel is twice differentiable, as is the case with the Gaussian kernel, then for all $a \in [d]$, $(\partial_a k)_x := t \to \partial k(x,t)/\partial x_a$ (the derivative w.r.t the a-th component of x) also belongs to $\mathcal{H}$ for any $x \in \mathcal{X}$. Moreover, for any $f \in \mathcal{H}$, the partial derivative of any $f \in \mathcal{H}$ at x with respect to x_a can be computed as $\frac{\partial f(x)}{\partial x_a} = \langle f, (\partial_a k)_x\rangle_{\mathcal{H}}$. This

property allows us to compute the gradients at the data points directly using the RKHS structure, bypassing the need for finite differences. Additionally, empirical risk minimisation in an RKHS is typically tractable due to the representer theorem, which ensures that the minimiser can be expressed as a linear combination of the kernel functions at the data points.

Formally, the KTNGrad estimator $\hat{f}$ is then defined by solving the following optimisation problem

$$\hat{f}_\tau = \arg\min_{f \in \mathcal{H}} \widehat{\mathcal{R}}(f) + 2\tau \|\nabla_n f\|_* + \tau\nu \|f\|_{\mathcal{H}}^2,$$

where the loss defining the risk is the square loss and $\tau > 0$ is a regularisation parameter that needs to be chosen. The added regularisation using the RKHS norm is used for computational and statistical stability with a fixed and very small parameter ν.

To compute the estimator, we employ an adapted version of the representer theorem, which allows us to express the estimator as a linear combination of the functions in the sets $\{k(x_i, \cdot) \mid i \in [n]\}$ and $\{\partial_a k_{x_i} \mid i \in [n], a \in [d]\}$. To handle the trace norm penalty, we reformulate the problem using a variational approach, resulting in a convex minimisation problem involving two sets of variables. The optimisation process alternates between fixing one variable and solving for the other in closed form. This alternating minimisation is proved to converge, with each iteration incurring a computational cost of $O(n^3 d^4)$.

The features can then be computed by taking the leading right singular vectors of the sample matrix of gradients $\nabla_n \hat{f}_\tau$, with their number $\hat{s}$ estimated by the rank of $\nabla_n \hat{f}_\tau$ leading to $\hat{P}$, or if s is known to $\hat{P}_s$.

Main result. The main statistical properties of KTNGrad are summarised in the following informal theorem, which outlines the key assumptions as well as the prediction and feature learning capabilities.

Theorem 1 (Informal)**.** *We assume that the true regression function f^* belongs to $\mathcal{H}$, with a twice differentiable reproducing kernel.*

- ***Convergence of the expected risk:*** *The expected risk of KTNGrad converges to the minimal risk $\mathcal{R}(f^*)$ without exponential dependency in the data dimension d. Specifically, there exists a universal constant $C > 0$ such that for any $\delta \in (0, 1]$, with probability at least $1 - \delta$,*

$$\mathcal{R}(\hat{f}_\tau) - \mathcal{R}(f^*) \leq C \left(\frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{\tau\nu}} + 1 \right)^2 + \sqrt{\frac{\tau}{\nu}} \frac{d^{5/4}}{n^{1/4}} \right) \log \frac{6 + 2d}{\delta} + \tau \left(2\|\nabla f^*\|_* + \nu \|f^*\|_{\mathcal{H}}^2 \right).$$

- ***Recovery of the hidden linear features:*** *The method is capable of recovering the underlying feature space in Frobenius norm when the dimension is known and otherwise in a safe filter manner, when the sample size increases. For any positive sequence $(\tau_n)_{n \in \mathbb{N}}$ such that $\tau_n \to 0$ and $(\sqrt{n}\tau_n^2)^{-1} \to 0$ as $n \to \infty$, with Π_Q the projection matrix associated to a matrix Q, we have*

$$\|\Pi_P - \Pi_{\hat{P}_s}\|_F^2 \xrightarrow{P} 0 \quad \text{and} \quad \|\Pi_P (I_d - \Pi_{\hat{P}})\|_F^2 \xrightarrow{P} 0.$$

Analysis. We summarise the key contributions and insights gained from Chapter 2.

- **Rate of convergence:** KTNGrad achieves convergence rates for the expected risk that do not depend exponentially on the data dimension, addressing a major

challenge in high-dimensional non-parametric learning. However, this result is based on the strong assumption that the model is well-specified ($f^* \in \mathcal{H}$), and there is still a polynomial dependency on the data dimension.

- **Feature recovery:** KTNGrad demonstrates strong capability in recovering the underlying feature space, as supported by both statistical analysis and experimental results. However, while the method consistently identifies the correct feature space, it does not recover its dimension. This is visible in both experiments and theory, as we only manage to prove that the estimate of the dimension is asymptotically larger than the true dimension of the feature space due to the lower semi-continuity of the rank, resulting in a safe filter. However, using an adaptive method might alleviate this issue.

- **Computational complexity:** The method requires solving a convex optimisation problem with a trace norm penalty on the sample gradient matrix. While this ensures good convergence properties, this comes with a substantial computational cost of $O(n^3d^4)$. Although employing Nyström approximation could help reduce some of the computational burden, the reliance on the derivatives of all variables and the context of working in a reproducing kernel Hilbert space make the method inherently resource-intensive.

- **Inadequate function space:** A significant conceptual limitation of KTNGrad is that RKHS associated with usual kernels are not well-suited to the multi-index model. The core issue lies in the incompatibility between the assumptions that $f^* \in \mathcal{H}$ and $f^* = g^*(P^\top \cdot)$, a critique which also applies to the variable selection framework discussed by Rosasco et al. [2013]. For instance, belonging to the RKHS corresponding to the Gaussian kernel requires that all first-order derivatives of f^* be square-integrable with respect to the Lebesgue measure on $\mathbb{R}^d$. However, in the simple case of one relevant variable $f^*(x) = g^*(x_1)$, this condition is $\int_{\mathbb{R}^d} \left((g^*)'(x_1)\right)^2 \mathrm{d}x_1 \dots \mathrm{d}x_d < \infty$, which is not possible except in edge cases. This reasoning leads us to explore a Hilbert space of functions with an orthonormal Hermite polynomial basis in the next chapter. Decomposing functions in this basis reveals that the function space aligns well with the multi-index model and variable selection framework, offering a clear interpretation of dependency on a few variables or linear projections through the coefficients in the basis.

3.2 Group LASSO Penalty on Hermite Polynomials Decomposition

This contribution corresponds to the contents of Chapter 3, which has been accepted by the Electronic Journal of Statistics: Follain and Bach [2024b], while the code is available at `https://github.com/BertilleFollain/RegFeaL`.

Method. The proposed method, RegFeaL, leverages the orthogonality and rotational invariance of normalised Hermite polynomials to perform either variable selection or feature learning. First, we highlight the relevant properties of the Hermite polynomials. The normalised one-dimensional Hermite polynomials $(h_k(x))_{k\geq 0}$ form an orthonormal basis for the standard Gaussian measure on $\mathbb{R}$. The first few polynomials are given by $h_0(x) = 1, \quad h_1(x) = x, \quad h_2(x) = \frac{1}{\sqrt{2}}(x^2 - 1), \quad h_3(x) = \frac{1}{\sqrt{6}}(x^3 - 3x)$. These polyno-

mials are extended to the multivariate case by defining, for $\alpha \in \mathbb{N}^d$,

$$H_\alpha(x) = \prod_{a=1}^{d} h_{\alpha_a}(x_a).$$

This family forms an orthonormal basis for the Hilbert space of squared integrable function with the distribution q, $L^2(q)$, where $q(x) = e^{-\|x\|^2/2}/(2\pi)^{d/2}$ denotes the standard normal distribution on $\mathbb{R}^d$.

In this context, if a function $f \in L^2(q)$ is expressed as $f = \sum_{\alpha\in\mathbb{N}^d} \hat{f}(\alpha)H_\alpha$, then the function f does not depend on a variable x_a if and only if all coefficients $\hat{f}(\alpha)$ for $\alpha \in \mathbb{N}^d$ such that $\alpha_a > 0$ are zero.[1] This specific sparsity pattern in the coefficients motivates the use of an overlapping group LASSO type of penalty. Consequently, the Hermite polynomial basis is well-suited for variable selection. To achieve this, we introduce a sparsity-inducing penalty, depending on hyper-parameters $r \in (0, +\infty)$ and $(c_k)_{k\in\mathbb{N}^*}$ (which is either $c_k = \mathbb{1}_{k\leq M}$ or $c_k = \rho^k$ with $M \in \mathbb{N}^*$ and $\rho \in (0, 1)$)

$$\Omega_{\text{var}}(f) = \left(\sum_{a=1}^{d} \left(\sum_{\alpha\in(\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} \right)^{1/r},$$

where $|\alpha| = \sum_{a=1}^{d} \alpha_a$. This penalty encourages sparsity in the dependency of f on individual variables. The condition

$$\left(\sum_{\alpha\in(\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} = 0 \iff \int_{\mathbb{R}^d} \left(\frac{\partial f}{\partial x_a} \right)^2 q = 0$$

highlights that the penalty enforces the nullity of the derivative of f with respect to x_a. We estimate f^* in the variable selection setting by solving the following optimisation problem

$$f^{\lambda,\mu}_{\text{var}} := \operatorname*{arg\,min}_{f\in\mathcal{F}} \ \widehat{\mathcal{R}}(f) + \lambda\Omega_0^2(f) + \mu\Omega^r_{\text{var}}(f),$$

where λ is a fixed parameter for the smoothness-inducing penalty Ω_0, μ is a hyperparameter to be chosen, and the loss defining the risks is any convex loss. When $r \geq 1$ and the loss function is convex, the objective function is strongly convex, ensuring a unique global minimiser. For $r < 1$, which is typically used in practice to bypass the issues arising with LASSO-type method enforcing too much bias, only a local minimiser can be found.

The rotational invariance property of the Hermite polynomials is central to extending this method to feature learning. Specifically, for any $x, x' \in \mathbb{R}^d$, any $k \in \mathbb{N}$, and any orthogonal $d \times d$ matrix R,

$$\sum_{|\alpha|=k} H_\alpha(x)H_\alpha(x') = \sum_{|\alpha|=k} H_\alpha(Rx)H_\alpha(Rx').$$

This property allows for the development of a penalty suited for feature learning, defined as

$$\Omega_{\text{feat}}(f) = \left(\text{tr}\left(M_f^{r/2}\right)\right)^{1/r},$$

[1] Note that here $\hat{f}(\alpha)$ corresponds to coefficients in the Hermite polynomials decomposition of f, not to the estimator of f^*.

where the matrix M_f is given by

$$(M_f)_{a,b} = \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \sqrt{\alpha_a + 1}\sqrt{\alpha_b + 1}\hat{f}(\alpha + e_a)\hat{f}(\alpha + e_b), \quad a, b \in [d].$$

Again, there is a link between the nullity of the derivatives and the definition of the penalty, which is described in the main text. The matrix M_f is positive semi-definite, and the penalty $\Omega_{\text{feat}}(f)$ encourages sparsity by pushing the eigenvalues of M_f towards zero, thus promoting a low-rank solution. Importantly, $c_{|\alpha|}$ depends solely on $|\alpha|$, ensuring the penalty remains rotation invariant, which is crucial to prevent the penalty from favouring specific directions.

The eigendecomposition $M_f = UDU^\top$ reveals that if the rank of D is s, then the function f depends only on s linear combinations of the original variables, corresponding to the directions in U with non-zero eigenvalues. Moreover, one can construct a rotated function $g = f(U\cdot)$ such that the feature penalty on f is equivalent to the variable selection penalty on g. This shows that feature learning can be seen as an extension of variable selection that allows for rotations in the feature space.

The estimator for f^* in the feature learning setting is thus defined similarly to the variable selection setting by switching Ω_{var} for Ω_{feat}. To compute the estimator, we employ a variational formulation for $\Omega_{\text{feat}}(f)$ that reformulates the problem as the minimisation over two variables: the function f and an auxiliary variable Λ. Specifically, we solve

$$f^{\lambda,\mu}_{\text{feat}},\ \Lambda^{\lambda,\mu}_{\text{feat}} = \operatorname*{arg\,min}_{f \in \mathcal{F},\ \Lambda \in \mathbb{R}^{d \times d}} \widehat{\mathcal{R}}(f) + \lambda \Omega_0^2(f) + \mu \text{trace}(\Lambda^{-1} M_f),$$

subject to the constraints that $\Lambda = R\text{Diag}(\eta)R^\top$ with R an orthogonal $d \times d$ matrix and $\sum_{a=1}^d \eta_a^{r/(2-r)} = 1$, with η a positive vector. The optimisation in closed-form (for the square loss, otherwise we need to use other methods such as gradient descent to compute the function f) alternates between fixing f and updating Λ by computing the eigendecomposition of M_f and fixing Λ and updating f by solving a kernel ridge regression problem, with the reproducing kernel k_Λ defined by Hermite polynomials

$$k_\Lambda(x, x') = \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|} H_\alpha(R^\top x) H_\alpha(R^\top x')}{\lambda + \mu \alpha^\top \eta^{-1}}.$$

This iterative procedure can be viewed as progressively rotating the data to uncover and align with the underlying features while simultaneously learning the prediction function.

As the kernel function is defined as an infinite sum over Hermite polynomials, direct computation is infeasible, and it is approximated via sampling. We use a tailored importance sampling technique where we sample from the distribution of Hermite coefficients α guided by the auxiliary variable η. The whole optimisation process converges rapidly in practice, typically within a small number of iterations. The complexity of one iteration is

$$O\Big(\underbrace{nm'd + nd^2}_{\text{Hermite features}} + \underbrace{d^2(m')^2 + d^3}_{M_f \text{ and its eigendecomposition}} + \underbrace{md}_{\text{Sampling}} + \underbrace{nm' \max(n, m')}_{\text{Kernel Ridge}}\Big),$$

where m is the number of samples drawn for α (and m' the resulting unique samples). This complexity can be substantial, as m' must be sufficiently large to ensure that the kernel's representation is accurate.

Main result. The primary statistical finding of this chapter is that the RegFeaL estimator, under minimal assumptions, achieves convergence of the expected risk to the minimal risk with high probability despite being sensitive to the dimensionality of the data. We illustrate this with an informal theorem for the case where the covariates are bounded and the regularising sequence is chosen as $c_k = \rho^k$.

Theorem 2 (Informal)**.** *Assume that the covariates X are bounded, i.e., $\|X\|_2 \leq R$ almost surely, and that the loss function ℓ is Lipschitz with constant G. Let the true regression function f^* exist and belong to $L^2(q)$. The regularisation parameter λ is set to zero and μ is chosen based on known problem parameters, and we define the norm $\Omega(f)$ as $\Omega_{\text{feat}}(f) + |\hat{f}(0)|$ or $\Omega_{\text{var}}(f) + |\hat{f}(0)|$. Then for any $\delta \in (0,1)$, with probability greater than $1-\delta$, the expected risk $\mathcal{R}(f^\mu)$ of the RegFeaL estimator f^μ satisfies*

$$\mathcal{R}(f^\mu) \leq \mathcal{R}(f^*) + \Omega(f^*) \cdot \frac{G}{\sqrt{n}} \sqrt{1 + \frac{e^{R^2/2}}{(1-\rho)^d}} \left(16\sqrt{\frac{\pi}{2}} + 4\sqrt{2}\sqrt{\log \frac{2}{\delta}} \right),$$

and the norm of the estimator is bounded by $\Omega(f^\mu) \leq 2\Omega(f^*)$.

Chapter 3 presents this result in a more general setting, where the data is not necessarily bounded and where any choice of the hyper-parameter sequence $(c_k)_{k>0}$ can be considered. This informal result underscores the method's sensitivity to the dimensionality d, particularly due to the exponential dependency introduced by the infinite Hermite polynomial basis, while still maintaining a favourable dependency on the sample size. The assumption on the true function f^* is mild, given the broadness of the considered function space.

Analysis. We summarise the key contributions and insights gained from Chapter 3.

- **Use of Hermite polynomials:** The method leverages the orthogonality and rotational invariance of Hermite polynomials to effectively align the data with its leading directions. The structure of Hermite polynomials is particularly well-suited for variable selection and feature learning, as it allows for the definition of an overlapping group LASSO-type penalty that characterises dependencies on a few variables or linear features. Compared to the previous function space of Chapter 2, the Hermite decomposition is advantageous because it naturally accommodates functions that depend on a small subset of variables or linear features.

- **Statistical results:** The method offers statistical guarantees under minimal assumptions, particularly with respect to the function space in which the true prediction function resides. This broad applicability is a key strength. However, the reliance on an infinite basis introduces an exponential dependency on the dimensionality of the data, which poses significant challenges in high-dimensional settings where multi-index models are most relevant. While alternative proof techniques might reduce this dependency, such an approach is not yet apparent.

- **Computational cost:** The method's foundation on an infinite Hermite polynomial basis necessitates a sophisticated sampling scheme to approximate the kernel at each iteration of the optimisation process. This approach, while theoretically sound, is computationally intensive and hinders the method's practical applicability, especially in large-scale problems.

- **Function space limitations:** The infinite Hermite polynomial basis and corresponding Hilbert space of function are well-suited for capturing the sparsity patterns crucial for variable selection and feature learning. Compared to the RKHS considered in the previous chapter, the Hilbert space is actually compatible with the multi-index model and the variable selection setting, and it is also broader, which allows for milder assumptions on the true regression function. However, this very expansiveness also introduces significant challenges. The reliance on an infinite basis leads to an exponential dependency on the dimensionality in the statistical results and necessitates a complex sampling scheme in the optimisation procedure. This suggests that alternative function spaces might be more appropriate. In the next chapter, we consider another function space based on the fusion of infinite-width one hidden-layer neural networks and kernel methods. Although this space is also computationally intractable due to its integral-based definition, it can be more efficiently approximated using particles, offering a simpler alternative to the sampling scheme employed for the Hilbert space used in RegFeaL.

3.3 Integrating Neural Networks and Kernel Methods

This contribution corresponds to the contents of Chapter 4, which are available in the preprint (under review by the Journal of Machine Learning Research): Follain and Bach [2024a], while the code is available at `https://github.com/BertilleFollain/BKerNN`.

Method. In this chapter, we introduce a novel approach called Brownian kernel neural network (BKerNN), which merges neural networks and kernel methods. The key idea behind BKerNN is the construction of a custom function space inspired by the infinite-width limit of single hidden layer neural networks, where the non-linearity is replaced by a function from a RKHS. The function space, denoted by $\mathcal{F}_\infty$, allows each function to be represented as an integral over linear combinations of input features, weighted by a probability measure. Specifically, functions in this space take the form

$$f(x) = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top x)\,\mathrm{d}\mu(w),$$

where c is a constant, w is a direction vector lying on the unit sphere $\mathcal{S}^{d-1}$ for a non-specified norm $\|\cdot\|$ (either ℓ_1 or ℓ_2), g_w is a function that varies with w and belongs to a Sobolev space which is also a RKHS $\mathcal{H}$, and μ is a probability measure over the sphere. The space $\mathcal{H}$ contains functions with square-integrable weak derivatives, ensuring a certain degree of smoothness, and such that $g(0) = 0$. Its inner product is defined as $\langle g, \tilde{g}\rangle = \int_{\mathbb{R}} g'\tilde{g}'$. The RKHS $\mathcal{H}$ is associated to the reproducing kernel $k^{(B)}(a,b) = (|a| + |b| - |a-b|)/2 = \min(|a|,|b|)\mathbb{1}_{ab>0}$, which is the Brownian motion kernel.

In practice, we approximate this infinite-dimensional space with a finite-width version $\mathcal{F}_m$, where the integral is replaced by a finite sum over m particles (analogous to neurons in a neural network). Thus, functions in $\mathcal{F}_m$ are expressed as

$$f(x) = c + \frac{1}{m}\sum_{j=1}^{m} g_j(w_j^\top x),$$

where $(w_j)_{j\in[m]}$ are direction vectors, and $(g_j)_{j\in[m]}$ are corresponding functions from the space $\mathcal{H}$. The learning process in BKerNN is guided by a regularisation term that controls

the complexity of the learned function. The basic regularisation is defined as

$$\Omega_0(f) = \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}} \, \mathrm{d}\mu(w),$$

where $\|g_w\|_{\mathcal{H}}$ measures the roughness of the function g_w. This regularisation induces sparsity in the learned representations by limiting the number of non-zero functions g_w, which indirectly promotes feature selection. To further enhance the feature learning capability of BKerNN, other regularisation terms can be introduced. For instance, a variable penalty encourages the model to depend on only a few variables by penalising the row norms of the weight matrix containing the $(w_j)_{j\in[m]}$. A feature penalty, on the other hand, promotes learning a low-rank representation by applying a nuclear norm penalty to the weight matrix, encouraging dependency on only a few linear transformations of the data. These penalties can also be made concave, which, although more challenging to optimise in theory, can lead to even sparser solutions by promoting boundary solutions.

The optimisation objective is to minimise the regularised empirical risk

$$\hat{f}_\lambda = \operatorname*{arg\,min}_{f\in\mathcal{F}} \ \widehat{\mathcal{R}}(f) + \lambda\Omega_{\text{weights}}(f),$$

where λ is the regularisation parameter and Ω_{weights} is any of the considered penalties.

Interestingly, BKerNN can be viewed from two different perspectives: as a kernel method and as a neural network. From the kernel perspective, the learning process consists in kernel ridge regression with a kernel that is learnt during training. The kernel matrix is defined as

$$K = \frac{1}{m}\sum_{j=1}^{m} K^{(w_j)},$$

where $K^{(w_j)}$ is the kernel matrix associated with the Brownian kernel for the training data projected onto the direction w_j.

From the neural network perspective, BKerNN resembles a one-hidden-layer neural network where the weights from the input layer to the hidden layer are the direction vectors $(w_j)_{j\in[m]}$, and the activation functions are the learned functions $(g_j)_{j\in[m]}$. Unlike traditional neural networks, where activation functions are predefined and only a multiplicative factor is learnt, BKerNN learns the activation functions directly, which adds flexibility to the model.

The function space $\mathcal{F}_\infty$ is broader than the space of functions that can be represented by traditional infinite-width one-hidden layer neural networks with ReLU activations. This can be observed through Fourier transform analysis, indicating that BKerNN is capable of capturing a wider variety of functions. Remarkably, this expanded representational power does not result in increased optimisation complexity.

The computation of BKerNN is based on using an adapted version of the representer theorem, which yields a parametric formulation for minimisation. We focus on the square loss here for ease of exposition and the availability of closed-form solutions. However, the method is generalisable to other loss functions using gradient-based techniques. The optimisation problem can be formulated as follows

$$\min_{w_1,\dots,w_m\in\mathbb{R}^d, c\in\mathbb{R}, \alpha\in\mathbb{R}^n} \frac{1}{2n}\|Y - K\alpha - c\mathbb{1}_n\|_2^2 + \frac{\lambda}{2}\alpha^\top K\alpha + \lambda\Omega_{\text{weights}}(w_1,\dots,w_m),$$

where K is the kernel matrix defined by the weights $(w_j)_{j\in[m]}$ (which are no longer constrained anymore after some reformulation) and α appears through the representer theo-

rem. The optimisation process then alternates between two steps: optimising the coefficients α and the intercept c while keeping the weights $(w_j)_{j\in[m]}$ fixed, and then optimising the weights $(w_j)_{j\in[m]}$ using a proximal gradient descent approach.

When the weights $(w_j)_{j\in[m]}$ are fixed, the kernel matrix K becomes constant, allowing the optimisation of α and c to be solved explicitly, akin to solving a classical kernel ridge regression problem. The complexity of this step is $O(n^3 + n^2 d)$, which can be computationally demanding for large datasets. To reduce the computational cost, techniques like the Nyström method can be employed to approximate the kernel matrix.

The next step involves optimising the weights $(w_j)_{j\in[m]}$ while keeping α and c fixed. This is more challenging because the resulting objective function G is not convex with respect to the weights, and it is only differentiable almost everywhere. The weights are therefore updated using a proximal gradient descent approach. The proximal operator depends on the penalty. For instance, the update for the basic penalty Ω_0 is given by

$$w_j \leftarrow \mathrm{prox}_{\lambda\gamma\Omega}\left(w_j - \gamma\frac{\partial G}{\partial w_j}\right) \quad \text{where } \mathrm{prox}_{\lambda\gamma\Omega}(u) = \left(1 - \frac{\lambda\gamma}{2m}\frac{1}{\|u\|}\right)_+ u,$$

where γ is the step-size, adjusted through a backtracking line search to ensure efficient optimisation. Each proximal step is easy to compute using the explicit formulas, with complexities ranging from $O(md)$ for the basic and variable penalties and $O(md\min(m,d))$ for the feature penalties.

The optimisation procedure leverages the homogeneity of the Brownian kernel, which ensures well-behaved optimisation dynamics. The method's convergence is supported by theoretical insights that align with established results in mean-field neural networks. Despite the lack of a formal proof due to the non-differentiability of the Brownian kernel, the procedure is robust in practice, with experiments confirming its effectiveness.

Main result. The following informal theorem provides insight into the generalisation capabilities of BKerNN by offering a high-probability bound on the expected risk of the estimator.

Theorem 3 (Informal)**.** *Consider the BKerNN estimator $\hat{f}_\lambda$ with the basic penalty Ω_0. Assume that the loss is convex and Lipschitz with constant L, that the true regression function f^* belongs to $\mathcal{F}_\infty$ and that $1 + \sqrt{\|X\|^*}$ is subgaussian with variance proxy σ^2, with $\|\cdot\|^*$ the dual norm of the one used to define the sphere $\mathcal{S}^{d-1}$. Then, with λ chosen using known problem parameters (independent of $\Omega_0(f^*)$), with probability at least $1-\delta$, the expected risk of $\hat{f}_\lambda$ is bounded by*

$$\mathcal{R}(\hat{f}_\lambda) \le \mathcal{R}(f^*) + \Omega_0(f^*)CL\left(\frac{1}{\sqrt{n}} + G_n + \frac{\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}\right),$$

where C is a universal constant, and G_n denotes the Gaussian complexity of the function class with Ω_0 constrained below a certain threshold. The quantity G_n is further bounded using another universal constant, C', as follows

$$G_n \le C' \min\left(\sqrt{\frac{d}{n}}\sqrt{\log(n)}\sqrt{\mathbb{E}_X\|X\|^*},\ \frac{1}{n^{1/6}}(\log d)^{1/4}\left(\mathbb{E}_{X_1\dots X_n}\big(\max_{i\in[n]}\|X_i\|^*\big)^2\right)^{1/4}\right).$$

The bound on G_n is presented in two forms: a dimension-dependent bound and a dimension-independent bound. The dimension-dependent bound scales well with the sam-

ple size and shows only a square-root dependency on the dimension. In contrast, the dimension-independent bound, while scaling less favourably with the sample size at $n^{-1/6}$, depends on the dimension only logarithmically. The dependency of the distribution-dependent terms in the data dimension is also analysed and is reasonable. This favourable outcome is achieved without imposing strong assumptions on the true regression function as the well-specified model assumption is reasonable given the broad function space $\mathcal{F}_\infty$. We conjecture that the true bound might actually be logarithmic in dimension while maintaining the standard $n^{-1/2}$ rate with respect to sample size. Additionally, the assumption of the data distribution's subgaussianity is mild, making this result widely general.

Analysis. We summarise the key contributions and insights gained from Chapter 4.

- **Combining strengths:** BKerNN efficiently merges some of the benefits of kernel methods and infinite-width one-hidden-layer neural networks. This is achieved by substituting the traditional non-linear activation function with a function drawn from a reproducing kernel Hilbert space, enhancing the model's expressive power.
- **Efficient optimisation:** The optimisation process is straightforward and robust compared to neural networks, benefiting from the positive homogeneity of the Brownian kernel. This property ensures that the optimisation aligns with insights from the mean-field analysis of neural networks, making the process theoretically sound. The approximation of the infinite-width space using particles is easy and principled compared to the sampling process used for RegFeaL
- **Generalisation guarantees:** The statistical analysis provides high-probability bounds on the expected risk, showing that BKerNN achieves competitive rates of convergence. Two types of bounds are provided: a dimension-dependent bound that scales well with sample size and a dimension-independent bound that scales less favourably with sample size but only logarithmically with the data dimension. The mild assumptions on data distribution and model specification make these results broadly applicable.
- **Practical performance:** Extensive numerical experiments validate the theoretical findings, with BKerNN outperforming traditional kernel methods and competing favourably with neural networks on real-world datasets.
- **Adaptivity in misspecified models:** If the model is not well-specified but the Bayes predictor f^* is Lipschitz continuous, neural networks with ReLU activation and bounded Banach norm achieve a convergence rate of $O(n^{-1/(d+5)})$, while kernel methods achieve $O(n^{-1/(d+1)})$, both constrained by the curse of dimensionality. In misspecified settings under the multi-index model where $f^* = g^*(P^\top \cdot)$, RKHS-based methods fail to exploit the reduced dimensionality, resulting in unchanged rates. However, neural networks can adapt to this structure, yielding rates that depend on the lower dimension of P rather than d. BKerNN shares this adaptivity, as indicated by the fact that $\Omega_0(f^*) \leq \Omega_0(g^*)$, which, along with its strong theoretical guarantees in well-specified models and excellent practical performance, underscores its significance in the field of non-parametric supervised learning with hidden linear features.

This concludes the presentation of the contributions. Each contribution is presented in detail in the following Chapters 2, 3, and 4. See the Conclusion for perspectives related to this work.

CHAPTER 2

Trace Norm Penalty on Sample Matrix of Gradients

The contents of this chapter have yet to be published while the code is available at `https://github.com/BertilleFollain/KTNGrad`.

Abstract

In this work, we tackle the challenges of high-dimensional nonparametric learning by focusing on multi-index models, i.e., where the regression function is represented as a composition of a low-dimensional linear projection and a nonlinear function. We are particularly interested in the empirical risk minimisation framework due to its versatility, as it can be applied to a wide range of problems beyond traditional square loss and regression tasks. We introduce a novel method that uses empirical risk minimisation within a reproducing kernel Hilbert space (RKHS), augmented by a trace norm penalty on the sample matrix of gradients. Our approach is computationally efficient, featuring a convex optimisation procedure that converges in just a few iterations and offers an explicit convergence rate via a reweighting technique. We establish the theoretical convergence of our method, demonstrating that it achieves convergence rates that do not depend exponentially on the data dimension for the expected risk of the function estimator in well-specified settings, while reliably recovering the underlying feature space in a safe filter manner. The effectiveness of our approach, named KTNGrad, is validated through a series of experiments that highlight its performance and behaviour.

Contents

1 Introduction

We focus on multi-index models [Xia, 2008], which provide an effective strategy for addressing the challenges posed by high-dimensional data in nonparametric supervised learning. In these models, the regression function f^*, which captures the relationship between the response and the covariates, is expressed as $f^* = g^*(P^\top \cdot)$, where g^* is an unknown nonparametric function, and P is a low-rank matrix that reduces the dimensionality of the data. This matrix P can be interpreted as a set of linear features that are relevant to the learning problem.

The multi-index model is particularly valuable because it mitigates the challenges associated with high-dimensional nonparametric models, where exponential dependency on the data dimension is often observed [Dalalyan et al., 2008], a manifestation of the curse of dimensionality [Bellman, 1961]. By leveraging the structure assumed in the multi-index model, prediction performance can be significantly enhanced. Additionally, learning the linear features represented by P offers intrinsic interpretability, making it an attractive approach for both predictive accuracy and understanding the underlying relationships in the data.

Related work. Various approaches have been developed to address multi-index models. One prominent avenue is the method of moments, which constructs specific moments that eliminate the unknown function, isolating the impact of the features. The foundational work by Brillinger [2012] initially focused on Gaussian data and feature spaces of dimension 1. This method was later broadened to handle distributions with differentiable log-densities, giving rise to the average derivative estimation (ADE) method [Stoker, 1986]. Further advancements allowed the method to accommodate higher-dimensional feature spaces through techniques such as slicing (e.g., slice inverse regression, SIR [Li, 1991]) and the use of second-order moments (e.g., principal Hessian directions, PHD [Li, 1992]) under assumptions like elliptical symmetry in the data distribution. Despite these developments, the efficacy of these methods often hinges on strong, and sometimes impractical, assumptions about the distribution's shape, which may also need to be known beforehand.

Beyond moment-based techniques, optimisation-driven approaches have also been explored. These methods, as seen in the works of Fukumizu et al. [2009] and Xia et al. [2002], employ local averaging to build an objective function, which is then minimised to estimate the feature subspace. Although these approaches can theoretically suffer from exponential dependence on the data dimension, they have demonstrated strong practical performance. Among them, the MAVE method [Xia et al., 2002] stands out as a leading tool in practical applications.

We are particularly interested in using regularised empirical risk minimisation due to its versatility, allowing it to be applied across various contexts beyond traditional square loss supervised learning. We consider this work as a first step toward expanding multi-index models to address more complex problems, such as those found in control systems. Our goal is to develop a method that makes minimal assumptions about the distribution and has limited dependency on the data dimension. Our approach involves regularising the empirical risk using derivatives, based on the observation that, under certain smoothness conditions, the gradient of f^* satisfies $\nabla f^* = P \nabla g^*(P^\top \cdot)$, thereby containing crucial information about the underlying feature space P. This regularisation by derivatives is common, such as in classical spline regularisation in Sobolev space [Wahba, 1990], while for linear subspace estimation, it has been used by Babichev and Bach [2018] in conjunction with the SADE method. We implement our method within reproducing kernel

Hilbert spaces, which allows us to avoid the potential instability associated with computing derivatives via finite differences. This use of RKHS has also been see in Cabannes et al. [2021] for semi-supervised learning or Rosasco et al. [2013] for variable selection. Our goal is precisely to extend the work of Rosasco et al. [2013] to the linear feature learning setting, which results in our framework being similar though their penalty is designed to select variables, whereas ours is focused on identifying linear subspaces.

Contributions. In this chapter, we introduce a novel approach for simultaneously estimating the regression function, the underlying feature space, and its dimension within the context of multi-index models. Our method leverages the regularised empirical risk minimisation framework with the square loss, working within a reproducing kernel Hilbert space (RKHS) that includes the partial derivatives of the reproducing kernel. Inspired by the relationship between gradients and the underlying features, we consider the trace norm of the sample matrix of gradients as a regularising penalty.

We establish the convergence of the optimisation procedure used to compute the estimators. In a well-specified setting, we demonstrate that the risk of the function estimator converges to the minimal risk at a rate that does not depend exponentially on the data dimension. Additionally, we show that our approach reliably recovers the underlying features in a safe filter manner. Finally, we present a set of experiments to illustrate the performance of our method.

Notations. For any $n \in \mathbb{N}^*$, we denote the set $\{1, \dots, n\}$ by $[n]$. The notation $A_n \xrightarrow{P} 0$ indicates that the random variables A_n converge to 0 in probability. For a vector $x \in \mathbb{R}^d$ and $a \in [d]$, $x^{(a)}$ denotes the a-th component of x, and $\|x\|_2$ represents its ℓ_2 norm. The identity matrix of size n is denoted by I_n.

For a matrix $M \in \mathbb{R}^{n\times m}$, $\|M\|_*$ denotes its trace norm (sum of singular values), $\mathrm{rank}(M)$ its rank (number of non-zero singular values), $\mathrm{tr}(M)$ its trace (when $n = m$), $\|M\|_{\mathrm{op}}$ its operator norm (largest singular value), $\|M\|_F$ its Frobenius norm (the ℓ_2 norm of its singular values), $M_{i,j}$ its (i,j)-th element, and $\|M\|_\infty$ its infinity norm (maximum absolute element value). The matrix $\mathrm{Diag}(M)$ is diagonal with the diagonal elements of M as its diagonal, and $M \succeq N$ (or $M \succ N$) indicates that $M - N$ is positive semi-definite (or positive definite), assuming M and N are symmetric.

For a tensor $W \in \mathbb{R}^{n\times d\times m}$, $W_{:,:,k}$ represents the matrix in $\mathbb{R}^{n\times d}$ obtained by slicing along the third dimension, and $W_{i,j,:}$ is the vector in $\mathbb{R}^m$ obtained by slicing along both the first and second dimensions. Multiplication of a tensor W by a vector $x \in \mathbb{R}^m$ along the third dimension is defined as $Wx = \sum_{k=1}^m W_{:,:,k}x^{(k)} \in \mathbb{R}^{n\times d}$, and for a matrix $Q \in \mathbb{R}^{d\times d}$, $WQW^T \in \mathbb{R}^{m\times m}$ is defined as $\sum_{i,j,l} W_{i,j,:}Q_{j,l}W^T_{i,l,:}$.

In a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, for functions $f, g \in \mathcal{H}$, $\|f\|_{\mathcal{H}}$ represents the RKHS norm, and $\langle f, g\rangle$ denotes the inner product in $\mathcal{H}$.

2 Problem Setting and Estimators

We consider a classical supervised learning setting with a training set $(x_i, y_i)_{i\in[n]}$ of factor/response pairs. The training data are assumed to be independent realisations of the random variables (X, Y), which have a probability measure ρ on $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$. The random variable Y is assumed to have finite variance, and the marginal probability measure of X is denoted ρ_X. Our objective is to find the function f^* that minimises the population risk $\mathcal{R}(f) = \mathbb{E}_\rho\left((y - f(x))^2\right)$ over the class of functions $L^2(\rho_X) := \{f : \mathcal{X} \to$

$\mathcal{Y}, \int_{\mathcal{X}} f(x)^2 d\rho_X < \infty\}$. It is known that the minimiser $f^*(x)$ is $\mathbb{E}_\rho(Y|X = x)$. We then define the empirical risk as $\widehat{\mathcal{R}}(f) := \frac{1}{n}\sum_{i=1}^n (y_i - f(x_i))^2$.

For convenience, we use the quadratic loss in our analysis, though it could be easily adapted to any convex loss function, which is a strength of our method. We adopt the framework of regularised empirical risk minimisation due to its versatility, allowing it to be applied to a wide range of problems where a risk can be defined. Although we focus on typical supervised learning scenarios using the square loss for simplicity, our long-term goal is to extend this approach to other problems, such as in control, which is not so easily feasible with moment-based methods [Xia et al., 2002, Li, 1991].

We further impose the following multi-index model [Xia, 2008, Yuan, 2011], as discussed in the introduction.

Assumption 1 (Multi-Index Model)**.** $\forall x \in \mathcal{X},\ f^*(x) = g^*(P^T x)$*, with* $g^* : \mathbb{R}^s \mapsto \mathcal{Y}$*, with* $P \in \mathbb{R}^{d\times s}$ *($s \leq d$, unknown) a full-rank matrix, i.e.,* $\operatorname{rank} P = s$*.*

This means that the dependency of Y on X is fully characterised by $P^T X \in \mathbb{R}^s$ which has a smaller dimension than $X \in \mathbb{R}^d$. It is therefore of interest to estimate P. Some methods suffer from first estimating g and then P, leading to stronger bias on P [Xia et al., 2002]. Instead, we aim to estimate both g^* and P, or rather f^* and P, at the same time. We further remark that P cannot be exactly recovered, even in the noiseless case, because multiple (g^*, P) pairs can verify equality to f^*. We can therefore only hope to recover the column space of P.

2.1 Low-Rank Penalty

We remark that under Assumption 1, if the gradient is well-defined, we have $\forall x \in \mathcal{X}, \nabla f^*(x) = P\nabla g(P^T x)$. This was first noticed in effective dimension one by Härdle and Stoker [1989]. Other lines of work use this idea, like the one started by Li [1992]. Therefore, for any $x \in \mathcal{X}$, $\nabla f^*(x)$ belongs to the column space of P. Furthermore, for any $f \in H^1(\rho_X) := \{f \in L^2(\rho_X), \forall a \in [d], \partial f(x)/\partial x^{(a)} \in L^2(\rho_X)\}$, which is a Sobolev space [see Adams and Fournier, 2003], we can write

$$\operatorname{cov}(\nabla f) := \operatorname{cov}(\nabla f(X)) := \mathbb{E}_{\rho_X}\left(\nabla f(X)\nabla f(X)^T\right) \in \mathbb{R}^{d\times d},$$

and we then have $\operatorname{cov}(\nabla f^*) = P \operatorname{cov}\left(\nabla g^*(P^T X)\right) P^T$. We therefore make the next assumption.

Assumption 2 (Full-Rank of Covariance Matrix)**.** *We assume that* $f^* \in H^1(\rho_X)$ *and* $\operatorname{rank}\operatorname{cov}(\nabla f^*) = \operatorname{rank} P = s$*.*

This amounts to assuming that $\operatorname{rank}\operatorname{cov}(\nabla g^*(P^T X)) = s$. The projection $\Pi_P = P(P^T P)^{-1}P^T \in \mathbb{R}^{d\times d}$ on the column space of P is therefore also the projection on the column space of $\operatorname{cov}(\nabla f^*)$.

Following in the footsteps of the Lasso [Tibshirani, 1996] and other sparsity inducing penalties such as the group Lasso [Yuan and Lin, 2006], the fused lasso [Tibshirani et al., 2005], a trace norm penalty for matrix estimation [Bach, 2008], and most similar to us a penalty for variable selection [Rosasco et al., 2013], we consider adding a rank-penalty on the matrix of gradients to the quadratic loss, yielding the following optimisation problem

$$\operatorname*{arg\,min}_{f\in H^1(\rho_X)} \widehat{\mathcal{R}}(f) + 2\tau \operatorname{rank}\operatorname{cov}(\nabla f),$$

with τ a positive regularisation parameter to be chosen.

However, as the rank penalty is non-convex, we replace it by a convex surrogate, i.e.,

$$\|\nabla f\|_* := \operatorname{tr}\left(\sqrt{\operatorname{cov}(\nabla f)}\right),$$

for ease of optimisation, which is a well-known technique, [see Candès and Recht, 2009, Recht et al., 2010, Bach et al., 2012]. We cannot compute this quantity for a fixed f, since ρ_X is not known a priori. We can nevertheless estimate it using the training set. We define

$$\nabla_n f := (\nabla f(x_1)^T, \nabla f(x_2)^T, \dots, \nabla f(x_n)^T)^T/\sqrt{n} \in \mathbb{R}^{n\times d},$$

the (normalised) matrix of gradients at the observed data points, or sample matrix of gradients. We then have that $\nabla_n f^T \nabla_n f \in \mathbb{R}^{d\times d}$ is equal to $\frac{1}{n}\sum_{i=1}^n \nabla f(x_i)\nabla f(x_i)^T$, which estimates $\operatorname{cov}(\nabla f)$, and is actually also equal to the non-centred covariance matrix of $\nabla f(X)$ under the empirical measure of the data. We therefore overload the notation and write $\operatorname{cov}(\nabla_n f) := \nabla_n f^T \nabla_n f$. We then have that our penalty is equal to the trace norm of the sample matrix of gradients

$$\|\nabla_n f\|_* = \operatorname{tr}\left(\sqrt{\operatorname{cov}(\nabla_n f)}\right).$$

2.2 Reproducing Kernel Hilbert Space

We decide to restrict the functions we study to a reproducing kernel Hilbert space $\mathcal{H}$, with a twice differentiable reproducing kernel k. This is motivated by multiple factors. First, this will allow us to efficiently compute gradients instead of using finite differences which are unstable, especially in high-dimensions, as in the RKHS we consider, partial derivatives are bounded linear functionals. An equivalent of the representer theorem (see Lemma 1 below) allows us to actually compute the estimator. Finally, empirical risk minimisation in RKHS is known to avoid the curse of dimensionality if the problem is well-specified, which is the setting we will study in Section 4. Estimation in RKHS also naturally adapts to smoothness, which is an interesting property when considering misspecified settings [Bach, 2024, Chapter 7].

Let us denote k_x the function $t \mapsto k(x,t)$ and $(\partial_a k)_x$ the function $t \mapsto \partial k(x,t)/\partial x^{(a)}$. We recall that $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ is the unique Hilbert space associated to a symmetric positive definite function k, such that $\forall f \in \mathcal{H}, f(x) = \langle f, k_x\rangle_{\mathcal{H}}$. This is the reproducing property [Aronszajn, 1950]. More information on RKHS can be found in Appendix B for the interested reader. We now make a few additional assumptions on the reproducing kernel k of $\mathcal{H}$ and on the data-generating mechanism.

Assumption 3 (Constraints on the Kernel and Outputs)**.**

1. *(Bounded features). There exists $K_1 > 0$ such that $\forall x \in \mathcal{X}, k(x,x) \leq K_1^2$.*
2. *(Regular kernel). k is $\mathcal{C}^2(\mathcal{X},\mathcal{X})$, i.e., twice continuously differentiable, and there exists $K_2 > 0$ such that $\forall a \in [d], x \in \mathcal{X}, \frac{\partial k(s,t)}{\partial s^{(a)}\partial t^{(a)}}|_{s=x,t=x} \leq K_2^2$.*
3. *(Bounded outputs). $\mathcal{Y} \subset [-M, M]$ for some $M \in \mathbb{R}_+$.*

From Zhou [2008, Theorem 1], because k is at least twice differentiable, $(\partial_a k)_x$ also belongs to $\mathcal{H}$ for any $x \in \mathcal{X}$ and further for any $f \in \mathcal{H}, \partial f(x)/\partial x^{(a)} = \langle f, (\partial_a k)_x\rangle_{\mathcal{H}}$. We remark that Assumption 3.1 and 3.2 are verified for many well-known kernels, such as

the Gaussian kernel, the Cauchy kernel, and if $\mathcal{X}$ is bounded, the polynomial, linear and sigmoid kernels. We also remark that $\mathcal{H} \subset H^1(\rho_X)$, because of Assumptions 3.1 and 3.2, see Appendix B.

For computational stability and to make the problem strongly convex, we add a penalty on the norm of f in the RKHS, with small regularisation parameter $\tau\nu$, where ν is fixed and τ has to be chosen in practice, usually through cross-validation. This yields that the final optimisation problem defining our estimator, which is coined KTNGrad (for kernel trace norm of gradients), is

$$\hat{f}_\tau := \operatorname*{arg\,min}_{f \in \mathcal{H}} \left(\hat{\mathcal{R}}_\tau(f) := \hat{\mathcal{R}}(f) + \tau \left(\nu \|f\|_{\mathcal{H}}^2 + 2 \|\nabla_n f\|_* \right) \right). \tag{2.1}$$

We remark that the solution to Equation (2.1) is always defined as the functional on f is coercive and $2\tau\nu$-strongly convex with respect to the $\mathcal{H}$-norm, so that existence and uniqueness of the minimiser is ensured [Ekeland and Témam, 1999], for any $\nu > 0$, regardless of assumptions on f^*. Our setting is therefore similar to that of Rosasco et al. [2013], except that they use $\operatorname{tr}(\sqrt{\operatorname{Diag}(\operatorname{cov}(\nabla f))})$, as a penalty (in our notation, where $\operatorname{Diag}(\operatorname{cov} \nabla f)$ is the matrix in $\mathbb{R}^{d \times d}$ with the diagonal of $\operatorname{cov}(\nabla f)$ as its diagonal, and 0 elsewhere), which reflects their different objective to select variables, rather than features.

Through $\hat{f}_\tau$, we can estimate P by $\hat{P}_s$ defined as the first s leading eigenvectors of $\operatorname{cov}(\nabla_n \hat{f}_\tau)$, if s is known. Otherwise s has to be estimated as well, and we can choose to take $\hat{s} := \operatorname{rank} \operatorname{cov}(\nabla_n \hat{f}_\tau)$ or the number of eigenvalues exceeding a certain threshold as an estimator of s, see Section 3.3. This yields that the estimator of P is $\hat{P}$, which is defined as the first $\hat{s}$ eigenvectors of $\operatorname{cov}(\nabla_n \hat{f}_\tau)$. Due to the number of notations, we give a recap of all the quantities in Table 2.1 in Appendix A.

3 Methodology of KTNGrad

In this section, we describe the methodology of the estimator KTNGrad, which first consists in rewriting the problem from Equation (2.1) into a parametric problem using an equivalent of the representer theorem, solving it through an iterative procedure and then considering the eigenvectors of the empirical covariance matrix of the gradients.

3.1 Parametric Formulation

Closely following a result from Rosasco et al. [2013], there is an equivalent of the representer theorem, allowing us to transform our problem in the RKHS $\mathcal{H}$ into a parametric one in $\mathbb{R}^{n(d+1)}$.

Lemma 1 (Representer Theorem)**.** *Under Assumption 3.2, there exists (potentially multiple) $\theta_* \in \mathbb{R}^{n(d+1)}$ such that the solution to Equation (2.1) can be written as*

$$\hat{f}_\tau = \sum_{i=1}^{n} \frac{1}{n} \theta_*^{(i)} k_{x_i} + \sum_{i=1}^{n} \sum_{a=1}^{d} \frac{1}{n} \theta_*^{(i+na)} (\partial_a k)_{x_i}. \tag{2.2}$$

The proof can be found in Appendix B. We can then restrict the optimisation to functions $f \in \mathcal{H}$ that can be expressed similarly to $\hat{f}_\tau$. This allows us to fully rewrite Equation (2.1) in a parametric form, with a change of variables to improve the conditioning of the problem.

Lemma 2 (Parametric Optimisation Problem). *Under Assumption 3.2, any* $\theta_* \in \mathbb{R}^{n(d+1)}$ *from Equation (2.2) is such that* $V^{1/2}\theta_* = \beta_*$, *with*

$$\beta_* = \underset{\beta \in \mathbb{R}^{n(d+1)}}{\arg\min} \frac{\|Y - U\beta\|_2^2}{n} + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{2\tau}{\sqrt{n}}\|W\beta\|_*, \tag{2.3}$$

where $Y := (y_1, \ldots, y_n)^T \in \mathbb{R}^n, U \in \mathbb{R}^{n \times n(d+1)}, V \in \mathbb{R}^{n(d+1) \times n(d+1)}, W \in \mathbb{R}^{n \times d \times n(d+1)}$ *are defined in Definition 1. We also have* $W\beta_*/\sqrt{n} = \nabla_n \hat{f}_\tau$.

The proof can be found in Appendix C. Remark that Y is the vector of responses and V is the Gram matrix of the set $\{k_{x_i}, (\partial_a k)_{x_i} | i \in [n], a \in [d]\}$ in $\mathcal{H}$. U and W are extracted from $V^{1/2}$. Moreover β_* is unique but θ_* may not be if V is not invertible. We give these matrix calculations for the Gaussian kernel in Definition 2, which is the choice we use in practice in the numerical experiments of Section 5.

If β_* is known, we can derive $\hat{f}_\tau$ using Lemma 1 with any $\theta_* \in \mathbb{R}^{n(d+1)}$ such that $\beta_* = V^{1/2}\theta_*$. Additionally, $\hat{P}$ can be easily obtained by taking the right singular vectors (corresponding to non-zero singular values) of $\nabla_n \hat{f}_\tau = W\beta_*/\sqrt{n}$. Therefore, knowing β_* is sufficient to estimate the regression function, the feature space, and its dimension.

3.2 Optimisation Procedure

In this subsection, we address the central problem of estimating β_*. Given that it is a convex problem, there are multiple optimisation methods available. We choose to use a reweighted method [Bach et al., 2012], for reasons discussed at the end of this subsection.

For computational stability, we modify the optimisation problem of Equation (2.3) to the following

$$\beta'_* = \underset{\beta \in \mathbb{R}^{n(d+1)}}{\arg\min} \frac{1}{n}\|Y - U\beta\|_2^2 + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{2\tau}{\sqrt{n}} \operatorname{tr}\left(\sqrt{(W\beta)^T(W\beta) + n\epsilon I_d}\right), \tag{2.4}$$

with $\epsilon > 0$ being a small constant. This transformation converts Equation (2.1) into $\hat{f}'_\tau = \arg\min_{f \in \mathcal{H}} \hat{\mathcal{R}}'_\tau(f)$, as defined in Appendix A to account for ϵ.

Reweighted formulation. We follow the reweighted least squares approach as outlined in [Bach et al., 2012]. The key idea is based on the fact that for any $\beta \in \mathbb{R}^{n(d+1)}$,

$$\operatorname{tr}\left(\sqrt{(W\beta)^T(W\beta) + n\epsilon I_d}\right) = \frac{1}{2} \inf_{\substack{\Lambda \in \mathbb{R}^{d \times d} \\ \Lambda \succ 0}} \left(\operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right) + \operatorname{tr}\left(\Lambda + n\epsilon\Lambda^{-1}\right)\right),$$

where $\Lambda \succ 0$ denotes that Λ is positive definite. To minimise this quantity, we alternate between minimising with respect to β while keeping Λ fixed, and minimising with respect to Λ while keeping β fixed. This iterative process continues until the dual gap condition, parameterised by δ, is satisfied. Closed-form formulas for the updates (see Lemma 4 in Appendix C) and the duality gap (see Lemma 5 in Appendix C) facilitate this process.

The introduction of ϵ ensures that the matrix Λ does not become singular during the updates. As a computational trick, ϵ should be chosen to be small to avoid significantly disturbing the problem. The optimisation procedure we employ is an example of a reweighted-ℓ_2 method. For more details, refer to Argyriou et al. [2008] and Bach et al. [2012].

We present the pseudo-code for the method in Algorithm 1 below. This algorithm includes the procedure for selecting the dimension and relevant features, as discussed in Section 3.3.

Input: k a kernel, $X \in \mathbb{R}^{n\times d}, Y \in \mathbb{R}^n$, $\tau > 0, \nu > 0, \epsilon > 0, \delta > 0$

1. Compute $V \in \mathbb{R}^{n(d+1)\times n(d+1)}$, the Gram matrix of $\{k_{x_i}, (\partial_a k)_{x_i} | i \in [n], a \in [d]\}$ in $O(n^2d^3)$.
2. Compute $U \in \mathbb{R}^{n\times n(d+1)}$, $W \in \mathbb{R}^{n\times d\times n(d+1)}$ as in Definition 1 in $O(n^3d^3)$.
3. $t \leftarrow 0$
4. $\beta_t \leftarrow$ solution to $(U^TU + \tau\nu I_{n(d+1)})\beta = U^TY$ in $O(n^3d^3)$.

while *the dual gap for* β_t *from Equation* (2.7) *in Appendix C is larger than* δ **do**

1. $\Lambda_t \leftarrow \sqrt{(W\beta_t)^T(W\beta_t) + n\epsilon I_d}$ in $O(n^2d^2 + d^3)$.
2. $\beta_{t+1} \leftarrow$ solving $(U^TU + \tau\nu I_{n(d+1)} + \sqrt{n}\tau(W\Lambda_t^{-1}W^T))\beta = U^TY$ in $O(n^3d^4)$.
3. $t \leftarrow t + 1$

end

1. $\nabla_n f_{t+1} \leftarrow W\beta_{t+1}/\sqrt{n}$ in $O(n^2d^2)$.
2. Obtain Singular Value Decomposition of $\nabla_n f_{t+1}$ in $O(nd\min(n, d))$.
3. **if** s *unknown* **then**
   $\hat{s} \leftarrow \text{rank}(\nabla_n f_{t+1})$ (potentially computed using some threshold)
   $\hat{P} \leftarrow$ first $\hat{s}$ right singular vectors of $\nabla_n f_{t+1}$.
   else
   $\hat{P}_s \leftarrow$ first s right singular vectors of $\nabla_n f_{t+1}$.
   end

Output: $\hat{s}, \hat{P}$ or $\hat{P}_s$ in $O(n^3d^4)$

Algorithm 1: Pseudocode of KTNGrad.

Convergence of the optimisation. We now consider the guarantees on the convergence of the optimisation procedure. As our problem satisfies the conditions of Beck [2015, Theorem 3.3], we obtain the following convergence result.

Lemma 3 (Convergence of the Optimisation Procedure)**.** *Let* $(f_t)_{t\geq 0}$ *be the sequence of functions in* $\mathcal{H}$ *with coefficients* $(\theta_t)_{t\in\mathbb{N}^*}$ *generated by Algorithm 1 through the sequence* $(\beta_t)_{t\in\mathbb{N}^*}$*. Under Assumption 3.2, there exists a constant* $C > 0$ *defined in Equation* (2.8) *such that for all* $t \in \mathbb{N}^*$*,*

$$\hat{\mathcal{R}}'_\tau(f_t) - \hat{\mathcal{R}}'_\tau(\hat{f}'_\tau) \leq \frac{C}{t} \quad \textit{and} \quad \|f_t - \hat{f}'_\tau\|_{\mathcal{H}} \leq \sqrt{\frac{2C}{\tau\nu t}}.$$

The proof can be found in Appendix C. We note that with the introduction of ϵ, the

functional from Equation (2.4) becomes differentiable, allowing the use of gradient descent methods. However, in the setting where $\tau = 0$, our method achieves minimisation in a single step, finding the exact solution directly. Given that the penalty is typically small, convergence is observed in just a few iterations, which is much faster than the theoretical result might suggest, see the experiment in Section 5.1. Additionally, gradient descent is sensitive to ill-conditioning, whereas our method is not. This robustness is characteristic of reweighted formulations [Daubechies et al., 2010].

3.3 Choice of Dimension

We use the leading eigenvectors of $\operatorname{cov}(\nabla_n \hat{f}_\tau) \in \mathbb{R}^{d\times d}$ as an estimator of P. The number of eigenvectors to retain must be determined: either s is known, or it needs to be estimated using the data, which is typically the case in practice.

In Section 4, we discuss the statistical properties of $\operatorname{cov}(\nabla_n \hat{f}_\tau)$. We note that its rank, $\hat{s}$, is asymptotically greater than or equal to s (see the proof of Theorem 5). Furthermore, in practice, we only have access to f_t, which converges to $\hat{f}'_\tau$. Therefore, we recommend computing the singular values of $\nabla_n f_t$ and setting $\hat{s}$ to the number of singular values above a certain threshold, multiplied by the trace norm. This is the choice we use in practice in the numerical experiments in Section 5.2, although in the statistical analysis in Section 4 we consider $\hat{s} = \operatorname{rank}\operatorname{cov}(\nabla_n \hat{f}_\tau)$.

3.4 Computational Considerations

The computational complexity of Algorithm 1 is $O(n^3d^4)$, which can be prohibitive in many scenarios. To enhance computational efficiency, the well-studied Nyström approximation [Drineas and Mahoney, 2005] can be employed. This technique replaces the minimisation over the space $\operatorname{span}\{k_{x_i}, (\partial_a k)_{x_i} \mid i \in [n], a \in [d]\}$ by minimisation over $\operatorname{span}\{k_{x_i} \mid i \in S_p\}$ for some set $S_p \subset [n]$ of cardinal $p \leq n$. As p increases with the sample size n, the span $\operatorname{span}\{k_{x_i} \mid i \in S_p\}$ converges to $\mathcal{H}$, the closure of $\operatorname{span}\{k_x \mid x \in \operatorname{support}\rho_X\}$.

Furthermore, as demonstrated in Rudi et al. [2015], by choosing $p := n^\zeta \log(n)$ with $\zeta \in (0, 1]$ (depending on the RKHS and the regularity of the solution) and using sub-sampling to select S_p, the sample complexity is preserved up to a constant factor. This approach effectively reduces the computational cost of training from $O(n^3d^4)$ to $O(p^2nd^2)$, which simplifies to $O(n^{1+2\zeta}\log(n)^2d^2 + d^3)$. Additionally, the storage requirements decrease from $O(n^2d^2)$ to $O(nd + p^2)$, equating to $O(nd + n^{2\zeta}\log(n)^2)$.

4 Statistical Properties

In this section, we discuss the statistical properties of the estimators computed by KT-NGrad, that is the consistency of $\hat{f}_\tau$ in its estimation of f^* and $\hat{P}$ in its estimation of the underlying feature space P and its dimension s.

4.1 Estimation of f^*

The estimators $\hat{f}_\tau$, $\hat{P}$, and $\hat{s}$ exist whether or not f^* belongs to $\mathcal{H}$. However, most of our theoretical results hold only in the well-specified setting, except consistency of the expected risk without explicit rates.

Assumption 4 (Well-Specified Model)**.** *The model is well-specified: $f^* \in \mathcal{H}$.*

Nonetheless, we believe our results could be extended, especially for well-chosen regular kernels such that $\mathcal{H}$ is dense in $(H^1(\rho_X), \|\cdot\|_{H^1(\rho_X)})$. In particular, if the kernel is universal, i.e., $\mathcal{H}$ is dense in $(L^2(\rho_X), \|\cdot\|_{L^2(\rho_X)})$ (see Appendix B), then $\inf_{f\in\mathcal{H}} \mathcal{R}(f) = \mathcal{R}(f^*)$, and we can achieve consistent estimation in the $L^2(\rho_X)$ norm. We provide most results as convergence in probability and leave explicit bounds for future work, which should follow from standard arguments [Caponnetto and De Vito, 2007]. By adapting the proof from Rosasco et al. [2013] to our penalty, we achieve consistency of the expected risk with the square loss.

Theorem 4 (Convergence of Expected Risk)**.** *Under Assumptions 3 and 4, for any $\eta \in (0, 1]$, with probability at least $1 - \eta$,*

$$\mathcal{R}(\hat{f}_\tau) - \inf_{f\in\mathcal{H}} \mathcal{R}(f) \leq C_1 \left(\frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{\tau\nu}} + 1 \right)^2 + \sqrt{\frac{\tau}{\nu}} \frac{d^{5/4}}{n^{1/4}} \right) \log \frac{6 + 2d}{\eta} + \tau \left(\nu \|f^*\|_{\mathcal{H}}^2 + 2\|\nabla f^*\|_* \right),$$

where C_1 does not depend on n, d, τ, ν, or f^*. Under Assumption 3,

$$\mathcal{R}(\hat{f}_{\tau_n}) \xrightarrow{P} \inf_{f\in\mathcal{H}} \mathcal{R}(f)$$

for any positive sequence $(\tau_n)_{n\in\mathbb{N}}$ such that $\tau_n \to 0$ and $(\sqrt{n}\tau_n)^{-1} \to 0$ as $n \to \infty$.

In the well-specified case, it is notable that the rate of convergence does not depend on d in the powers of n, which is typical of well-specified models in RKHS. Hence we avoid the curse of dimensionality [Bellman, 1966] but at the cost of a strong assumption on the function f^*. The dependence on $n^{1/4}$ (compared to the usual $n^{1/2}$) appears in the estimation of the trace norm penalty in our proofs because we did not make additional assumptions on the smallest eigenvalue of $\operatorname{cov}(\nabla f^*)$ to remain general.

To obtain results on the estimation of P, we achieve consistency in the $\mathcal{H}$-norm, as shown in Lemma 7 in Appendix D, in a similar manner to Rosasco et al. [2013].

4.2 Estimation of the Underlying Subspace P

As discussed in Section 2, we cannot recover P exactly, as there may be multiple pairs (g^*, P) that satisfy $g^*(P^T\cdot) = f^*$. The estimator $\hat{P}$ is defined as the eigenvectors of $\operatorname{cov}(\nabla_n \hat{f}_\tau)$ associated with non-zero eigenvalues, i.e., the first $\hat{s} = \operatorname{rank} \operatorname{cov}(\nabla_n \hat{f}_\tau)$ eigenvectors. If s is known, $\hat{P}_s$ is compose of the s leading eigenvectors.

To measure the error between two projectors P and Q, where the projections matrices are $\Pi_P = P(P^TP)^{-1}P^T$ and $\Pi_Q = Q(Q^TQ)^{-1}Q^T$, we use the Frobenius norm, as in Dalalyan et al. [2008]

$$\|\Pi_P - \Pi_Q\|_F^2. \tag{2.5}$$

However, we are also interested in safe filters, i.e., those where the image of Π_P is included in that of Π_Q. Therefore, we define the following error, combining Xia et al. [2002] and Rosasco et al. [2013]

$$\|\Pi_P(I_d - \Pi_Q)\|_F^2. \tag{2.6}$$

If we have a safe filter (i.e., no information is lost), the error is 0. In the worst case, when all information is lost (i.e., $\Pi_Q = 0$), the error equals s. Note that this error does not penalise $\hat{P}$ with higher $\hat{s}$; in fact, the error is 0 for any matrix Q of rank d.

In Appendix D, we use the consistency in the $\mathcal{H}$-norm (Lemma 7) to establish the consistency of $\operatorname{cov}(\nabla_n \hat{f}_\tau)$ as an estimator of $\operatorname{cov}(\nabla f^*)$ (Lemma 8). This leads to the consistency of the eigenvectors (Lemma 9). Consequently, if the subspace dimension s is known, the Frobenius error converges to zero. If s is unknown, using the estimator $\hat{s} = \operatorname{rank} \operatorname{cov}(\nabla_n \hat{f}_\tau)$ and the property that rank is lower semi-continuous ensures an asymptotically safe filter.

Theorem 5 (Convergence of Subspace Estimator)**.** *Under Assumptions 1, 2, 3, and 4, for any positive sequence $(\tau_n)_{n\in\mathbb{N}}$ such that $\tau_n \to 0$ and $(\sqrt{n}\tau_n^2)^{-1} \to 0$ as $n \to \infty$, we have*

$$\|\Pi_P - \Pi_{\hat{P}_s}\|_F^2 \xrightarrow{P} 0 \quad \textit{and} \quad \|\Pi_P(I_d - \Pi_{\hat{P}})\|_F^2 \xrightarrow{P} 0.$$

A stronger result would be to precisely estimate the projection onto P, achieving a small Frobenius norm $\|\Pi_P - \Pi_{\hat{P}}\|_F^2$. For this to hold, we need $\hat{s} \xrightarrow{P} s$. However, in the literature on low-rank matrix estimation, it is well-documented that square loss minimisation using the naive trace norm penalty can be inconsistent in certain cases [Bach, 2008], highlighting the need for an adaptive version of the method.

4.3 Adaptive Method for Consistent Dimension Estimation

To enhance the consistency of our method in estimating the underlying subspace dimension, we introduce an adaptive version of KTNGrad, inspired by transformations in the Lasso framework [Zou, 2006] and low-rank matrix estimation [Bach, 2008].

The process begins by obtaining the ridge estimator $\hat{f}_R$, defined as

$$\hat{f}_R = \operatorname*{arg\,min}_{f\in\mathcal{H}} \hat{\mathcal{R}}(f) + \nu_R \|f\|_{\mathcal{H}}^2,$$

where, according to the representer theorem, $\hat{f}_R$ can be expressed as $\hat{f}_R = \sum_{i=1}^n \frac{1}{n}\alpha_R^{(i)} k_{x_i}$, with $\alpha_R = (K + \nu_R I_n)^{-1} Y \in \mathbb{R}^n$ and K defined in Definition 1.

Next, we compute the covariance matrix of the gradients $\operatorname{cov}(\nabla_n \hat{f}_R)$, which is given by $(\tilde{Z}\alpha_R)^T \tilde{Z}\alpha_R/n$, where $\tilde{Z}$ is defined in Definition 1. The eigenvalue decomposition of this matrix yields $\operatorname{cov}(\nabla_n \hat{f}_R) = V_R \operatorname{diag}(s_R) V_R^T$, with $V_R \in \mathbb{R}^{d\times d}$. We then define a diagonal scaling matrix $\Gamma = V_R \operatorname{diag}(s_R)^{-\gamma} V_R^T$ for some $\gamma \in (0, 1]$. The adaptive penalty is introduced by replacing $\|\nabla_n f\|_*$ with $\|\nabla_n f \Gamma\|_*$, leading to the adaptive estimator

$$\hat{f}_\tau^A = \operatorname*{arg\,min}_{f\in\mathcal{H}} \left(\hat{\mathcal{R}}_\tau(f) := \hat{\mathcal{R}}(f) + \tau\left(\nu\|f\|_{\mathcal{H}}^2 + 2\|\nabla_n f\Gamma\|_*\right)\right).$$

The matrix $\hat{P}^A$ is then composed of the eigenvectors corresponding to the non-zero eigenvalues of $\operatorname{cov}(\nabla_n \hat{f}_\tau^A)$, and $\hat{s}^A := \operatorname{rank} \operatorname{cov}(\nabla_n \hat{f}_\tau^A)$ provides the estimated dimension.

The rationale behind this adaptive approach lies in leveraging the consistent estimation properties of $\hat{f}_R$. Since $\hat{f}_R$ already provides a consistent estimator of f^*, the eigenpairs of $\operatorname{cov}(\nabla_n \hat{f}_R)$ can reliably approximate those of $\operatorname{cov}(\nabla f^*)$. The new adaptive penalty which uses these eigenpairs through Γ further enhances this by penalising smaller eigenvalues more heavily, effectively pushing them toward zero.

This adaptive method mirrors the strategy used in reweighted ℓ_1 algorithms [Candès et al., 2007], which iteratively adjust penalties to better approximate a concave objective. In this case, the adaptive procedure can be seen as a single step of such an algorithm, making it both practical and theoretically sound. The adaptive method can be combined with the Nyström approximation to improve computational costs.

The theoretical results from the original method can be seamlessly adapted to this new framework by substituting W with $W\Gamma$. This modification ensures that the convergence properties and consistency results remain intact. The only more involved adjustment requires adapting the consistency result in the $L^2(\rho_X)$ norm (Theorem 4) to accommodate the new penalty structure. However, achieving stronger consistency results, such as accurate rank estimation as demonstrated in Bach [2008], would require further development.

5 Numerical Experiments

In this section, we discuss the empirical properties of KTNGrad. The technical details of the experiments are available in Appendix F. The Python implementation of KTNGrad is fully integrated with Scikit-learn [Pedregosa et al., 2011], allowing for easy incorporation into standard machine learning workflows. The complete source code, including scripts for replicating the experiments, can be found at `https://github.com/BertilleFollain/KTNGrad`. We always consider KTNGrad with the Gaussian kernel.

We evaluate the performance using three key metrics: the R^2 score for prediction performance, the feature learning score and the dimension score.

R^2 score. This is measured by the R^2 coefficient of determination, a widely recognised statistic in regression analysis [Wright, 1921]. The R^2 value ranges from $-\infty$ to 1, where 1 indicates perfect prediction, 0 means the model is equivalent to using the mean of the target values, and negative values suggest the model performs worse than this baseline. The R^2 score is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

where y_i are the true values, $\hat{y}_i$ are the predicted values, $\bar{y}$ is the mean of the true values, and n is the number of samples. We can also evaluate the R^2 score on the test set.

Feature learning score. This score, ranging from 0 to 1, evaluates the model's ability to identify the true feature space, which is represented by the matrix $P \in \mathbb{R}^{d\times s}$. The score is relevant when all features are equally important and is computed as follows: first, the feature matrix $\hat{P}_s$ is estimated by extracting the leading s eigenvectors from $\operatorname{cov} \nabla_n \hat{f}_\tau$. We then calculate the projection matrices $\pi_{\hat{P}_s}$ and π_P, and define the score as

$$\text{Feature score} = 1 - \frac{\|\pi_P - \pi_{\hat{P}_s}\|_F^2}{\text{normalisation}},$$

where the normalisation term is $2s$ if $s \leq n_{\text{features}}/2$, and $2d - 2s$ otherwise. When $d = k$, the score is defined as 1.

Dimension score. This metric assesses the accuracy of the model in estimating the true dimensionality of the feature space and ranges from 0 to 1. The dimension $\hat{s}$ is determined by counting the significant eigenvalues of $\operatorname{cov} \nabla_n \hat{f}_\tau$ using a threshold. The dimension score is then given by

$$\text{Dimension score} = 1 - \frac{|\hat{s} - s|}{\text{denominator}},$$

where the denominator is $d - s$ if $s \leq d/2$, and s otherwise.

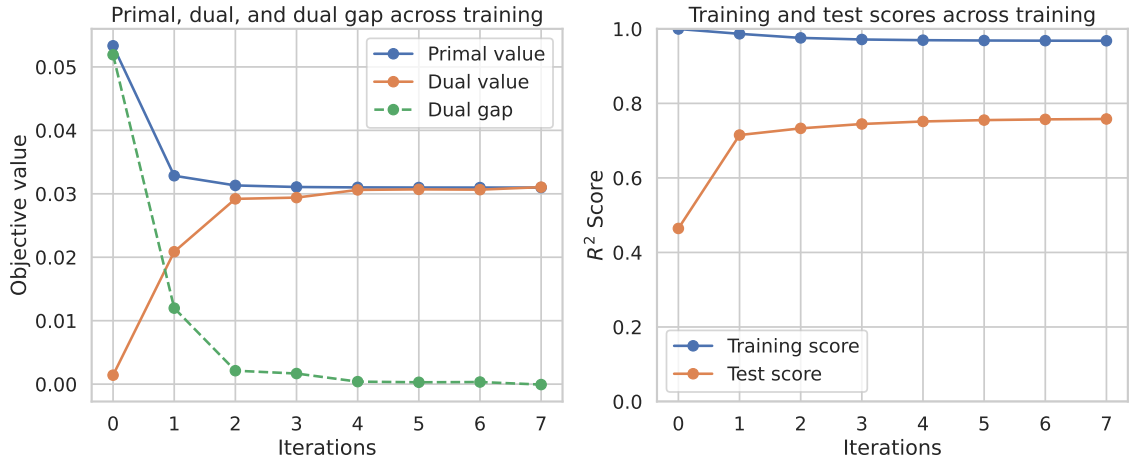


Figure 2.1: Primal, dual and scores across iterations during the optimisation procedure.

5.1 Optimisation Behaviour

In this experiment, we present the optimisation behaviour of KTNGrad and justify the claim made in Section 3.2 that the optimisation procedure converges in few iterations. We train KTNGrad on a single dataset and present the primal and dual value as well as the dual gap across training in the first sub-figure of Figure 2.1. In the second one, we display the R^2 score on the training set and the test set across iterations.

The synthetic data set has $n = 200$ samples and $d = 10$ features. The features were generated uniformly from $[-1, 1]^d$. We sample P uniformly from the set of $d \times d$ orthogonal matrices before truncating it to size $d \times s$ where $s = 2$. The target values y were generated as $y = \left|\sum_{a=1}^{s} \sin\left(P^\top X\right)_a\right| + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.1^2)$ represents Gaussian noise. An independent test set of $n_{\text{test}} = 100$ samples was generated similarly.

We observe that the dual gap closes rapidly, meaning that we have reached the unique minimiser of our convex minimisation problem in very few iterations. This compensates the fact that each iteration is quite computationally costly. In terms of prediction performance, the regularisation benefits the test error which is shown to improve across the iterations.

5.2 Performance dependency on sample size n and data dimension d

We now compare the performance of different estimators against varying sample size and data dimension. The estimators we consider are KTNGrad with the Gaussian kernel, KRR (basic kernel ridge regression with the Gaussian kernel), PyMave and KTNGrad, retrained. PyMave is simply MAVE from Xia et al. [2002] adapted to Python and combined with the regressor MARS from Friedman [1991] as MAVE is not a prediction method. To fairly compare the prediction performance which stems from the feature learning, we also consider KTNGrad, retrained, which consists in a MARS regressor trained on the features learnt by KTNGrad, which are the leading eigenvectors of $\text{cov}(\nabla_n \hat{f}_\tau)$, with their number chosen using a threshold, see Appendix F.

The datasets were all generated as follows. The input data X was uniformly sampled from $[-1, 1]^d$. The projection matrix P was sampled uniformly from the orthogonal group before being truncated to contain $s = 3$ projections. The target values were generated as $y = \left|\sum_{a=1}^{s} \sin\left(P^\top X\right)_a\right| + \epsilon$, where ϵ represents Gaussian noise with a standard deviation of 0.15. An independent test set comprising $n_{\text{test}} = 201$ samples was generated similarly.

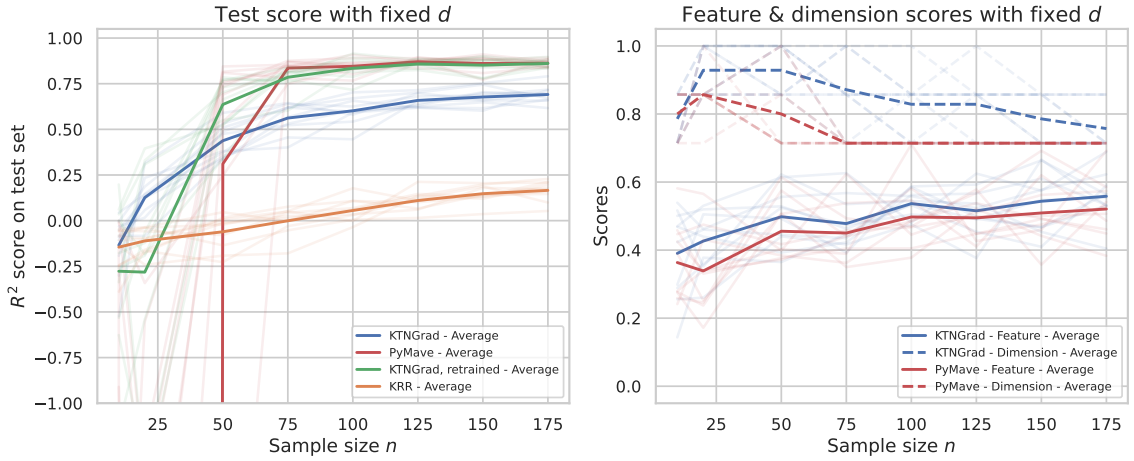


Figure 2.2: Performance with varying sample size n.

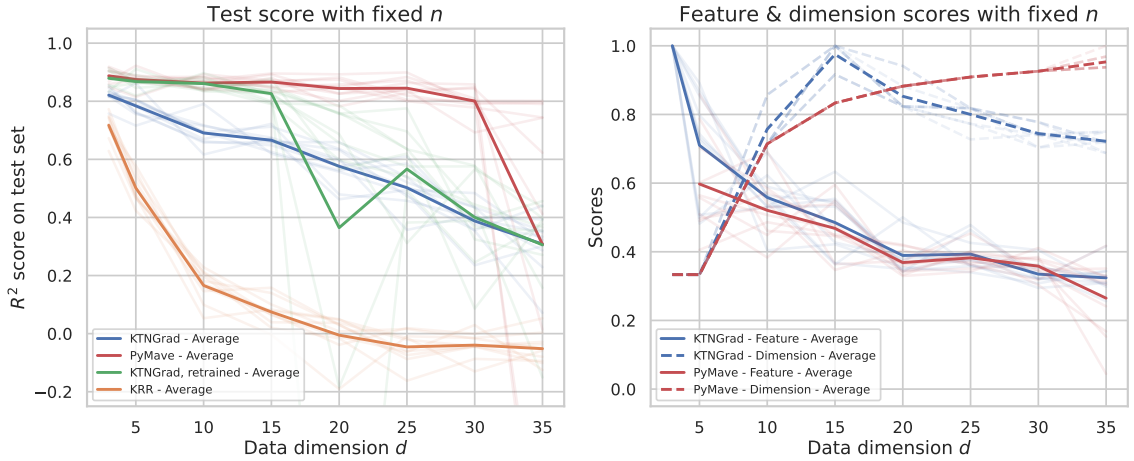


Figure 2.3: Performance with varying data dimension d.

Each experimental setup was repeated 10 times to ensure statistical reliability, and the results were averaged across these repetitions.

In Figure 2.2, the dimension d is fixed at 10, while the sample size n varies from 10 to 175. The results show that KRR performs poorly in terms of the R^2 score. Meanwhile, PyMave struggles with small sample sizes but outperforms other methods once a sufficient number of samples is reached, performing similarly to KTNGrad, retrained. This behaviour is expected, as the second sub-figure illustrates that both PyMave and KTNGrad exhibit comparable performance in feature learning and dimension estimation, with KTNGrad having a slight edge. Additionally, the significant improvement in prediction performance due to retraining KTNGrad highlights a well-known limitation of regularised methods: the regularisation necessary for effective feature or variable selection can negatively impact the quality of predictions [Hastie et al., 2001, Section 3.8.5], as seen in the basic version of KTNGrad.

In Figure 2.3, the sample size n was fixed to 175 while the data dimension d varied from 3 to 35. Similarly to the previous figure, we observe that the performance of KRR is inferior to the others. On the second sub-figure, we observe that the features are learnt similarly by KTNGrad and PyMave (notice however that PyMave did not run successfully when $d = s = 3$). Nonetheless the threshold method we have chosen to

determine the dimension for KTNGrad is not as effective as that of PyMave, which also explains the discrepancy in terms of R^2 score between KTNGrad, retrained and PyMave. The basic version of KTNGrad is not so robust to high dimension as PyMave, but this flaw might be alleviated by using KTNGrad, retrained with a better dimension selection technique.

6 Conclusion

In this chapter, we introduced a novel method for joint linear feature learning and function estimation by incorporating a trace norm penalty on the sample matrix of gradients. Leveraging the unique properties of reproducing kernel Hilbert spaces (RKHS) which include their own partial derivatives, we developed a computationally feasible approach based on convex optimisation. Our method, KTNGrad, avoids strong assumptions about the data distribution and demonstrates convergence to the minimal risk at a rate that does not depend exponentially on the data dimension, an expected outcome in well-specified settings within RKHS. Moreover, KTNGrad effectively recovers the underlying features in a safe filter manner.

The numerical experiments showed that KTNGrad is competitive with state-of-the-art methods like MAVE in learning features in some settings. However, several challenges remain. Our statistical analysis relies on the strong assumption that the regression function belongs to the RKHS. Additionally, KTNGrad performs less well than MAVE in selecting the dimension of the underlying feature space and is computationally expensive. Future work could include deriving explicit convergence rates for feature learning and studying misspecified settings. The consistency in dimension estimation of the adaptive method could also be studied. Finally, extending our framework to support additional loss functions would broaden its applicability to a wider range of problems.

However, the method is inherently slightly flawed, a critique that is valid for the work of Rosasco et al. [2013] as well, due to the choice of function space. Indeed, we have made two different assumptions that are not actually compatible when considered with the usual kernels: first that f^* belongs to the regular RKHS $\mathcal{H}$ and that the multi-index model is verified, i.e., that there exists g^* and P such that $f^* = g^*(P^\top \cdot)$. Indeed, in our work and that of Rosasco et al. [2013], the Gaussian kernel was taken as an example, even though f^* belonging to the corresponding RKHS requires in particular that all first-order derivatives of f^* are square integrable w.r.t the Lebesgue measure on $\mathbb{R}^d$ [Bach, 2024, Chapter 7]. In the basic case of one relevant variable $f^*(x) = g^*(x^{(1)})$, this means that $\int_{\mathbb{R}^d} \left((g^*)'(x^{(1)})\right)^2 \mathrm{d}x^{(1)} \dots \mathrm{d}x^{(d)} < \infty$, which is not possible except in edge cases.

While the two assumptions can be true simultaneously if they are only considered approximately, this limits the appropriateness and efficiency of the method. Despite these issues, this work represents a promising step toward developing methods for multi-index models using the regularised empirical risk minimisation framework. In the next chapters, we will consider different function spaces that are better adapted to feature learning.

Appendix

A Notations and Definitions

We first (re-)introduce some notations used in the main text and the appendix in Table 2.1.

Table 2.1: Notations of the main text.

Symbol	Definition	Space
(X, Y)	factor/response pair of random variables	$\mathcal{X} \times \mathcal{Y}$
n	number of data pairs in training set	$\mathbb{N}$
d	$\mathcal{X} \subset \mathbb{R}^d$	$\mathbb{N}$
ρ	Joint distribution of (X, Y)	
ρ_X	Marginal distribution of X	
$L^2(\rho_X)$	$\{f : \mathcal{X} \to \mathcal{Y}, \int_{\mathcal{X}} f(x)^2 d\rho_X < +\infty\}$	
$H^1(\rho_X)$	$\{f \in L^2(\rho_X), \forall a \in [d], \partial f(x)/\partial x^{(a)} \in L^2(\rho_X)\}$	
$\mathcal{H}$	Reproducing kernel Hilbert space	
k	Reproducing kernel associated to $\mathcal{H}$	$\mathcal{X} \to \mathcal{X}$
K_1	$\forall x \in \mathcal{X}, k(x, x) \leq K_1^2$	$\mathbb{R}^+$
K_2	$\forall x \in \mathcal{X}, \frac{\partial k(s,t)}{s^{(a)} t^{(a)}}\|_{s=x,t=x} \leq K_2^2$	$\mathbb{R}^+$
M	$\mathcal{Y} \subset [-M, M]$	$\mathbb{R}^+$
$\operatorname{cov} \nabla f$	$\mathbb{E}_{\rho_X}\left(\nabla f(x) \nabla f(x)^T\right)$	$\mathbb{R}$
$\operatorname{cov} \nabla_n f$	$\frac{1}{n}\sum_{i=1}^n \nabla f(x_i) \nabla f(x_i)^T$	$\mathbb{R}$
$\|\nabla f\|_*$	$\operatorname{tr}\left(\sqrt{\operatorname{cov} \nabla f}\right)$	$\mathbb{R}$
$\|\nabla_n f\|_*$	$\operatorname{tr}\left(\sqrt{\operatorname{cov} \nabla_n f}\right)$	$\mathbb{R}$
τ	Penalisation parameter	$\mathbb{R}^{+*}$
ν	Hyper-parameter	$\mathbb{R}^{+*}$
ϵ	Hyper-parameter	$\mathbb{R}^{+*}$
δ	Hyper-parameter	$\mathbb{R}^{+*}$
$\mathcal{R}(f)$	$\mathbb{E}_{\rho_X}\left((y - f(x))^2\right)$	$\mathbb{R}$
$\mathcal{R}_\tau(f)$	$\mathcal{R}(f) + \tau(\nu\|f\|_{\mathcal{H}}^2 + 2\|\nabla f\|_*)$	$\mathbb{R}$
$\hat{\mathcal{R}}(f)$	$\frac{1}{n}\sum_{i=1}^n (y_i - f(x_i))^2$	$\mathbb{R}$
$\hat{\mathcal{R}}_\tau(f)$	$\hat{\mathcal{R}}(f) + \tau(\nu\|f\|_{\mathcal{H}}^2 + 2\|\nabla_n f\|_*)$	$\mathbb{R}$
$\hat{\mathcal{R}}'_\tau(f)$	$\hat{\mathcal{R}}(f) + \tau\left(\nu\|f\|_{\mathcal{H}}^2 + 2\operatorname{tr}\left(\sqrt{\nabla_n f^T \nabla_n f + \epsilon I_d}\right)\right)$	$\mathbb{R}$
f^*	$\arg\min_{f \in \mathcal{H}} \mathcal{R}(f)$	$\mathcal{H}$
f_τ	$\arg\min_{f \in \mathcal{H}} \mathcal{R}_\tau(f)$	$\mathcal{H}$
$\hat{f}_\tau$	$\arg\min_{f \in \mathcal{H}} \hat{\mathcal{R}}_\tau(f)$	$\mathcal{H}$
$\hat{f}'_\tau$	$\arg\min_{f \in \mathcal{H}} \hat{\mathcal{R}}'_\tau(f)$	$\mathcal{H}$
β_*	See Equation (2.3)	$\mathbb{R}^{n(d+1)}$
β'_*	See Equation (2.4)	$\mathbb{R}^{n(d+1)}$
β_t	Sequence produced by Algorithm 1	$\mathbb{R}^{n(d+1)}$
f_t	function represented by any θ_t such that $V^{1/2}\theta_t = \beta_t$	$\mathcal{H}$
s	$\operatorname{rank} \operatorname{cov}(\nabla f^*)$	$\mathbb{N}$
$\hat{s}$	$\operatorname{rank} \operatorname{cov}(\nabla_n \hat{f}_\tau)$	$\mathbb{N}$
P	Projection to estimate, see Assumption 1	$\mathbb{R}^{d \times s}$
$\hat{P}_s$	Leading first s eigenvectors of $\operatorname{cov} \nabla_n \hat{f}_\tau$	$\mathbb{R}^{d \times \hat{s}}$
$\hat{P}$	Eigenvectors of $\operatorname{cov} \nabla_n \hat{f}_\tau$ for strictly positive eigenvalues	$\mathbb{R}^{d \times \hat{s}}$

We then give the precise form of the matrices used in Equation (2.3) and in Algorithm 1.

Definition 1 (Kernel Matrices)**.** *K is the classical (rescaled) kernel matrix*

$$K := \frac{1}{n}(k(x_i, x_j))_{i,j} \in \mathbb{R}^{n\times n}.$$

For all $a \in [d]$,

$$Z_a := \frac{1}{n}\left(\frac{\partial k(x, x_j)}{\partial x^{(a)}}|_{x=x_i}\right)_{i,j} \in \mathbb{R}^{n\times n},$$

combined to give

$$Z = (Z_1^T, \dots, Z_d^T) \in \mathbb{R}^{n\times nd}.$$

For all $a, b \in [d]$,

$$L_{a,b} := \frac{1}{n}\left(\frac{\partial^2 k(s,t)}{\partial s^{(a)}\partial t^{(b)}}|_{s=x_i,t=x_j}\right)_{i,j} \in \mathbb{R}^{n\times n},$$

combined to give

$$L_a := (L_{a,1}, \dots, L_{a,d}) \in \mathbb{R}^{n\times nd}$$

and

$$L := (L_1^T, \dots, L_d^T)^T \in \mathbb{R}^{nd\times nd}.$$

Combining K, Z, L, we obtain

$$\begin{aligned}
\tilde{U} &:= (K, Z) \in \mathbb{R}^{n\times n(d+1)} \\
V &:= \begin{pmatrix} K & Z \\ Z^T & L \end{pmatrix} \in \mathbb{R}^{n(d+1)\times n(d+1)} \\
\tilde{W}_a &:= (Z_a, L_a) \in \mathbb{R}^{n\times n(d+1)},\ \forall a \in [d] \\
\tilde{W} &:= (W_1, \dots, W_d) \in \mathbb{R}^{n\times d\times n(d+1)}
\end{aligned}$$

W is obtained by stacking the W_a matrices along the second dimension.

For the adaptive version of the method, we also need

$$\tilde{Z} := (Z_a, \dots, Z_d) \in \mathbb{R}^{n\times d\times n},$$

where the Z_a are stacked along the second dimension.

Since V is positive definite as the Gram matrix of $\{k_{x_i}, (\partial_a k)_{x_i} \mid i \in [n], a \in [d]\}$, it has a square root $V^{1/2} \in \mathbb{R}^{n(d+1)\times n(d+1)}$. We define $U \in \mathbb{R}^{n\times n(d+1)}$ as the matrix consisting of the first n rows of $V^{1/2}$. For $W \in \mathbb{R}^{n\times d\times n(d+1)}$, we define $W_{:,:,k} \in \mathbb{R}^{n\times d}$ as the k-th column of $V^{1/2}$ with the first n elements removed and then reshaped into an $n \times d$ matrix. More simply, this can be written as $W_{i,a,k} := V^{1/2}_{i+na,k}$.

We now provide the characterisation of the matrices K, Z, and L for the Gaussian kernel with parameter σ, which is the practical choice we recommend with σ the median of the euclidean distances between the data points.

Definition 2 (Gaussian Kernel Matrices)**.** *For any $x, y \in \mathbb{R}^d$, the Gaussian kernel is defined as $k(x, y) = \exp\left(-\frac{\|x-y\|_2^2}{2\sigma^2}\right)$. Consequently, the matrices are given by*

$$K = \left(\frac{1}{n}\exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right)\right)_{i,j} \qquad Z_a = \left(-\frac{x_i^{(a)} - x_j^{(a)}}{\sigma^2}K_{i,j}\right)_{i,j}, \quad \forall a \in [d],$$

$$L_{a,b} = \left(-\frac{(x_i^{(a)} - x_j^{(a)})(x_i^{(b)} - x_j^{(b)})}{\sigma^4} K_{i,j} \right)_{i,j}, \quad \forall a, b \in [d],\ a \neq b,$$
$$L_{a,a} = \left(-\frac{\left((x_i^{(a)} - x_j^{(a)})^2 - \sigma^2\right)}{\sigma^4} K_{i,j} \right)_{i,j}, \quad \forall a \in [d].$$

B Reproducing Kernel Hilbert Spaces

We recall that $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ is the unique Hilbert space associated to a symmetric positive definite kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, such that $k_x := k(x, \cdot) \in \mathcal{H}$ and $\forall f \in \mathcal{H}, f(x) = \langle f, k_x \rangle_{\mathcal{H}}$. This is the reproducing property [Aronszajn, 1950]. k being a positive definite function means that for any $n \in \mathbb{N}^*, (x_i)_{i\in[n]} \in \mathcal{X}^n$, the matrix $K = (k(x_i, x_j))_{i,j\in[n]} \in \mathbb{R}^{n\times n}$ is positive definite. We further remark that $\mathcal{H}$ is the closure of the set $\{k_x, x \in \mathcal{X}\}$ in $\mathcal{H}$, meaning that every function in $\mathcal{H}$ can be written as the limit when l grows of quantities of the form $\sum_{j=1}^{l} \alpha_j k_{x_j}$ where $\alpha_j \in \mathbb{R}$ and $x_j \in \mathcal{X}$.

One of the key results of reproducing kernel Hilbert space theory is the representer theorem [Wahba, 1990], which shows that many estimators based on finite data sets in RKHS can actually be written as a linear combination of the k_{x_i}, where the x_i are the data points.

However in our setting, the estimator is defined as the argmin of a quantity making use of gradients. From Zhou [2008, Theorem 1], because k is at least twice differentiable, we know that for any $x \in \mathcal{X}, (\partial_a k)_x$ also belongs to $\mathcal{H}$. Moreover $\forall f \in \mathcal{H}, \partial f(x)/\partial x^{(a)} \langle f, (\partial_a k)_x \rangle_{\mathcal{H}}$. First, this allows for efficient computation of the gradients, instead of using finite differences which are computationally costly and unstable. Additionally, this means that an equivalent of the representer theorem (Lemma 1) is available for $\hat{f}_\tau$ too, but using a linear combination of both the k_{x_i} and the $(\partial_a k)_{x_i}$, for $a \in [d]$ and x_i the data points.

Proof of Lemma 1. Let $S := \operatorname{span}\{k_{x_i}, (\partial_a k)_{x_i} | a \in [d], i \in [n]\}$. Since S is a closed subspace of $\mathcal{H}$, we can write any function f of $\mathcal{H}$ as $f = f^{//} + f^{\perp}$ with $f^{//} \in S$ and $\langle f^{\perp}, g \rangle = 0$ for any $g \in S$. We can then rewrite Equation (2.1) as

$$\hat{f}_\tau = \underset{f\in\mathcal{H}, f=f^{//}+f^{\perp}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f^{//}(x_i) \right)^2 + \tau\nu \|f^{//}\|_{\mathcal{H}}^2 + \tau\nu \|f^{\perp}\|_{\mathcal{H}}^2 + 2\tau \|\nabla_n f^{//}\|_*,$$

which is clearly minimised when $f^{\perp} = 0$. This yields that $\hat{f}_\tau$ belongs to S, hence the lemma. □

We remark that Assumptions 3.1 and 3.2 imply that for all $x \in \mathcal{X}$, $\|k_x\|_{\mathcal{H}} \leq K_1$ and $\|(\partial_a k)_x\|_{\mathcal{H}} \leq K_2$. Consequently, for any $f \in \mathcal{H}$ and $x \in \mathcal{X}$, we have $f(x) = \langle f, k_x \rangle_{\mathcal{H}} \leq \|f\|_{\mathcal{H}} \|k_x\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}} K_1$ and $\partial f(x)/\partial x^{(a)} = \langle f, (\partial_a k)_x \rangle_{\mathcal{H}} \leq \|f\|_{\mathcal{H}} \|(\partial_a k)_x\|_{\mathcal{H}} \leq \|f\|_{\mathcal{H}} K_2$. These inequalities are used frequently in the proofs.

Furthermore, this implies that $\mathcal{H} \subset H^1(\rho_X)$, since for any $f \in \mathcal{H}$, the partial derivatives $\partial f/\partial x^{(a)} = \langle f, (\partial_a k)_x \rangle$ exist and are in $L^2(\rho_X)$, given that $\int_{\mathcal{X}} \left(\partial f/\partial x^{(a)} \right)^2 d\rho_X \leq \|f\|_{\mathcal{H}}^2 K_2^2$.

Universal kernels, such as the Gaussian kernel and the Matérn kernels, have the property that the associated RKHS $\mathcal{H}$ is dense in $L^2(\rho_X)$ with respect to the $L^2(\rho_X)$ norm [Sriperumbudur et al., 2011], provided that $\mathcal{X}$ is compact. This density property ensures that functions in $\mathcal{H}$ can approximate any function in $L^2(\rho_X)$ arbitrarily well.

C Optimisation Procedure from Section 3

Section 3 focuses on the methodology and presents results related to the optimisation procedure. We provide theoretical results that were instrumental in developing the pseudo-code for Algorithm 1, as well as proving the parametric formulation and convergence properties discussed in the main text.

The optimisation problem to solve is given by Equation (2.4), but with the penalty term rewritten using Equation (3.2), as follows

$$(\beta'_*, \Lambda_*) = \underset{\substack{\beta\in\mathbb{R}^{n(d+1)} \\ \Lambda\in\mathbb{R}^{d\times d},\ \Lambda\succ 0}}{\arg\min} \frac{1}{n}\|Y-U\beta\|_2^2+\frac{\tau\nu}{n}\|\beta\|_2^2+\frac{\tau}{\sqrt{n}}\big(\operatorname{tr}((W\beta)\Lambda^{-1}(W\beta)^T)+\operatorname{tr}(\Lambda+n\epsilon\Lambda^{-1})\big).$$

C.1 Useful Lemmas for the Pseudo-Code of Algorithm 1

We provide an explicit formula for the minimiser of Equation (C) when either Λ or β is fixed. This enables us to present the alternating optimisation procedure proposed in Algorithm 1.

Lemma 4 (Closed-Form Updates)**.** *Let $\Lambda \in \mathbb{R}^{d\times d}$ be fixed. If we define $\beta(\Lambda) \in \mathbb{R}^{n(d+1)}$ by*

$$\beta(\Lambda) = \underset{\beta\in\mathbb{R}^{n(d+1)}}{\arg\min} \frac{1}{n}\|Y-U\beta\|_2^2+\frac{\tau\nu}{n}\|\beta\|_2^2+\frac{\tau}{\sqrt{n}}\left(\operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right)+\operatorname{tr}(\Lambda+n\epsilon\Lambda^{-1})\right),$$

then

$$\beta(\Lambda) = \left(U^TU\tau\nu I_{n(d+1)}+\sqrt{n}\tau(W\Lambda_t^{-1}W^T)\right)^{-1}U^TY.$$

Conversely, if β is fixed and we define Λ as follows

$$\Lambda(\beta) = \underset{\Lambda\in\mathbb{R}^{d\times d},\Lambda\succ 0}{\arg\min} \frac{1}{n}\|Y-U\beta\|_2^2+\frac{\tau\nu}{n}\|\beta\|_2^2+\frac{\tau}{\sqrt{n}}\left(\operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right)+\operatorname{tr}(\Lambda+n\epsilon\Lambda^{-1})\right),$$

then

$$\Lambda(\beta) = \sqrt{(W\beta)^T(W\beta)+n\epsilon I_d}.$$

Proof of Lemma 4. Let Λ be fixed. We can simplify the definition of $\beta(\Lambda)$ to

$$\beta(\Lambda) = \underset{\beta\in\mathbb{R}^{n(d+1)}}{\arg\min} \frac{1}{n}\|Y-U\beta\|_2^2+\frac{\tau\nu}{n}\|\beta\|_2^2+\frac{\tau}{\sqrt{n}}\operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right).$$

We note that $\operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right) = \beta^T(W\Lambda^{-1}W^T)\beta$. Therefore, the expression within the arg min is differentiable with respect to β, and setting the derivative to zero gives

$$\frac{1}{n}\left(U^TU+\tau\nu I_{n(d+1)}+\sqrt{n}\tau(W\Lambda^{-1}W^T)\right)\beta(\Lambda)-\frac{1}{n}U^TY=0,$$

yielding the final equation

$$\beta(\Lambda) = \left(U^TU+\tau\nu I_{n(d+1)}+\sqrt{n}\tau(W\Lambda^{-1}W^T)\right)^{-1}U^TY.$$

Now, if β is fixed, we can simplify the definition of $\Lambda(\beta)$ to

$$\Lambda(\beta) = \underset{\substack{\Lambda\in\mathbb{R}^{d\times d}\\ \Lambda\succ 0}}{\arg\min}\, \operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right) + \operatorname{tr}(\Lambda + n\epsilon\Lambda^{-1}).$$

We then set the derivative to zero, yielding

$$-(W\beta)^T(W\beta)\Lambda^{-2} + I_d - n\epsilon\Lambda^{-2} = 0.$$

Reorganising, we obtain

$$\Lambda(\beta) = \sqrt{(W\beta)^T(W\beta) + n\epsilon I_d},$$

which is the desired quantity. □

We provide the duality gap condition used to halt the optimisation procedure, following Bach et al. [2012, Section 1.4].

Lemma 5 (Duality Gap)**.** *For the primal problem*

$$\beta'_* = \underset{\beta\in\mathbb{R}^{n(d+1)}}{\arg\min}\, \frac{1}{n}\|Y - U\beta\|_2^2 + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{(W\beta)^TW\beta + n\epsilon I_d}\right),$$

the dual problem is

$$\max_{Z\in\mathbb{R}^{n\times d}, \|Z\|_{op}\le 2\tau/\sqrt{n}} \left(\frac{1}{n}\|Y\|_2^2 - \frac{1}{n}\left(B + \frac{n}{2}\operatorname{tr}(Z^TW)\right)^T A^{-1}\left(B + \frac{n}{2}\operatorname{tr}(Z^TW)\right) - \sqrt{\epsilon}\operatorname{tr}\left(\sqrt{4\tau^2 I_d - nZ^TZ}\right)\right),$$

with

$$\begin{aligned} A &:= U^TU + \tau\nu I_{n(d+1)},\\ B &:= U^TY,\\ \operatorname{tr}(Z^TW) &:= \left(\operatorname{tr}(Z^TW_{:,:,1}), \ldots, \operatorname{tr}(Z^TW_{:,:,n(d+1)})\right)^T \in \mathbb{R}^{n(d+1)}. \end{aligned}$$

An admissible Z for any β can be computed using

$$Z(\beta) = \min\left(1, \frac{2\tau}{\sqrt{n}\|Z_{opt}(\beta)\|_{op}}\right) Z_{opt}(\beta),$$

with $Z_{opt}(\beta)$ being any solution to $\operatorname{tr}(Z_{opt}(\beta)^TW) = \frac{2}{n}(A\beta - B)$.

The dual gap used in Algorithm 1 is then equal to

$$\begin{aligned} &\frac{1}{n}\|Y - U\beta\|_2^2 + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{(W\beta)^TW\beta + n\epsilon I_d}\right)\\ &- \left(\frac{1}{n}\|Y\|_2^2 - \frac{1}{n}\left(B + \frac{n}{2}\operatorname{tr}(Z(\beta)^TW)\right)^T A^{-1}\left(B + \frac{n}{2}\operatorname{tr}(Z(\beta)^TW)\right)\right)\\ &+ \sqrt{\epsilon}\operatorname{tr}\left(\sqrt{4\tau^2 I_d - nZ(\beta)^TZ(\beta)}\right). \end{aligned} \tag{2.7}$$

Proof of Lemma 5. The primal problem can be rewritten as

$$\min_{\beta\in\mathbb{R}^{n(d+1)}} \frac{1}{n}\|Y\|_2^2 - \frac{2}{n}B^T\beta + \frac{\beta^T A\beta}{n} + \frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{(W\beta)^T W\beta + n\epsilon I_d}\right).$$

We add the constraint $\nabla = W\beta$, yielding

$$\min_{\beta\in\mathbb{R}^{n(d+1)},\nabla\in\mathbb{R}^{n\times d},\nabla=W\beta} \frac{1}{n}\|Y\|_2^2 - \frac{2}{n}B^T\beta + \frac{\beta^T A\beta}{n} + \frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{(W\beta)^T W\beta + n\epsilon I_d}\right).$$

With the Lagrangian multiplier Z, we obtain

$$\max_{Z\in\mathbb{R}^{n\times d}} \min_{\beta\in\mathbb{R}^{n(d+1)},\nabla\in\mathbb{R}^{n\times d}} \frac{1}{n}\|Y\|_2^2 - \frac{2}{n}B^T\beta + \frac{\beta^T A\beta}{n} \frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{\nabla^T\nabla + n\epsilon I_d}\right) + \operatorname{tr}\left(Z^T(\nabla - W\beta)\right),$$

or more simply

$$\max_{Z\in\mathbb{R}^{n\times d}} \left(\min_{\beta\in\mathbb{R}^{n(d+1)}} \frac{1}{n}\|Y\|_2^2 - \frac{2}{n}B^T\beta + \frac{\beta^T A\beta}{n} - \operatorname{tr}\left(Z^T(W\beta)\right) + \min_{\nabla\in\mathbb{R}^{n\times d}} \left(\frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{\nabla^T\nabla + n\epsilon I_d}\right) + \operatorname{tr}\left(Z^T\nabla\right)\right)\right).$$

The first minimum is equal to

$$\min_{\beta\in\mathbb{R}^{n(d+1)}} \frac{1}{n}\|Y\|_2^2 - \frac{2}{n}B^T\beta + \frac{\beta^T A\beta}{n} - \operatorname{tr}(Z^T W)^T\beta,$$

which, when differentiated, yields

$$\frac{1}{n}\|Y\|_2^2 - \frac{1}{n}\left(B + \frac{n}{2}\operatorname{tr}(Z^T W)\right)^T A^{-1}\left(B + \frac{n}{2}\operatorname{tr}(Z^T W)\right).$$

For the second minimum, we have

$$\min_{\nabla\in\mathbb{R}^{n\times d}} \frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{\nabla^T\nabla + n\epsilon I_d}\right) + \operatorname{tr}(Z^T\nabla),$$

which, using the singular value decomposition of Z and ∇, is equal to

$$-\sqrt{\epsilon}\operatorname{tr}\left(\sqrt{4\tau^2 I_d - nZ^T Z}\right) \text{ if } \|Z\|_{op} \leq 2\tau \text{ and } -\infty \text{ otherwise.}$$

The dual problem is therefore

$$\max_{Z\in\mathbb{R}^{n\times d},\|Z\|_{op}\leq\frac{2\tau}{\sqrt{n}}} \frac{1}{n}\|Y\|_2^2 - \frac{1}{n}\left(B + \frac{n}{2}\operatorname{tr}(Z^T W)\right)^T A^{-1}\left(B + \frac{n}{2}\operatorname{tr}(Z^T W)\right) - \sqrt{\epsilon}\operatorname{tr}\left(\sqrt{4\tau^2 I_d - nZ^T Z}\right).$$

The duality gap for fixed β and Z can be computed. At optimum, there is a link

between β and Z, which we obtained when differentiating,

$$\beta'_* = A^{-1}\left(B + \frac{n}{2}\operatorname{tr}(Z_*^T W)\right) \text{ or } \operatorname{tr}(Z_*^T W) = \frac{2}{n}(A\beta'_* - B),$$

where Z_* is the minimiser of the dual problem.

We can obtain an admissible Z for any β by using

$$Z(\beta) = \min\left(1, \frac{2\tau}{\|Z_{opt}(\beta)\|_{op}\sqrt{n}}\right) Z_{opt}(\beta),$$

with $Z_{opt}(\beta)$ any solution to

$$\operatorname{tr}(Z_{opt}(\beta)^T W) = \frac{2}{n}(A\beta - B).$$

We have $Z(\beta'_*) = Z_{opt}(\beta'_*) = Z_*$. We can obtain $Z_{opt} = \sum_{i=1}^{n(d+1)} c_i W_{:,:,i}$ by solving

$$Cc = \frac{2}{n}(A\beta - B),$$

with $C = (\operatorname{tr}(W_{:,:,i}^T W_{:,:,j}))_{i,j} \in \mathbb{R}^{n(d+1)\times n(d+1)}, c \in \mathbb{R}^{n(d+1)}$. □

C.2 Proofs of Section 3

In this section, we present the proofs of Lemma 2 and Lemma 3, which are used to respectively characterise the estimator in a parametric way and the behaviour of the optimisation procedure. We remark that the proof of Lemma 1 is in Appendix B.

Proof of Lemma 2. Lemma 1 states that $\hat{f}_\tau \in S := \operatorname{span}\{k_{x_i}, (\partial_a k)_{x_i} \mid a \in [d], i \in [n]\}$. Therefore, Equation (2.1) has the same solution as

$$\hat{f}_\tau = \arg\min_{f\in S} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \tau\nu\|f\|_{\mathcal{H}}^2 + 2\tau\|\nabla_n f\|_*.$$

We can rewrite these quantities for a function $f \in S$ by expressing f as

$$f = \sum_{i=1}^{n}\frac{1}{n}\theta^{(i)}k_{x_i} + \sum_{i=1}^{n}\sum_{a=1}^{d}\frac{1}{n}\theta^{(i+na)}(\partial_a k)_{x_i},$$

using $\tilde{U}, V, \tilde{W}$ defined in Definition 1, the vector of responses Y, and the coefficients $\theta_0 := (\theta^{(1)}, \dots, \theta^{(n)})^T$, $\theta_1 := (\theta^{(1+n)}, \dots, \theta^{(2n)}, \dots, \theta^{(1+dn)}, \dots, \theta^{(n(d+1))})^T$, and $\theta = (\theta_0^T, \theta_1^T)^T \in \mathbb{R}^{n(d+1)}$.

First, for the data fitting term, $\forall i \in [d], f(x_i) = (K\theta_0)^{(i)} + (Z\theta_1)^{(i)}$, therefore

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = \frac{1}{n}\|Y - K\theta_0 - Z\theta_1\|_2^2 = \frac{1}{n}\|Y - \tilde{U}\theta\|_2^2.$$

For the $\mathcal{H}$-norm, we have

$$\|f\|_{\mathcal{H}}^2 = \frac{\theta_0^T K\theta_0}{n} + \frac{\theta_1^T L\theta_1}{n} + \frac{2\theta_0^T Z\theta_1}{n} = \frac{\theta^T V\theta}{n}.$$

Finally, for the trace norm, $\forall a \in [d], i \in [n]$, $\frac{\partial f}{\partial x^{(a)}}(x_i) = (Z_a\theta_0)^{(i)} + (L_a\theta_1)^{(i)}$. Therefore, since $\nabla_n f = \left(\frac{\partial f}{\partial x^{(a)}}(x_i)/\sqrt{n}\right)_{i,a}$, we have $\nabla_n f = \tilde{W}\theta/\sqrt{n}$, yielding

$$\|\nabla_n f\|_* = \|\tilde{W}\theta\|_*/\sqrt{n}.$$

We then have the following equality for any $f \in S$

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \tau\nu\|f\|_{\mathcal{H}}^2 + 2\tau\|\nabla_n f\|_* = \frac{\|Y - \tilde{U}\theta\|_2^2}{n} + \frac{\tau\nu\theta^T V\theta}{n} + \frac{2\tau}{\sqrt{n}}\|\tilde{W}\theta\|_*,$$

and writing θ_* for the coefficients of $\hat{f}_\tau$ in S, we have

$$\theta_* \in \operatorname*{arg\,min}_{\theta\in\mathbb{R}^{n(d+1)}} \frac{\|Y - \tilde{U}\theta\|_2^2}{n} + \frac{\tau\nu\theta^T V\theta}{n} + \frac{2\tau}{\sqrt{n}}\|\tilde{W}\theta\|_*.$$

Now, with the change of variable $\beta = V^{1/2}\theta$, which is possible since V is positive semi-definite, we get

$$\beta_* = \operatorname*{arg\,min}_{\beta\in\mathbb{R}^{n(d+1)}} \frac{\|Y - U\beta\|_2^2}{n} + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{2\tau}{\sqrt{n}}\|W\beta\|_*,$$

because $\tilde{U}\theta = U\beta$, $\theta^T V\theta = \|\beta\|_2^2$, and $\tilde{W}\theta = W\beta$. The argmin is unique because the quantity is strongly convex in β. □

Proof of Lemma 3. We consider the optimisation problem

$$(\beta'_*, \Lambda_*) = \operatorname*{arg\,min}_{\substack{\beta\in\mathbb{R}^{n(d+1)} \\ \Lambda\in\mathbb{R}^{d\times d},\ \Lambda\succ 0}} \frac{1}{n}\|Y - U\beta\|_2^2 + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{\tau}{\sqrt{n}}\left(\operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right) + \operatorname{tr}(\Lambda + n\epsilon\Lambda^{-1})\right).$$

Since for fixed β, the optimal Λ is $\sqrt{(W\beta)^T(W\beta) + n\epsilon I_d}$, we can change the constraint on Λ to $\Lambda \succeq n\epsilon I_d$ without altering the solution to the optimisation problem. We then define

$$h(\beta, \Lambda) := \frac{1}{n}\|Y - U\beta\|_2^2 + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{\tau}{\sqrt{n}}\operatorname{tr}\left((W\beta)\Lambda^{-1}(W\beta)^T\right),$$

and

$$g(\Lambda) := \operatorname{tr}(\Lambda + n\epsilon\Lambda^{-1}) \text{ if } \Lambda \succeq n\epsilon I_d$$
$$+\infty \text{ otherwise.}$$

The problem is now equivalent to

$$\operatorname*{arg\,min}_{\beta\in\mathbb{R}^{n(d+1)}, \Lambda\in\mathbb{R}^{d\times d}} h(\beta, \Lambda) + g(\Lambda),$$

with

$$H(\beta) := \operatorname*{arg\,min}_{\Lambda\in\mathbb{R}^{d\times d}} h(\beta, \Lambda) = \frac{1}{n}\|Y - U\beta\|_2^2 + \frac{\tau\nu}{n}\|\beta\|_2^2 + \frac{2\tau}{\sqrt{n}}\operatorname{tr}\left(\sqrt{(W\beta)^T(W\beta) + n\epsilon I_d}\right),$$

which fits the setting described in Beck [2015]. We now verify the conditions to apply Theorem 3.3 from Beck [2015].

1. Condition [A] is verified because g is a closed and proper convex function, differentiable over its domain.
2. Condition [B] is verified because h is continuously differentiable over $\mathbb{R}^{n(d+1)} \times \operatorname{dom} g$.
3. Condition [C] is verified with (note the use of the operator norm $\|\cdot\|_{\text{op}}$ and the infinity norm $\|\cdot\|_\infty$)

$$L := \sup_{\Lambda \in \mathbb{R}^{d\times d}, \Lambda \succeq n\epsilon I_d} \left\| \frac{1}{n} \left(U^T U + \tau\nu I_{n(d+1)} + \sqrt{n}\tau(W\Lambda^{-1}W^T) \right) \right\|_{\text{op}}.$$

   L is finite because a rough bound on L is

$$\left\| \frac{1}{n} \left(U^T U + \tau\nu I_{n(d+1)} \right) \right\|_{\text{op}} + d^2 n^{3/2} \tau \max_{m\in[n], l\in[d], k\in[d]} \|W_{m,l,:} W_{m,l,:}^T\|_{op} \|\Lambda^{-1}\|_\infty,$$

   and $\|\Lambda^{-1}\|_\infty \leq \frac{d}{n\epsilon}$.
4. Condition [D] is not needed.
5. Condition [E] is verified because H is strongly convex, so β'_* exists, and the minimiser Λ_* is uniquely defined (see Lemma 4). For fixed β or fixed Λ within the admissible domain, the optimisation problem in the other variable has a minimiser (see Lemma 4).

We now define the level set

$$\{(\beta, \Lambda) \in \mathbb{R}^{n(d+1)} \times \mathbb{R}^{d\times d} \mid \Lambda \succeq n\epsilon I_d, h(\beta, \Lambda) + g(\Lambda) \leq h(\beta_0, \Lambda_0) + g(\Lambda_0)\},$$

which is compact because it is closed and bounded (since h is coercive). We also denote the squared diameter

$$R^2 = \max_{\beta,\Lambda} \left(\|\beta - \beta'_*\|_2^2 + \|\Lambda - \Lambda_*\|_2^2 \mid h(\beta, \Lambda) + g(\Lambda) \leq h(\beta_0, \Lambda_0) + g(\Lambda_0) \right).$$

We can now apply Theorem 3.3 from Beck [2015]. Let $(\beta_t, \Lambda_t)_{t\geq 0}$ be the sequence generated by Algorithm 1, then for all $t \in \mathbb{N}^*$,

$$h(\beta_t, \Lambda_t) + g(\Lambda_t) \leq \frac{3\max\left(h(\beta_0, \Lambda_0) + g(\Lambda_0) - h(\beta'_*, \Lambda_*) - g(\Lambda_*), LR^2\right)}{t}.$$

We then have that for all β,

$$H(\beta) = h(\beta, \Lambda(\beta)) + g(\Lambda(\beta)),$$

where $\Lambda(\beta)$ is defined in Lemma 4 and is always in $\operatorname{dom} g$. We also always have $\Lambda_t = \Lambda(\beta_t)$ by construction. Therefore, we obtain the inequality

$$H(\beta_t) - H(\beta'_*) \leq \frac{3\max\left(H(\beta_0) - H(\beta'_*), LR^2\right)}{t} = \frac{C}{t},$$

where

$$C := 3\max\left(H(\beta_0) - H(\beta'_*), LR^2\right). \tag{2.8}$$

Using the strong convexity of H, we get

$$\frac{\tau\nu}{2n}\|\beta_t - \beta'_*\|_2^2 \leq H(\beta_t) - H(\beta'_*) \leq \frac{C}{t}.$$

Transposing this inequality to the RKHS $\mathcal{H}$, since $H(\beta_t) = \hat{\mathcal{R}}'_\tau(f_t)$ and $H(\beta'_*) = \hat{\mathcal{R}}'_\tau(\hat{f}'_\tau)$ by construction, we get

$$\hat{\mathcal{R}}'_\tau(f_t) - \hat{\mathcal{R}}'_\tau(\hat{f}'_\tau) \leq \frac{C}{t}.$$

Now, using the strong convexity of $\hat{\mathcal{R}}'_\tau$ with a constant at least $2\tau\nu$, we get the inequality

$$\frac{\tau\nu}{4}\|f_t - \hat{f}'_\tau\|_{\mathcal{H}}^2 \leq \frac{\hat{\mathcal{R}}'_\tau(f_t)}{2} + \frac{\hat{\mathcal{R}}'_\tau(\hat{f}'_\tau)}{2} - \hat{\mathcal{R}}'_\tau\left(\frac{\hat{f}'_\tau}{2} + \frac{f_t}{2}\right) \leq \frac{\hat{\mathcal{R}}'_\tau(f_t) - \hat{\mathcal{R}}'_\tau(\hat{f}'_\tau)}{2},$$

yielding

$$\|f_t - \hat{f}'_\tau\|_{\mathcal{H}} \leq \sqrt{\frac{2C}{\tau\nu t}}.$$

□

D Consistency of $\hat{f}_\tau$ from Section 4.1

Section 4.1 deals with the consistent estimation of $\inf_{f\in\mathcal{H}} \mathcal{R}(f)$. We also consider the consistency of the estimation of $f^* = \arg\min_{f\in\mathcal{H}} \mathcal{R}(f)$ in $\mathcal{H}$-norm, as long as $f^* \in \mathcal{H}$, as stated in Lemma 7 below.

Lemma 6 (Consistency of Trace Norm Penalty)**.** *Let $r \in \mathbb{R}^+$. Then under Assumption 3.2, $\forall \eta \in (0, 1]$,*

$$\mathbb{P}\left(\sup_{\|f\|_{\mathcal{H}}\leq r} |\|\nabla_n f\|_* - \|\nabla f\|_*| \geq \frac{\sqrt{2\sqrt{2}}}{n^{1/4}} d^{5/4} r K_2 \sqrt{\log\frac{2d}{\eta}}\right) < \eta.$$

Proof of Lemma 6. For any $f \in \mathcal{H}$,

$$\begin{aligned}
|\|\nabla_n f\|_* - \|\nabla f\|_*| &= \left|\operatorname{tr}\left(\sqrt{\operatorname{cov}(\nabla_n f)} - \sqrt{\operatorname{cov}(\nabla f)}\right)\right| \\
&\leq \sqrt{d}\|\sqrt{\operatorname{cov}(\nabla_n f)} - \sqrt{\operatorname{cov}(\nabla f)}\|_F \\
&\leq \sqrt{d}\sqrt{\|\operatorname{cov}(\nabla_n f) - \operatorname{cov}(\nabla f)\|_*} \\
&\quad \text{(using the Powers–Størmer inequality [Powers and Størmer, 1970])} \\
&\leq \sqrt{d}\sqrt{\sqrt{d}\|\operatorname{cov}(\nabla_n f) - \operatorname{cov}(\nabla f)\|_F} \\
&\leq d^{3/4}\left(\sum_{a,b=1}^{d}\left(\sum_{i=1}^{n}\frac{\partial f}{\partial x^{(a)}}(x_i)\frac{\partial f}{\partial x^{(b)}}(x_i) - \mathbb{E}_{\rho_X}\frac{\partial f}{\partial x^{(a)}}(x)\frac{\partial f}{\partial x^{(b)}}(x)\right)^2\right)^{1/4}.
\end{aligned}$$

We then take the supremum over functions such that $\|f\|_{\mathcal{H}} \leq r$ and apply standard

concentration inequalities [Pinelis and Sakhanenko, 1986]. For each $a, b \in [d]$,

$$\sup_{\|f\|_{\mathcal{H}} \leq r} \left| \sum_{i=1}^{n} \frac{\partial f}{\partial x_a}(x_i) \frac{\partial f}{\partial x_b}(x_i) - \mathbb{E}_{\rho_X} \frac{\partial f}{\partial x_a}(x) \frac{\partial f}{\partial x_b}(x) \right| \leq \frac{2\sqrt{2}}{\sqrt{n}} r^2 K_2^2 \log \frac{2}{\eta}$$

with probability at least $1 - \eta$, $\eta \in (0, 1]$. Applying this d^2 times yields

$$\sup_{\|f\|_{\mathcal{H}} \leq r} |\|\nabla_n f\|_* - \|\nabla f\|_*| \leq d^{5/4} \sqrt{\frac{2\sqrt{2}}{\sqrt{n}} r^2 K_2^2 \log \frac{2d}{\eta}}$$

with probability at least $1 - \eta$. Therefore, $\forall \eta \in (0, 1]$,

$$\mathbb{P}\left(\sup_{\|f\|_{\mathcal{H}} \leq r} |\|\nabla_n f\|_* - \|\nabla f\|_*| \geq \frac{\sqrt{2\sqrt{2}}}{n^{1/4}} d^{5/4} r K_2 \sqrt{\log \frac{2d}{\eta}} \right) < \eta.$$

□

D.1 Proof of Theorem 4

Using Lemma 6, we can now prove the result from the main text on the consistency of the expected risk of the estimator.

Proof of Theorem 4. We have

$$\mathcal{R}(\hat{f}_\tau) - \inf_{f \in \mathcal{H}} \mathcal{R}(f) \leq \left| \mathcal{R}(\hat{f}_\tau) - \mathcal{R}_\tau(f_\tau) \right| + \mathcal{R}_\tau(f_\tau) - \inf_{f \in \mathcal{H}} \mathcal{R}(f),$$

where $\mathcal{R}_\tau(f) = \mathcal{R}(f) + \tau \left(\nu \|f\|_{\mathcal{H}}^2 + 2\|\nabla f\|_* \right)$ and $f_\tau = \arg\min_{f \in \mathcal{H}} \mathcal{R}_\tau(f)$.

First, we note that

$$\tau\nu \|\hat{f}_\tau\|_{\mathcal{H}}^2 \leq \hat{\mathcal{R}}_\tau \left(\hat{f}_\tau \right) \leq \hat{\mathcal{R}}_\tau (0) = \|Y\|^2 / n \leq M^2,$$

and therefore $\|\hat{f}_\tau\|_{\mathcal{H}} \leq \frac{M}{\sqrt{\tau\nu}}$. Similarly, we get $\|f_\tau\|_{\mathcal{H}} \leq \sqrt{\mathbb{E}\left(y^2\right) / \tau\nu} \leq \frac{M}{\sqrt{\tau\nu}}$.

For the first term,

$$\begin{aligned}
\left| \mathcal{R}(\hat{f}_\tau) - \mathcal{R}_\tau(f_\tau) \right| &\leq \left| \mathcal{R}(\hat{f}_\tau) - \hat{\mathcal{R}}(\hat{f}_\tau) \right| + \left| \hat{\mathcal{R}}(\hat{f}_\tau) - \mathcal{R}_\tau(f_\tau) \right| \\
&\leq \left| \mathcal{R}(\hat{f}_\tau) - \hat{\mathcal{R}}(\hat{f}_\tau) \right| + \left| \hat{\mathcal{R}}_\tau(\hat{f}_\tau) - \mathcal{R}_\tau(f_\tau) \right| \\
&\leq \left| \mathcal{R}(\hat{f}_\tau) - \hat{\mathcal{R}}(\hat{f}_\tau) \right| + \left| \hat{\mathcal{R}}_\tau(f_\tau) - \mathcal{R}_\tau(f_\tau) \right| \\
&\leq \left| \mathcal{R}(\hat{f}_\tau) - \hat{\mathcal{R}}(\hat{f}_\tau) \right| + \left| \hat{\mathcal{R}}(f_\tau) - \mathcal{R}(f_\tau) \right| + \tau \left| \|\nabla_n f_\tau\|_* - \|\nabla f_\tau\|_* \right| \\
&\leq 2 \sup_{\|f\|_{\mathcal{H}} \leq M/\sqrt{\tau\nu}} \left| \mathcal{R}(f) - \hat{\mathcal{R}}(f) \right| + \tau \sup_{\|f\|_{\mathcal{H}} \leq M/\sqrt{\tau\nu}} \left| \|\nabla_n f\|_* - \|\nabla f\|_* \right|.
\end{aligned}$$

Using Rosasco et al. [2013, Lemma 19] (with $\eta = 3\eta'/(3 + d)$) and Lemma 6 (with

$\eta = d\eta'/(3+d)$), we have

$$\left|\mathcal{R}(\hat{f}_\tau) - \mathcal{R}_\tau(f_\tau)\right| \leq \frac{4\sqrt{2}}{\sqrt{n}} M^2 \left(\frac{K_1^2}{\tau\nu} + \frac{2K_1}{\sqrt{\tau\nu}} + 1\right) \log\frac{6+2d}{\eta'} + \tau \frac{\sqrt{2\sqrt{2}}}{n^{1/4}} d^{5/4} \frac{M}{\sqrt{\tau\nu}} K_2 \sqrt{\log\frac{6+2d}{\eta'}}$$

with probability at least $1-\eta'$.

We can further bound this term by the simpler quantity

$$\left|\mathcal{R}(\hat{f}_\tau) - \mathcal{R}_\tau(f_\tau)\right| \leq \left(C_1 \frac{M^2}{\sqrt{n}} \left(\frac{K_1^2}{\tau\nu} + \frac{2K_1}{\sqrt{\tau\nu}} + 1\right) + C_2 \frac{\sqrt{\tau} d^{5/4} M K_2}{n^{1/4}\sqrt{\nu}}\right) \log\frac{6+2d}{\eta'}.$$

Now for the second term, if Assumption 4 holds, we have that

$$\mathcal{R}_\tau(f_\tau) - \inf_{f\in\mathcal{H}} \mathcal{R}(f) = \mathcal{R}_\tau(f_\tau) - \mathcal{R}_\tau(f^*) + \tau(\nu\|f^*\|_{\mathcal{H}}^2 + 2\|\nabla f^*\|_*) \leq \tau(\nu\|f^*\|_{\mathcal{H}}^2 + 2\|\nabla f^*\|_*).$$

Therefore, with probability at least $1-\eta'$,

$$\mathcal{R}(\hat{f}_\tau) - \inf_{f\in\mathcal{H}} \mathcal{R}(f) \leq \left(C_1 \frac{M^2}{\sqrt{n}} \left(\frac{K_1^2}{\tau\nu} + \frac{2K_1}{\sqrt{\tau\nu}} + 1\right) + C_2 \frac{\sqrt{\tau} d^{5/4} M K_2}{n^{1/4}\sqrt{\nu}}\right) \log\frac{6+2d}{\eta'} + \tau\left(\nu\|f^*\|_{\mathcal{H}}^2 + 2\|\nabla f^*\|_*\right).$$

Thus, if $\tau_n \to 0$ and $(\tau_n\sqrt{n})^{-1} \to 0$, then $\mathcal{R}(\hat{f}_{\tau_n}) - \inf_{f\in\mathcal{H}} \mathcal{R}(f)$ tends to 0 in probability.

If Assumption 4 does not hold, we still have that $\mathcal{R}_{\tau_n}(\hat{f}_{\tau_n}) - \inf_{f\in\mathcal{H}} \mathcal{R}(f) \to 0$ if $\tau_n \to 0$ as n goes to infinity, which is a standard result in regularisation theory [see Dontchev and Zolezzi, 2006, Rosasco et al., 2013, Proposition 20]. The condition $(\tau_n\sqrt{n})^{-1} \to 0$ still ensures that the first term converges to 0 in probability, yielding the final result. □

D.2 Consistency in $\mathcal{H}$-norm

For the results on the estimation of the subspace that follow, we need the estimation of f^* by $\hat{f}_\tau$ to be consistent in the $\mathcal{H}$-norm, which we prove in the lemma below.

Lemma 7 (Consistency in $\mathcal{H}$-Norm)**.** *Under Assumption 3 and Assumption 4, for any positive sequence $(\tau_n)_{n\in\mathbb{N}}$ such that $\tau_n \to_{n\to\infty} 0$ and $(\sqrt{n}\tau_n^2)^{-1} \to_{n\to\infty} 0$,*

$$\|\hat{f}_{\tau_n} - f^*\|_{\mathcal{H}} \xrightarrow{P} 0.$$

Proof of Lemma 7. The proof follows almost directly from that of Rosasco et al. [2013, Theorem 9]. The primary difference is our penalty term, but Lemma 6 mirrors Theorem 7 from Rosasco et al. [2013], with the same bound except that d is replaced by $d^{5/4}$ in our case. Consequently, all of the equations can be modified with this minor adjustment, yielding the desired result. □

E Estimation of Feature Space P in Section 4.2

Section 4.2 focuses on recovering the lower-dimensional subspace generated by the columns of P. Since P is always estimated through the eigenvectors of $\mathrm{cov}(\nabla_n \hat{f}_\tau)$, we first need

consistency results for the eigenvectors of $\mathrm{cov}(\nabla_n \hat{f}_\tau)$ in their estimation of $\mathrm{cov}(\nabla f^*)$ before we can establish results of the main text.

E.1 Estimation of Eigenvectors of $\mathrm{cov}(\nabla f^*)$

We begin by showing that the matrix $\mathrm{cov}(\nabla f^*)$ is well-estimated by $\mathrm{cov}(\nabla_n \hat{f}_{\tau_n})$ asymptotically under certain conditions on the sequence τ_n.

Lemma 8 (Consistency of Covariance of Gradients)**.** *Under Assumptions 3 and 4,*

$$\left\|\mathrm{cov}\left(\nabla_n \hat{f}_{\tau_n}\right) - \mathrm{cov}\left(\nabla f^*\right)\right\|_F \xrightarrow{P} 0$$

for any positive sequence $(\tau_n)_{n\in\mathbb{N}}$ *where* $\tau_n \to 0$ *and* $(\sqrt{n}\tau_n^2)^{-1} \to 0$ *as* $n \to \infty$.

Proof of Lemma 8. We can decompose the difference into two parts

$$\begin{aligned}\left\|\mathrm{cov}\left(\nabla_n \hat{f}_{\tau_n}\right) - \mathrm{cov}\left(\nabla f^*\right)\right\|_F \leq{}& \left\|\mathrm{cov}\left(\nabla_n \hat{f}_{\tau_n}\right) - \mathrm{cov}\left(\nabla \hat{f}_{\tau_n}\right)\right\|_F \\ &+ \left\|\mathrm{cov}\left(\nabla \hat{f}_{\tau_n}\right) - \mathrm{cov}\left(\nabla f^*\right)\right\|_F.\end{aligned}$$

For the first part, since $\|\hat{f}_{\tau_n}\|_{\mathcal{H}} \leq \frac{M}{\sqrt{\tau\nu}}$, using the proof of Lemma 6, if $(\tau_n\sqrt{n})^{-1} \to 0$, we get

$$\left\|\mathrm{cov}\left(\nabla_n \hat{f}_{\tau_n}\right) - \mathrm{cov}\left(\nabla \hat{f}_{\tau_n}\right)\right\|_F \xrightarrow{P} 0.$$

For the second part, for any $a, b \in [d]$,

$$\begin{aligned}\left(\mathrm{cov}\left(\nabla \hat{f}_{\tau_n}\right) - \mathrm{cov}\left(\nabla f^*\right)\right)_{a,b} &\leq \mathbb{E}_\rho \left| \frac{\partial \hat{f}_{\tau_n}}{\partial x^{(a)}}(x) \frac{\partial \hat{f}_{\tau_n}}{\partial x^{(b)}}(x) - \frac{\partial f^*}{\partial x^{(a)}}(x) \frac{\partial f^*}{\partial x^{(b)}}(x) \right| \\ &\leq \mathbb{E}_\rho \left(\left| \left(\frac{\partial \hat{f}_{\tau_n}}{\partial x^{(a)}}(x) - \frac{\partial f^*}{\partial x^{(a)}}(x) \right) \frac{\partial \hat{f}_{\tau_n}}{\partial x^{(b)}}(x) \right| \right. \\ &\qquad \left. + \left| \left(\frac{\partial \hat{f}_{\tau_n}}{\partial x^{(b)}}(x) - \frac{\partial f^*}{\partial x^{(b)}}(x) \right) \frac{\partial f^*}{\partial x^{(a)}}(x) \right| \right) \\ &\leq \|\hat{f}_{\tau_n} - f^*\|_{\mathcal{H}} K_2^2 \left(\|\hat{f}_{\tau_n}\|_{\mathcal{H}} + \|f^*\|_{\mathcal{H}} \right) \\ &\leq \|\hat{f}_{\tau_n} - f^*\|_{\mathcal{H}} K_2^2 \left(\|\hat{f}_{\tau_n} - f^*\|_{\mathcal{H}} + 2\|f^*\|_{\mathcal{H}} \right).\end{aligned}$$

We then have

$$\begin{aligned}\left\|\mathrm{cov}\left(\nabla \hat{f}_{\tau_n}\right) - \mathrm{cov}\left(\nabla f^*\right)\right\|_F &= \sqrt{\sum_{a,b=1}^{d} \left(\mathbb{E}_\rho \frac{\partial \hat{f}_{\tau_n}}{\partial x_a}(x) \frac{\partial \hat{f}_{\tau_n}}{\partial x_b}(x) - \mathbb{E}_\rho \frac{\partial f^*}{\partial x_a}(x) \frac{\partial f^*}{\partial x_b}(x) \right)^2} \\ &\leq d\|\hat{f}_{\tau_n} - f^*\|_{\mathcal{H}} K_2^2 \left(\|\hat{f}_{\tau_n} - f^*\|_{\mathcal{H}} + 2\|f^*\|_{\mathcal{H}} \right).\end{aligned}$$

From Lemma 7, for any positive sequence $(\tau_n)_{n\in\mathbb{N}}$ such that $\tau_n \to 0$ and $(\sqrt{n}\tau_n^2)^{-1} \to 0$ as $n \to \infty$, we have $\|\hat{f}_{\tau_n} - f^*\|_{\mathcal{H}} \xrightarrow{P} 0$, yielding the desired result. □

We note that the proof of the above result requires only that $\hat{f}_\tau$ is consistent in the $H^1(\rho_X)$ norm, rather than in the $\mathcal{H}$ norm. Convergence in the $\mathcal{H}$ norm is a much stronger condition and directly implies convergence in the $H^1(\rho_X)$ norm.

Using matrix perturbation theory, the previous lemma enables us to achieve consistency in estimating the eigenvectors of $\operatorname{cov}(\nabla f^*)$.

Lemma 9 (Consistency of Eigenvectors)**.** *Let V_0 be the first s eigenvectors of $\operatorname{cov}(\nabla f^*)$ and V_1 be the remaining $d-s$ eigenvectors (in descending order of eigenvalue). Similarly, let $\hat{V}_0$ be the first s eigenvectors of $\operatorname{cov}\left(\nabla_n \hat{f}_\tau\right)$ and $\hat{V}_1$ be the remaining $d-s$ eigenvectors.*

Under Assumptions 1, 2, 3, and 4, with Δ as the minimal eigenvalue of $\operatorname{cov}(\nabla f^*)$, we have

$$\max\left(\|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F, \|V_1V_1^T - \hat{V}_1\hat{V}_1^T\|_F\right) \leq \sqrt{2}\frac{\left\|\operatorname{cov}\left(\nabla_n \hat{f}_\tau\right) - \operatorname{cov}\left(\nabla f^*\right)\right\|_F}{\Delta}.$$

Hence, for any positive sequence $(\tau_n)_{n\in\mathbb{N}}$ such that $\tau_n \to 0$ and $(\sqrt{n}\tau_n^2)^{-1} \to 0$ as $n \to \infty$, we have

$$V_0V_0^T \xrightarrow{P} \hat{V}_0\hat{V}_0^T \quad \text{and} \quad V_1V_1^T \xrightarrow{P} \hat{V}_1\hat{V}_1^T.$$

Proof of Lemma 9. Under Assumptions 1 and 2, V_0 are the eigenvectors corresponding to the non-zero eigenvalues, and V_1 are those corresponding to the zero eigenvalues. Let $\Delta = \lambda_s$, where $(\lambda_1, \ldots, \lambda_d)$ are the eigenvalues of $\operatorname{cov}(\nabla f^*)$ in descending order. Thus, Δ represents the eigengap between the non-zero and zero eigenvalues of $\operatorname{cov}(\nabla f^*)$.

From matrix perturbation theory [Stewart and Sun, 1990, Theorem 3.4], we know that

$$\|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F = \sqrt{2}\|V_0V_0^T(I_d - \hat{V}_0\hat{V}_0^T)\|_F \leq \sqrt{2}\frac{\left\|\operatorname{cov}\left(\nabla_n \hat{f}_\tau\right) - \operatorname{cov}\left(\nabla f^*\right)\right\|_F}{\Delta},$$

which also implies

$$\|V_1V_1^T - \hat{V}_1\hat{V}_1^T\|_F \leq \sqrt{2}\frac{\left\|\operatorname{cov}\left(\nabla_n \hat{f}_\tau\right) - \operatorname{cov}\left(\nabla f^*\right)\right\|_F}{\Delta},$$

since $V_0V_0^T = I_d - V_1V_1^T$ and $\hat{V}_0\hat{V}_0^T = I_d - \hat{V}_1\hat{V}_1^T$.

Using Lemma 8, we know that for any positive sequence $(\tau_n)_{n\in\mathbb{N}}$ such that $\tau_n \to 0$ and $(\sqrt{n}\tau_n^2)^{-1} \to 0$ as $n \to \infty$,

$$\left\|\operatorname{cov}\left(\nabla_n \hat{f}_{\tau_n}\right) - \operatorname{cov}\left(\nabla f^*\right)\right\|_F \xrightarrow{P} 0.$$

Therefore,

$$\|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F \xrightarrow{P} 0 \quad \text{and} \quad \|V_1V_1^T - \hat{V}_1\hat{V}_1^T\|_F \xrightarrow{P} 0,$$

yielding the desired result. □

E.2 Limit of the Errors when the Subspace Estimator Dimension is Fixed

In this section, we fix the number s' of leading eigenvectors of $\operatorname{cov}(\nabla_n \hat{f}_\tau)$ to form the estimator, denoted as $\hat{P}_{s'}$. This approach aims to illustrate why accurately estimating s is essential for achieving consistency in the Frobenius norm. Additionally, it provides results for a data-independent choice of dimension, which might be applicable when a reasonable upper bound on s or its exact value is known and used. We present the following lemma on the asymptotic behaviour of the Frobenius error.

Lemma 10 (Frobenius Error with Fixed Dimension)**.** *Under Assumptions 1, 2, 3, and 4, for any positive sequence* $(\tau_n)_{n\in\mathbb{N}}$ *such that* $\tau_n \to 0$ *and* $(\sqrt{n}\tau_n^2)^{-1} \to 0$ *as* $n \to \infty$*, we have*

$$\|\Pi_P - \Pi_{\hat{P}_{s'}}\|_F^2 \xrightarrow{P} |s - s'|.$$

We observe that the error converges to the difference between the true dimension of the subspace and the dimension of the estimated subspace. Therefore, s' must be exactly equal to s for the error to be asymptotically null.

Proof of Lemma 10. Let V_0 be the first s eigenvectors of $\mathrm{cov}(\nabla f^*)$ and V_1 be the other $d-s$ ones (in descending order of eigenvalue). Similarly, let $\hat{V}_0$ be the first s eigenvectors of $\mathrm{cov}(\nabla_n \hat{f}_\tau)$ and $\hat{V}_1$ be the remaining $d-s$ ones.

We recall that $\Pi_P = V_0 V_0^T$ and $\Pi_{\hat{P}} = \hat{P}_{s'}\hat{P}_{s'}^T$, where $\hat{P}_{s'}$ are the first s' eigenvectors of $\mathrm{cov}(\nabla_n \hat{f}_\tau)$. We denote by $\hat{V}_1^{s+1:s'}$ the first $s'-s$ columns of $\hat{V}_1$ and $\hat{V}_0^{s'+1:s}$ the last $s-s'$ columns of $\hat{V}_0$ when those quantities make sense.

Now if $s' \geq s$,

$$\begin{aligned}
\|\Pi_P - \Pi_{\hat{P}_{s'}}\|_F^2 &= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T - \hat{V}_1^{s+1:s'}(\hat{V}_1^{s+1:s'})^T\|_F^2 \\
&= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F^2 + \|\hat{V}_1^{s+1:s'}(\hat{V}_1^{s+1:s'})^T\|_F^2 \\
&\quad - 2\langle V_0V_0^T - \hat{V}_0\hat{V}_0^T, \hat{V}_1^{s+1:s'}(\hat{V}_1^{s+1:s'})^T\rangle \\
&= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F^2 + s' - s \\
&\quad - 2\langle V_0V_0^T, \hat{V}_1^{s+1:s'}(\hat{V}_1^{s+1:s'})^T\rangle \\
&= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F^2 + s' - s \\
&\quad - 2\langle \Pi_{V_0}, \Pi_{\hat{V}_1^{s+1:s'}}\rangle.
\end{aligned}$$

By Lemma 9, the first and third terms of the above inequality tend to 0 in probability, yielding $s'-s$ as the limit.

If $s' < s$,

$$\begin{aligned}
\|\Pi_P - \Pi_{\hat{P}_{s'}}\|_F^2 &= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T + \hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T\|_F^2 \\
&= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F^2 + \|\hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T\|_F^2 \\
&\quad + 2\langle V_0V_0^T - \hat{V}_0\hat{V}_0^T, \hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T\rangle \\
&= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F^2 + s - s' \\
&\quad - 2\langle V_1V_1^T, \hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T\rangle \\
&= \|V_0V_0^T - \hat{V}_0\hat{V}_0^T\|_F^2 + s - s' \\
&\quad - 2\langle \Pi_{V_1}, \Pi_{\hat{V}_0^{s+1:s'}}\rangle.
\end{aligned}$$

By Lemma 9, the first and third terms of the above inequality tend to 0 in probability, yielding $s-s'$ as the limit. We then have that $\|\Pi_P - \Pi_{\hat{P}_{s'}}\|_F^2$ converges to $|s'-s|$ in probability. □

As discussed in the main text, we are also concerned with the safe filter error defined in Equation (2.6). We now present the following result regarding its asymptotic behaviour when the dimension of the estiamted subspace is fixed.

Lemma 11 (Safe Filter Error with Fixed Dimension)**.** *Under Assumptions 1, 2, 3, and 4, for any positive sequence* $(\tau_n)_{n\in\mathbb{N}}$ *such that* $\tau_n \to 0$ *and* $(\sqrt{n}\tau_n^2)^{-1}0$ *as* $n \to \infty$*, we have*

- *if* $s' \geq s$, *then*
$$\|\Pi_P(I_d - \Pi_{\hat{P}_{s'}})\|_F^2 \xrightarrow{P} 0,$$
- *if* $s' < s$, *then*
$$\|\Pi_P(I_d - \Pi_{\hat{P}_{s'}})\|_F^2 \xrightarrow{P} s - s'.$$

We note that it is only necessary for s' to be larger than s to achieve an asymptotically safe filter. Therefore, any choice of s' that exceeds s but remains small enough to effectively reduce the dimensionality of the problem is appropriate.

Here's an improved version of the proof with better phrasing and clarity:

Proof of Lemma 11. Let V_0 be the first s eigenvectors of $\mathrm{cov}(\nabla f^*)$ and V_1 be the remaining $d - s$ eigenvectors (in descending order of eigenvalue). Similarly, let $\hat{V}_0$ be the first s eigenvectors of $\mathrm{cov}(\nabla_n \hat{f}_\tau)$ and $\hat{V}_1$ be the remaining $d - s$ eigenvectors.

We have
$$\|\Pi_P(I_d - \Pi_{\hat{P}_{s'}})\|_F^2 = s - \mathrm{tr}(\Pi_P \Pi_{\hat{P}_{s'}})$$
where $\Pi_P = V_0 V_0^T$ and $\Pi_{\hat{P}_{s'}} = \hat{P}_{s'} \hat{P}_{s'}^T$.

Let $\hat{V}_1^{s+1:s'}$ denote the first $s' - s$ columns of $\hat{V}_1$ and $\hat{V}_0^{s'+1:s}$ denote the last $s - s'$ columns of $\hat{V}_0$, when these quantities make sense.

If $s' \geq s$:
$$\begin{aligned}\|\Pi_P(I_d - \Pi_{\hat{P}_{s'}})\|_F^2 &= s - \mathrm{tr}(V_0 V_0^T(\hat{V}_0 \hat{V}_0^T + \hat{V}_1^{s+1:s'}(\hat{V}_1^{s+1:s'})^T)) \\ &= s - \mathrm{tr}(V_0 V_0^T \hat{V}_0 \hat{V}_0^T) - \mathrm{tr}(V_0 V_0^T \hat{V}_1^{s+1:s'}(\hat{V}_1^{s+1:s'})^T).\end{aligned}$$

By Lemma 9, the second term tends to s and the third term to 0 in probability, yielding the desired result.

If $s' < s$:
$$\begin{aligned}\|\Pi_P(I_d - \Pi_{\hat{P}_{s'}})\|_F^2 &= s - \mathrm{tr}(V_0 V_0^T(\hat{V}_0 \hat{V}_0^T - \hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T)) \\ &= s - \mathrm{tr}(V_0 V_0^T \hat{V}_0 \hat{V}_0^T) + \mathrm{tr}((I_d - V_1 V_1^T)\hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T) \\ &= s - \mathrm{tr}(V_0 V_0^T \hat{V}_0 \hat{V}_0^T) + \mathrm{tr}(\hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T) - \mathrm{tr}(V_1 V_1^T \hat{V}_0^{s'+1:s}(\hat{V}_0^{s'+1:s})^T).\end{aligned}$$

By Lemma 9, the second term tends to 1, the third term to $s - s'$, and the fourth term to 0 in probability, yielding the desired result. □

E.3 Proof of Theorem 5

We now have all the necessary components to prove the main result on the consistency of the subspace estimator. When considering the estimated dimension of the estimated subspace, the result crucially depends on the lower semi-continuity of the rank.

Proof of Theorem 5. For the first result, we apply Lemma 10 with $s' = s$.

For the second result, we have
$$\|\Pi_P(I_d - \Pi_{\hat{P}})\|_F^2 = s - \mathrm{tr}(\Pi_P \Pi_{\hat{P}}).$$

We denote by $\hat{V}_1^{s+1:\hat{s}}$ the first $\hat{s} - s$ columns of $\hat{V}_1$ and by $\hat{V}_0^{\hat{s}+1:s}$ the last $s - \hat{s}$ columns of $\hat{V}_0$ when these quantities make sense.

Now, we have

$$\|\Pi_P(I_d - \Pi_{\hat{P}})\|_F^2 = s - \operatorname{tr}(V_0 V_0^T \hat{V}_0 \hat{V}_0^T) + \mathbb{1}_{\hat{s}\geq s} \operatorname{tr}(V_0 V_0^T \hat{V}_1^{s+1:\hat{s}} (\hat{V}_1^{s+1:\hat{s}})^T) + \mathbb{1}_{\hat{s}<s} \left(\operatorname{tr}(\hat{V}_0^{\hat{s}+1:s} (\hat{V}_0^{\hat{s}+1:s})^T) - \operatorname{tr}(V_1 V_1^T \hat{V}_0^{\hat{s}+1:s} (\hat{V}_0^{\hat{s}+1:s})^T) \right).$$

By Lemma 9, we know that

$$\operatorname{tr}(V_0 V_0^T \hat{V}_0 \hat{V}_0^T) \xrightarrow{P} s \text{ and } \operatorname{tr}(V_0 V_0^T \hat{V}_1^{s+1:\hat{s}} (\hat{V}_1^{s+1:\hat{s}})^T) \xrightarrow{P} 0.$$

Additionally, we know that $\left(\operatorname{tr}(\hat{V}_0^{\hat{s}+1:s} (\hat{V}_0^{\hat{s}+1:s})^T) - \operatorname{tr}(V_1 V_1^T \hat{V}_0^{\hat{s}+1:s} (\hat{V}_0^{\hat{s}+1:s})^T) \right)$ is bounded.

Since the rank is lower semi-continuous and $\operatorname{cov}(\nabla_n \hat{f}^\tau) \xrightarrow{P} \operatorname{cov}(\nabla f^*)$, we necessarily have

$$\mathbb{P}\left(\operatorname{rank} \operatorname{cov}(\nabla_n \hat{f}_{\tau_n}) \geq s \right) \xrightarrow{P} 1,$$

which can also be written as $\mathbb{1}_{\hat{s}\geq s} \xrightarrow{P} 1$. Therefore,

$$\|\Pi_P(I_d - \Pi_{\hat{P}})\|_F^2 \xrightarrow{P} 0.$$

□

F Numerical Experiments

In this section we provide technical details of the numerical experiments. Recall that the code is provided at https://github.com/BertilleFollain/KTNGrad/.

In the first experiment presented in Section 5.1, the regularisation parameters were set as follows: $\nu = 10^{-6}$, $\tau = \frac{1}{8n}$, $\mu = 10^{-16}$ (used simply to avoid issues with the Cholesky decomposition, see the code) and $\epsilon = 10^{-8}$. The optimisation process used a convergence threshold $\delta = 10^{-6}$ and a maximum of 10 iterations. The Gaussian kernel was parameterised by σ, chosen as the median of the pairwise Euclidean distances between the training samples.

In the second experiment presented in Section 5.2, we evaluate and compare the performance of several methods using synthetic datasets generated with varying sample sizes and dimensions. The experiment uses a range of values for both sample size n and dimension d. The sample sizes n were set to 10, 20, 50, 75, 100, 125, 150, and 175, while the dimensions d explored were 3, 5, 10, 15, 20, 25, 30, and 35. The fixed values were $d = 10$ when varying n, and $n = 175$ when varying d.

For KTNGrad and KTNGrad, retrained, the regularisation parameters were set as $\nu = 10^{-5}$, $\mu = 10^{-8}$, and $\epsilon = 10^{-8}$, while τ was defined as $\frac{1}{2n}$. The convergence threshold was $\delta = 10^{-3}$, with the optimisation process capped at a maximum of 5 iterations. For KRR, the regularisation parameter λ was set as $\frac{1}{n}$. PyMave and MARS were run using their default settings.

The threshold used to select the dimension of the features, either for the dimension score, or for training MARS in KTNGrad, retrained is actually used on the singular values of $\nabla_n \hat{f}_\tau$, with the threshold above which values are kept set to $\operatorname{tr}(\nabla_n \hat{f}_\tau)/(2d)$.

CHAPTER 3

Group Lasso Penalty on Hermite Polynomials Decomposition

The contents of this chapter are available in the article B. Follain and F. Bach. Nonparametric Linear Feature Learning in Regression Through Regularisation. Electronic Journal of Statistics, 18(2):4075–4118, 2024, while the code is available at `https://github.com/BertilleFollain/RegFeaL`.

Abstract

Representation learning plays a crucial role in automated feature selection, particularly in the context of high-dimensional data, where non-parametric methods often struggle. In this study, we focus on supervised learning scenarios where the pertinent information resides within a lower-dimensional linear subspace of the data, namely the multi-index model. If this subspace were known, it would greatly enhance prediction, computation, and interpretation. To address this challenge, we propose a novel method for joint linear feature learning and non-parametric function estimation, aimed at more effectively leveraging hidden features for learning. Our approach employs empirical risk minimisation, augmented with a penalty on function derivatives, ensuring versatility. Leveraging the orthogonality and rotation invariance properties of Hermite polynomials, we introduce our estimator, named RegFeaL. By using alternative minimisation, we iteratively rotate the data to improve alignment with leading directions. We establish that the expected risk of our method converges in high-probability to the minimal risk under minimal assumptions and with explicit rates. Additionally, we provide empirical results demonstrating the performance of RegFeaL in various experiments.

Contents

1 Introduction

The increasing availability of high-dimensional data has created a demand for effective feature selection methods that can handle complex datasets. Representation learning, which aims to automate the feature selection process, plays a crucial role in extracting meaningful information from such data. However, non-parametric methods often struggle in high-dimensional settings.

A sensible approach is to consider that there are a lower number of unknown relevant linear features, or linear transformations of the original data, that explain the relationship between the response and factors. A popular way to model this is to consider the multi-index model [Xia, 2008], where we assume that the prediction function is the composition of few linear features which form a linear subspace (the effective dimension reduction (e.d.r.) subspace) and a non-parametric function. The multi-index model has been used in practice in many fields, such as ecology [Stenseth et al., 2022] or bio-informatics [Antoniadis et al., 2003]. If the features were known, learning would be much easier due to the lower dimensionality of the problem, and their low number allows for a simpler, more explainable model, as well as a lesser need for computational and storage resources. Although these relevant features are not known a priori, recognising their existence enables the development of methods that incorporate them, potentially resulting in better estimators for prediction.

Related work. A wide range of methods have been proposed to estimate the e.d.r. space in the context of multi-index models. Brillinger [2012] introduced the method of moments, initially designed for Gaussian data and an e.d.r. of dimension one. This method uses specific moments to eliminate the unknown function and focuses solely on the influence of the e.d.r. space. Extensions of this approach for distributions with differentiable log-densities have been provided, resulting in the average derivative estimation (ADE) method [Stoker, 1986].

To incorporate subspaces of any dimension, several methods have been proposed. Slicing methods, such as slice inverse regression (SIR) [Li, 1991], use second-order moments to account for subspaces. Principal Hessian directions (PHD) [Li, 1992] extend the approach to elliptically symmetric data. Combining these techniques, sliced average derivative estimation (SADE) [Babichev and Bach, 2018] offers a comprehensive approach. However, these methods heavily rely on assumptions about the distribution shape and require prior knowledge of the distribution, limiting their applicability.

Iterative improvements have been suggested for both the one-dimensional latent subspace case [Hristache et al., 2001] and the general case [Dalalyan et al., 2008]. Other optimisation-based methods, such as local averaging, aim to minimise an objective function to estimate the subspace [Fukumizu et al., 2009, Xia et al., 2002]. Although these procedures exhibit favourable performance in practice, particularly the MAVE method [Xia et al., 2002], the theoretical guarantees provided by Xia et al. [2002] show exponential dependency in the dimension of the original data. Nonetheless, the recent work by Jing Zeng and Zhang [2024] has made significant contributions to sufficient dimension reduction (SDR) by providing robust theoretical results for high-dimensional data that do not exhibit exponential dependency. However, their method, designed primarily for dimension reduction and variable selection in the specific setting of the square loss, relies on the linearity condition, which holds for example under the assumption that the covariates follow an elliptically contoured distribution.

In our work, we consider regularising the empirical risk by incorporating derivatives, a

technique employed in various contexts. Classical splines, such as Sobolev spaces regularisation [Wahba, 1990], have used derivative-based regularisation. More recently, derivative regularisation has been employed in the context of semi-supervised learning [Cabannes et al., 2021], as well as in linear subspace estimation using SADE [Babichev and Bach, 2018].

Contributions. We propose a novel approach for joint function estimation and effective dimension reduction space estimation in multi-index models.

We employ the empirical risk minimisation framework, compatible with a wide range of loss functions, which is regularised by a penalty on the derivatives of the prediction function. The proposed regularisation enforces dependence on a reduced set of projected dimensions. Our method addresses the discussed limitations of previous methods. Indeed the assumptions on the distribution of the covariates are minimal (typically subgaussianity of the norm), and does not require said distribution to be known a priori. We are also able to provide explicit rates for the high-probability convergence of the expected risk of our estimator to the minimal risk, again with limited assumptions.

To construct our estimator, which we coin RegFeaL, we exploit the advantageous properties of Hermite polynomials, which exhibit orthogonality and rotation invariance. By incorporating alternative minimisation on a variational formulation of the problem, we enable iterative rotation of the data to better align with the leading directions, as well as easy computation of the unknown relevant dimension of the e d.r. space. Furthermore, for the specific case of the variable selection problem, that is, when only a subset of the coordinates of the original data is relevant, we can simplify our proposed penalty term which yields a computationally more efficient algorithm.

While our primary objective is to leverage the existence of a dependency on only a few variables or features, we also offer principled ways to estimate the dimension of the feature space and select the relevant features.

We provide detailed explanations about the efficient computation of our estimator, ensuring its practical usability. Additionally, we present theoretical results that establish the high-probability convergence to the minimal risk of the expected risk of our estimator, with limited assumptions on the loss and data distribution. This allows for a deeper understanding of the performance of the method and the dependency on certain parameters such as the dimension of the original data and the number of samples.

To demonstrate the strengths of our approach, we conduct an extensive set of experiments focusing on training behaviour, dependency on sample size and dimension, and comparison to other methods.

Importantly, our regularisation strategy is applicable to a wide range of problems where empirical risk can be formulated, making it a versatile tool for feature learning and dimensionality reduction tasks, potentially extending beyond statistics to fields such as signal processing and control.

In summary, our contributions encompass the introduction of a novel empirical risk minimisation framework with derivative-based regularisation for prediction and e.d.r. subspace estimation in multi-index models. We provide efficient computational techniques, theoretical insights, and empirical evidence, highlighting the advantages of our proposed method.

Chapter organisation. The chapter is organised as follows: we begin by describing the problem, our penalties, and the use of Hermite polynomials in Section 2. Then, we address the question of effectively computing our estimator RegFeaL in Section 3. In

Section 4, we discuss the convergence of the empirical risk of our estimator. In Section 5, we present numerical studies to illustrate the behaviour of RegFeaL. Finally, in Section 6, we summarise our findings, highlight the contributions of our research, and discuss potential future directions.

Notations. Let $\mathbb{N}$ denote the set of non-negative integers and $\mathbb{N}^*$ the set of positive integers. For $d \in \mathbb{N}$, let $[d] = 1, \ldots, d$. Given $x \in \mathbb{R}^d$ and $a \in [d]$, x_a represents the a-th component of x. Similarly, for $S \subset [d]$, x_S denotes $(x_a)_{a \in S}$. Let $p, d \in \mathbb{N}^*$, and consider a matrix $A \in \mathbb{R}^{p \times d}$. The matrix A_S corresponds to the columns of A extracted using indices from S, while $A_{i,j}$ represents the element of A in the j-th position of row i. The cardinality of a set S is denoted by $|S|$. I_d represents the $d \times d$ identity matrix, and O_d denotes the set of $d \times d$ orthogonal matrices. For any $d \times d$ matrix A, $\mathrm{tr}(A)$ denotes its trace, and $\mathrm{Diag}(A)$ represents the diagonal matrix of size $d \times d$ with the diagonal elements of A. The transpose of a matrix B is denoted by $B^\top$. For an invertible matrix Λ, Λ^{-1} represents its inverse. Given $\eta \in \mathbb{R}^d$, $\mathrm{Diag}(\eta)$ is the diagonal matrix of size $d \times d$ with η as its diagonal. For $r > 0$, $\|\eta\|_r = \left(\sum_{a=1}^d |\eta_a|^r\right)^{1/r}$. For any $\alpha \in \mathbb{N}^d$, $|\alpha| = \sum_{a=1}^d \alpha_a$.

2 Preliminaries

2.1 Problem Description

We consider a standard regression problem, where we have a dataset $(x^{(i)}, y^{(i)})_{i \in [n]}$, $n \in \mathbb{N}^*$consisting of independent and identically distributed (i.i.d.) realisations of a pair of random variables (X, Y) with probability measure ν on $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$. Our objective is to estimate the regression function $f^* := \arg\min_{f \in \mathcal{F}} \mathcal{R}(f)$, where $\mathcal{R}(f) := \mathbb{E}_\nu(\ell(Y, f(X)))$ is the risk, ℓ is a loss function and $\mathcal{F}$ a space of functions from $\mathbb{R}^d$ to $\mathbb{R}$. At this stage, we do not impose any assumptions regarding the choice of loss function or the data distribution.

We consider the multi-index model [Xia, 2008], i.e., a model where the regression function depends on a low-rank linear transformation of the original variables.

Assumption 5 (Feature Learning)**.** *We assume that the regression function f^* can be expressed as the combination of a rank s linear transformation P and a function g^* from $\mathbb{R}^s \to \mathbb{R}$, i.e.,*

$$\exists s \in [d],\ \exists P \in \mathbb{R}^{d \times s},\ P^\top P = I_s,\ \exists g^* : \mathbb{R}^s \to \mathbb{R},\ \forall x \in \mathbb{R}^d,\ f^*(x) = g^*(P^\top x).$$

We do not assume any prior knowledge about the value of s. The model is nonparametric hence it remains broad. Our objective is to simultaneously estimate both f^* and the associated linear transformation P, as well as the dimension s, by means of regularised empirical risk minimisation. Recall the definition of the empirical risk $\widehat{\mathcal{R}}(f) := \frac{1}{n}\sum_{i=1}^n \ell(y^{(i)}, f(x^{(i)}))$. This approach offers versatility, allowing its application to various scenarios. Although our focus lies on the regression setting, we acknowledge the potential of the regularisation-based method for future work in any setting where a risk can be defined.

2.2 Penalising by Derivatives

With these assumptions, it is common to employ derivative-based regularisation techniques [Babichev and Bach, 2018, Rosasco et al., 2013]. Under mild regularity assumptions, if we express f as $f = g(Q^\top \cdot)$ with $Q \in \mathbb{R}^{d \times s}$, then for all $x \in \mathbb{R}^d$, $\nabla f(x) \cdot \nabla f(x)^\top =$

$Q\nabla g(x)\cdot\nabla g(x)^\top Q^\top$, where $\nabla f(x) \in \mathbb{R}^d$ denotes the gradient of f at point x. Consequently, we observe that

$$\int_{\mathcal{X}} \nabla f \nabla f^\top \nu = \left(\int_{\mathcal{X}} \frac{\partial f}{\partial x_a} \frac{\partial f}{\partial x_b} \nu \right)_{a,b\in[d]}$$

has a rank of at most s. This observation motivates us to employ the rank of $\int_{\mathcal{X}} \nabla f \nabla f^\top \nu$ as a penalisation. However, the discontinuous nature of the rank makes this approach challenging for optimisation. To address this, we could penalise instead by $\operatorname{tr}\left(\int_{\mathcal{X}} \nabla f \nabla f^\top \nu\right)$ as a convex relaxation [Recht et al., 2010].

This strategy would extend the work of Rosasco et al. [2013], which focuses on variable selection, a special case of feature learning. It corresponds to the constraint that P from Assumption 5 only contains 0 and 1 (with exactly a single one in each column), resulting in a model where the regression function depends on a limited number of the original variables.

Assumption 6 (Variable Selection)**.** *We assume that f^*, the regression function, actually only depends on s of the d variables, i.e.,*

$$\exists s \in [d],\ \exists S \subset [d],\ |S| = s,\ \exists\ g^* : \mathbb{R}^s \to \mathbb{R},\ \forall x \in \mathbb{R}^d,\ f^*(x) = g^*(x_S).$$

In this variable selection setting, we can remark that it suffices to penalise by a simpler quantity. Specifically, under some mild regularity assumptions on the function f, f does not depend on variable x_a if and only if the partial derivative of f with respect to x_a, denoted by $\frac{\partial f}{\partial x_a}$, is null everywhere on $\mathcal{X}$. Hence, the task is to design a penalty that enforces sparsity in the dependence on different variables.

To address this, we can draw inspiration from the group Lasso [Yuan and Lin, 2006], which extends the Lasso method to enable structured sparsity. The group Lasso encourages groups of related quantities to be selected or excluded together by penalising the sum over each group using an appropriate penalty. For example, the derivatives with regard to a variable x_a at data points $x^{(i)}$ should all be null if the function does not depend on variable x_a. Hence, they constitute a relevant group for group Lasso.

Combining these observations, Rosasco et al. [2013] proposed a strategy using the fact that for all $a \in [d]$, f does not depend on x_a if and only if $\int_{\mathcal{X}} \left(\frac{\partial f}{\partial x_a}(x)\right)^2 \nu = 0$. They introduced penalties on each variable and summed them to obtain the penalty $\sum_{a=1}^d \left(\int_{\mathcal{X}} \left(\frac{\partial f}{\partial x_a}(x)\right)^2 \nu(x)\mathrm{d}x\right)^{1/2}$. However, since these quantities are intractable due to the unknown nature of ν, they use a data-dependent penalty instead

$$\sum_{a=1}^{d} \left(\frac{1}{n} \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_a}(x^{(i)}) \right)^2 \right)^{1/2}.$$

By assuming that f belongs to some regular reproducing kernel Hilbert space (RKHS), the partial derivatives are easily computable, and so is the penalty [Rosasco et al., 2013] [for a good introduction to RKHS, see Aronszajn, 1950]. However, this regularisation by an estimate of the L^2 norms of derivatives in the context of RKHS is not suitable. Functions that depend on a single variable, such as x_1, do not belong to the RKHS, making it an inappropriate space for addressing this type of problem. Additionally, another regularisation by the norm in the RKHS is required, introducing an extra hyperparameter. Moreover, using derivatives only at the data points limits the exploitation of the power of regularity.

We are confronted with two challenges here. First, how can the penalisation scheme be

improved for variable selection? Second, how can it be adapted for feature learning? While our primary goal is the latter, we consider the former as a by-product of our methodology.

To address both challenges, we employ Hermite polynomials [Hermite, 2009], although it is worth noting that various other alternatives could have been considered for the first problem where rotation invariance is not needed.

2.3 Hermite Polynomials for Variable Selection

To facilitate understanding, let us first consider the simpler case of variable selection. We employ multidimensional Hermite polynomials due to their suitability for both variable selection and feature learning. The normalised one-dimensional Hermite polynomials $(h_k(x))_{k\geq 0}$ form an orthonormal polynomial basis for the standard Gaussian measure on $\mathbb{R}$ with density $\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$. The first few polynomials are given by[1]

$$h_0(x) = 1,\ h_1(x) = x,\ h_2(x) = \frac{1}{\sqrt{2}}(x^2-1),\ h_3(x) = \frac{1}{\sqrt{6}}(x^3-3x).$$

These polynomials possess useful properties that allow their recursive computation and characterise their growth and their derivatives[2]

$$h_{n+2}(x) = \frac{x}{\sqrt{n+2}}\cdot h_{n+1}(x) - \sqrt{\frac{n+1}{n+2}}\cdot h_n(x) \tag{3.1}$$

$$h_n'(x) = \sqrt{n}\cdot h_{n-1}(x) \tag{3.2}$$

$$|h_n(x)| \leq \exp(x^2/4). \tag{3.3}$$

Next, we define the multivariate polynomials as follows

$$(H_\alpha)_{\alpha\in\mathbb{N}^d} \text{ where } \forall x\in\mathbb{R}^d,\ H_\alpha(x) = \prod_{a=1}^{d} h_{\alpha_a}(x_a). \tag{3.4}$$

This family forms an orthonormal basis of the space $L^2(q) := \{f : \mathbb{R}^d \to \mathbb{R},\ \int_{\mathbb{R}^d} f^2 q < +\infty\}$ where $q(x) = \frac{1}{(2\pi)^{d/2}}e^{-\|x\|^2/2}$ denotes the standard normal distribution on $\mathbb{R}^d$. We now present a Lemma which justifies the use of the multivariate Hermite polynomials in the variable selection setting.

Lemma 12 (Equivalence for Dependency on Variables)**.** *Let $f \in L^2(q)$ and express it as $f = \sum_{\alpha\in\mathbb{N}^d}\hat{f}(\alpha)H_\alpha$.*[3] *Then for any $b \in [d]$,*

$$f \text{ does not depend on variable } x_b \iff \forall\alpha\in(\mathbb{N}^d)^*,\ \alpha_b \neq 0 \implies \hat{f}(\alpha) = 0.$$

[1] Given the regular "physicist" Hermite polynomials H_k (not to be confused with multivariate polynomials defined in Equation (3.4)), we have $h_k(x) = \frac{1}{\sqrt{2^k k!}}H_k(x/\sqrt{2})$ for any $k\in\mathbb{N}$ and for the "probabilist" Hermite polynomials He_k, we have $h_k(x) = \frac{1}{\sqrt{n!}}He_k(x)$.

[2] The last property can be proved using Hermite functions and Cramer's inequality [Szegő, 1939].

[3] Note that in this chapter $\hat{f}$ corresponds to coefficients in the Hermite polynomials decomposition of f, not to the estimator of f^*.

Proof of Lemma 12. For $x \in \mathbb{R}^d$, we have $h_0(x) = 1$ and

$$f(x) = \underbrace{\hat{f}(0) + \sum_{\alpha \in (\mathbb{N}^d)^*,\ \alpha_b = 0} \hat{f}(\alpha) \prod_{a \in [d] \setminus \{b\}} h_{\alpha_a}(x_a)}_{\text{does not depend on } x_b} + \underbrace{\sum_{\alpha \in (\mathbb{N}^d)^*,\ \alpha_b > 0} \hat{f}(\alpha) \prod_{a \in [d]} h_{\alpha_a}(x_a)}_{\text{depends on } x_b},$$

i.e., f can be decomposed into two additive components, one of which does not depend on x_b. For the component that depends on x_b, it is the sum over $\alpha \in \mathbb{N}^d$ such that α_b is non-zero, yielding the result. □

We observe that when f does not depend on a variable, it corresponds to a specific sparsity pattern in the coefficients $\hat{f}(\alpha)$ with respect to the basis $(H_\alpha)_{\alpha \in \mathbb{N}^d}$. Indeed, if f does not depend on x_b, all coefficients $\hat{f}(\alpha)$ for α in the group $\{\alpha \in (\mathbb{N}^d)^*, \alpha_b > 0\}$ must be null. These groups overlap for different variables, and a similar argument holds for feature learning as we will see in Section 2.4. This specific sparsity pattern motivates the use of a penalty based on group Lasso [Yuan and Lin, 2006], and more specifically overlapping group Lasso [Jenatton et al., 2011].

Hence, the Hermite polynomial basis is well-suited to this variable selection setting, while the space $L^2(q)$ is sufficiently large to describe a wide range of functions. However, it is worth noting that other spaces and well-adapted bases, such as any orthonormal basis of square-integrable functions, could also be used. Moreover, we use the Gaussian measure only to define the basis, and our method can be applied to all distributions.

To define a penalty relevant to variable selection, we examine the derivatives of H_α. Here, we decompose any $f \in \mathcal{F}$ as $f = \sum_{\alpha \in \mathbb{N}^d} \hat{f}(\alpha) H_\alpha$. Let e_a denote the a-th element of the canonical basis of $\mathbb{R}^d$, for $a \in [d]$. Using Equation (3.2), we obtain the following identities

$$\frac{\partial H_\alpha}{\partial x_a} = \sqrt{\alpha_a} H_{\alpha - e_a} \tag{3.5}$$

$$\frac{\partial f}{\partial x_a} = \sum_{\alpha \in (\mathbb{N}^d)^*} \sqrt{\alpha_a} \hat{f}(\alpha) H_{\alpha - e_a} \tag{3.6}$$

$$\int_{\mathbb{R}^d} \left(\frac{\partial f}{\partial x_a} \right)^2 q = \sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_a \hat{f}(\alpha)^2. \tag{3.7}$$

However, we remark that Equation (3.7) corresponds to the expected version of the penalty proposed by Rosasco et al. [2013] (when $\nu = q$), which we deemed not suitable for our problem: indeed, penalising the L^2-norm of derivatives does not impose enough regularity for statistically efficient non-parametric estimation and thus requires extra regularisation, as specified by Rosasco et al. [2013].

We consider instead introducing a sequence $(c_k)_{k>0}$ of non-negative reals, to further regularise and avoid the need for additional regularisation. We consider the space $\mathcal{F}$, spanned by the family composed of H_α for $\alpha = 0$ or $\alpha \in (\mathbb{N}^d)$ such that $c_{|\alpha|} > 0$, i.e., $\mathcal{F} := \text{Span}\left(\{H_0\} \cup \{H_\alpha, \text{ for } \alpha \in (\mathbb{N}^d)^* \text{ such that } c_{|\alpha|>0}\}\right)$ and consider two penalties. First, we define a sparsity-inducing penalty, which depends on a hyper-parameter $r \in (0, +\infty)$

$$\Omega_{\text{var}}(f) = \left(\sum_{a=1}^{d} \left(\sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} \right)^{1/r}.$$

This penalty encourages sparsity in the dependence of f on individual variables, as it

pushes quantities of the form $\left(\sum_{\alpha\in(\mathbb{N}^d)^*}\alpha_a\frac{1}{c_{|\alpha|}}\hat{f}(\alpha)^2\right)^{r/2}$ to be 0. When this is the case, we obtain that $\forall\alpha\in(\mathbb{N}^d)^*,\ \alpha_a\neq 0, \hat{f}(\alpha)=0$, i.e., f does not depend on variable x_a (Lemma 12). When $r\geq 1$, Ω_{var} is a norm, which makes the problem easier to study from a theoretical point of view because if the loss is convex, this will yield a convex optimisation problem. However, estimators obtained through regularised empirical risk minimisation often suffer from bias due to the strong shrinkage associated with sparsity. Convex penalties can inadvertently reduce the significance of essential variables or features by excessive shrinkage to enforce sparsity. To address these issues, one can retrain on the set of selected variables or use concave penalties, which, despite presenting more analytical challenges, frequently deliver superior results by pushing the solution towards the boundary and enhancing sparsity [Zhang, 2010, Bach et al., 2012]. In this work, we adopt this strategy through the hyper-parameter r when $r<1$, which is the choice used in practice, while $r=1$ is used in the theoretical analysis.

The link with the nullity of the derivative can be seen using Equation (3.7)

$$\left(\sum_{\alpha\in(\mathbb{N}^d)^*}\alpha_a\frac{1}{c_{|\alpha|}}\hat{f}(\alpha)^2\right)^{r/2}=0 \iff \int_{\mathbb{R}^d}\left(\frac{\partial f}{\partial x_a}\right)^2 q=0.$$

Next, we introduce a smoothness-inducing norm, which penalises higher-order polynomials, i.e., those with large $|\alpha|$ (the dependence only on $|\alpha|$ is needed for future rotation invariance)

$$\Omega_0(f)=\left(\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{1}{c_{|\alpha|}}\hat{f}(\alpha)^2\right)^{1/2}.$$

It is important to note that Ω_0 is not integrated into the theoretical analysis and will be used with a much smaller and fixed parameter compared to Ω_{var}. Its primary purpose is to enforce numerical stability during the optimisation procedure, as discussed in Section 3.

The choice of $(c_k)_{k\in\mathbb{N}^*}$ significantly influences the behaviour of the penalties. In this work, we will consider two specific choices: $c_k=\mathbb{1}_{k\leq M}$ for some $M\in\mathbb{N}$ and $c_k=\rho^k$ for some $\rho\in[0,1)$. Both choices ensure that all three penalties are well-defined. Notably, when $M=1$, Ω_{var} considered with the quadratic loss reduces to the basic Lasso problem with linear features [Tibshirani, 1996].

It is worth mentioning that the coefficient $\hat{f}(0)$, which corresponds to the constant function $H_0=1$, is never penalised because it does not depend on any of the variables.

We then consider estimating f^* in the setting described in Assumption 6 by

$$f^{\lambda,\mu}_{\text{var}}:=\operatorname*{arg\,min}_{f\in\mathcal{F}}\ \widehat{\mathcal{R}}(f)+\lambda\Omega_0^2(f)+\mu\Omega^r_{\text{var}}(f), \tag{3.8}$$

with λ a fixed parameter and μ a hyper-parameter to be estimated. When $r\geq 1$ and the loss is convex, we obtain a strongly-convex objective function, hence with a unique global minimiser. When $r<1$, which we use in practice, only a local minimiser can be reached.

2.4 Hermite Polynomials for Feature Learning

We now turn to the feature learning setting described in Assumption 5. The Hermite polynomials are particularly well-suited for feature learning, as they allow us to bridge the gap between variable selection and feature learning with only a minor modification of the previous penalties. This suitability is visible in some important properties which we

now describe. First, the multivariate Hermite polynomials possess a rotation invariance property.

Lemma 13 (Rotational Invariance Property of Hermite Polynomials)**.** *For any $x, x' \in \mathbb{R}^d$, any $k \in \mathbb{N}$ and any orthogonal matrix $R \in O_d$,*

$$\sum_{|\alpha|=k} H_\alpha(x)H_\alpha(x') = \sum_{|\alpha|=k} H_\alpha(Rx)H_\alpha(Rx').$$

The proof of this lemma is available in Appendix A.1. This property will be extremely useful to characterise the statistical behaviour of our methods, as discussed in Section 4. Another key property is that for any $R \in O_d$, the family $\left(H_\alpha(R\cdot)\right)_{\alpha\in\mathbb{N}^d}$ also forms a basis of $L^2(q)$. Consequently, we can express any $f \in \mathcal{F}$ in this basis.

Moreover, we can characterise the derivatives of functions in $L^2(q)$ as in Equation (3.7). Let $f \in \mathcal{F}$ be written as $f = \sum_{\alpha\in\mathbb{R}^d} \hat{f}(\alpha)H_\alpha$, then using Equation (3.6), we have the following expressions for the derivatives

$$\int_{\mathbb{R}^d} \left(\frac{\partial f}{\partial x_a}\right)\left(\frac{\partial f}{\partial x_b}\right) q = \sum_{\alpha\in\mathbb{N}^d} \sqrt{(\alpha_a+1)}\sqrt{(\alpha_b+1)}\hat{f}(\alpha+e_a)\hat{f}(\alpha+e_b). \tag{3.9}$$

As before, we aim to enhance the regularisation using the sequence $(c_k)_{k>0}$ For $r \in (0, +\infty)$, we define

$$\begin{aligned} \Omega_{\text{feat}}(f) &= \left(\operatorname{tr}\left(M_f^{r/2}\right)\right)^{1/r} \\ \text{with } (M_f)_{a,b} &= \sum_{\alpha\in\mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \sqrt{\alpha_a+1}\sqrt{\alpha_b+1}\hat{f}(\alpha+e_a)\hat{f}(\alpha+e_b),\ a,b\in[d]. \end{aligned} \tag{3.10}$$

It is worth noting that M_f is a positive semi-definite matrix (see the proof of Lemma 14). The penalty Ω_{feat} pushes the eigenvalues of of M_f towards 0, and since the rank of M_f is equal to the number of its non-zero eigenvalues, the penalty encourages the rank of M_f to be low. It is crucial that $c_{|\alpha|}$ depends solely on $|\alpha|$ and not on any other quantities depending on α (e.g., $\max_{a\in[d]} \alpha_a$ for example). This property allows us to leverage the rotation invariance property described in Lemma 13, which is needed for our estimation algorithm in Section 3 and for obtaining statistical consistency results in Section 4.

Let us now examine some important properties of the proposed regularisation.

Lemma 14 (Properties of the Regularisation)**.** *For any $f \in \mathcal{F}$, the following properties hold*

1. *Let $R \in O_d$, if we define $g = f(R\cdot)$, then $M_f = RM_gR^\top$ and $\Omega_{\text{feat}}(f) = \Omega_{\text{feat}}(g)$.*
2. $\Omega_{\text{var}}(f) = \left(\operatorname{tr}\left(\operatorname{Diag}(M_f)^{r/2}\right)\right)^{1/r}$.
3. *If M_f is diagonal, $\Omega_{\text{feat}}(f) = \Omega_{\text{var}}(f)$.*
4. *Let $M_f = UDU^\top$ be the eigendecomposition of M_f, where $U \in O_d$ and D is a diagonal matrix. If we define $g = f(U\cdot)$, then $M_g = D$ is diagonal and thus $\Omega_{\text{feat}}(f) = \Omega_{\text{var}}(g)$.*
5. *Let $M_f = UDU^\top$ be the eigendecomposition as above. If the rank of D is s, then $g = f(U\cdot)$ only depends on variables x_a where $D_a > 0$ and $f = g(U^\top\cdot)$ only depends on s linear transformations of the original coordinates, namely of $(U^\top x)_a$ for a such that $D_a > 0$.*

6. *If* $r = 1$,
$$\Omega_{\text{feat}}(f) \geq \inf_{R \in O_d} \Omega_{\text{var}}(f(R\cdot)).$$

Proof of Lemma 14. We proceed by proving each assertion separately.

1. We have for $z \in \mathbb{R}^d$
$$\begin{aligned} z^\top M_f z &= \sum_{a,b=1}^{d} \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} z_a z_b \sqrt{\alpha_a + 1}\sqrt{\alpha_b + 1} \hat{f}(\alpha + e_a)\hat{f}(\alpha + e_b) \\ &= \sum_{a,b=1}^{d} \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} z_a z_b \Big\langle \frac{\partial f}{\partial x_a}, H_\alpha \Big\rangle_{L^2(q)} \Big\langle \frac{\partial f}{\partial x_b}, H_\alpha \Big\rangle_{L^2(q)} \\ &= \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \langle z^\top \nabla f, H_\alpha \rangle^2_{L^2(q)}. \end{aligned}$$

   This shows that M_f is positive semi-definite, writing $\mathcal{N}(0, I_d)$ for the standard normal distribution on $\mathbb{R}^d$, we then have
$$\begin{aligned} z^\top M_g z &= \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \Big(\mathbb{E}_{X \sim \mathcal{N}(0,I_d)} (z^\top \nabla g(X) H_\alpha(X)) \Big)^2 \\ &= \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \Big(\mathbb{E}_{X \sim \mathcal{N}(0,I_d)} (z^\top R^\top \nabla f(RX) H_\alpha(X)) \Big)^2 \\ &\quad \text{as } \nabla g(X) = R^\top \nabla f(RX) \\ &= \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \Big(\mathbb{E}_{X \sim \mathcal{N}(0,I_d)} (z^\top R^\top \nabla f(RX) H_\alpha(RX)) \Big)^2 \\ &\quad \text{by Lemma 13,} \\ &= \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \Big(\mathbb{E}_{X \sim \mathcal{N}(0,I_d)} (z^\top R^\top \nabla f(X) H_\alpha(X)) \Big)^2 \\ &\quad \text{by rotation invariance of the standard Gaussian,} \\ &= z^\top R^\top M_f R z, \end{aligned}$$

   that is $M_g = R^\top M_f R$. The second assertion follows by the rotation invariance of the trace.

2. It suffices to see that for any $a \in [d]$
$$\text{Diag}(M_f)_{a,a} = \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} (\alpha_a + 1)^2 \hat{f}(\alpha + e_a)^2 = \sum_{\alpha \in \mathbb{N}^d, \alpha_a > 0} \frac{1}{c_{|\alpha|}} \alpha_a \hat{f}(\alpha)^2,$$

   and therefore
$$\text{tr}\left(\text{Diag}(M_f)^{r/2}\right) = \sum_{a=1}^{d} \Big(\sum_{\alpha \in \mathbb{N}^d} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \Big)^{r/2} = \Omega_{\text{var}}(f)^r.$$

3. This is a direct consequence of the previous result, because of the definition of Ω_{feat}.

4. By applying the first result, we find that $\Omega_{\text{feat}}(f) = \Omega_{\text{feat}}(g)$ and $M_g = D$. Then, using the third result, we conclude that $\Omega_{\text{var}}(g) = \Omega_{\text{feat}}(g)$. This establishes the

desired result.

5. Consider the function $g = f(U\cdot)$. From the previous result, we know that $M_g = D$ is diagonal. According to the definition of Ω_{var}, we have $D_a = 0$ if and only if g does not depend on variable x_a. Consequently, if the rank of D is s, then g only depends on s variables, specifically those for which $D_a > 0$. As a result, we can conclude that $f = g(U^\top \cdot)$ depends solely on $(U^\top x)_a$ for a such that $D_a > 0$.

6. Let us examine Ω_{feat} and Ω_{var} as follows

$$\Omega_{\text{feat}}(f) = \big(\operatorname{tr}\big(M_f^{1/2}\big)\big), \qquad \Omega_{\text{var}}(f) = \big(\operatorname{tr}\big(\operatorname{Diag}(M_f)^{1/2}\big)\big).$$

We can decompose M_f as $M_f = UDU^\top$ using its eigendecomposition. If we define $g = f(U\cdot)$, then $M_g = D$ is diagonal, and we have $\Omega_{\text{feat}}(f) = \Omega_{\text{feat}}(g) = \Omega_{\text{var}}(g)$. Consequently, we obtain the inequality

$$\Omega_{\text{feat}}(f) \geq \inf_{R \in O_d} \Omega_{\text{var}}(f(R\cdot)).$$

□

The rotation invariance of Ω_{feat} is crucial in the context of feature learning, as it ensures that the penalty is not biased towards specific directions. Similarly, Ω_0 is also rotation invariant, as can be seen using Lemma 13.

We observe that given a function f and its associated matrix M_f, we can construct a function g consisting of a rotation of the data and f in such a way that the feature penalty on f is equal to the variable selection penalty on g. This highlights that the feature learning setting extends the variable selection problem by allowing data rotation. Furthermore, we can easily determine if g depends only on a few variables, and therefore if f depends only on a few linear transformations of the data, which aligns with our assumption for f^*. The last assertion of Lemma 14 will be useful to show that the proof of the consistency for the variable penalty easily extends to the feature learning setting, see Section 4.

With these considerations, we proceed to estimate f^* in the setting described by Assumption 5 by solving

$$f_{\text{feat}}^{\lambda,\mu} := \operatorname*{arg\,min}_{f \in \mathcal{F}} \ \widehat{\mathcal{R}}(f) + \lambda \Omega_0^2(f) + \mu \Omega_{\text{feat}}^r(f), \tag{3.11}$$

with λ a fixed parameter and μ a hyper-parameter. We refer to this estimator as the RegFeaL (regularised feature learning) estimator. As for the relevant features or variables and dimension, we discuss their computation in Section 3.1.

3 Estimator Computation

The computation of the solution for the optimisation problems delineated by (3.8) and (3.11) requires the employment of several strategic methodologies, which we will now discuss.

3.1 Variational Formulation

We first use the following quadratic variational formulation, similar to the approach presented in Bach et al. [2012]. This formulation is necessary since it is not possible to directly

optimise Equation (3.8) and Equation (3.11) due to the absence of closed-form solutions. Using other classical optimisation methods such as gradient-based methods would be less efficient as the overlapping group Lasso penalty we propose does not have efficient projection algorithms. Indeed, the variational formulation allows us to rewrite our optimisation problems as the minimisation over two variables of a specific quantity. Subsequently, we can alternate the minimisation with respect to each variable, leading to rapid convergence in practice.

We first give the following Lemma which is adapted from Jenatton et al. [2010], which provides a variational formulation of sums of powers.

Lemma 15 (Variational Formulation). *Let $r \in (0,2)$ and $u \in \mathbb{R}^d_+$, then*

$$\|u\|_{r/2}^{r/2} = \left(\sum_{a=1}^{d} u_a^{r/2}\right) = \min_{\eta \in \mathbb{R}^d_+,\ \|\eta\|_{r/(2-r)}=1} \sum_{a=1}^{d} \frac{u_a}{\eta_a},$$

with minimum attained at $\eta, \forall a \in [d],\ \eta_a = u_a^{(2-r)/2} / \left(\sum_{b=1}^{d} u_b^{r/2}\right)^{(2-r)/r}$.

Now, let us apply this approach to the penalty used for variable selection.

Lemma 16 (Variational Formulation of Variable Selection Penalty). *Let $f \in \mathcal{F}$ written as $f = \sum_{\alpha \in \mathbb{N}^d} \hat{f}(\alpha) H_\alpha$ and $r \in (0,2)$, then*

$$\begin{aligned}\Omega_{\text{var}}^r(f) &= \min_{\eta \in \mathbb{R}^d_+,\ \|\eta\|_{r/(2-r)}=1} \sum_{a=1}^{d} \left(\sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right) \eta_a^{-1} \\ &= \min_{\eta \in \mathbb{R}^d_+,\ \|\eta\|_{r/(2-r)}=1} \left(\sum_{\alpha \in (\mathbb{N}^d)^*} \alpha^\top \eta^{-1} \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right),\end{aligned}$$

where $\eta^{-1} = (1/\eta_1, \ldots, 1/\eta_d)$ and where the minimum is reached for η such that

$$\forall a \in [d],\ \eta_a = \frac{\left(\sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{(2-r)/2}}{\left(\sum_{b=1}^{d} \left(\sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_b \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} \right)^{(2-r)/r}}. \tag{3.12}$$

Proof of Lemma 16. Recall $\Omega_{\text{var}}(f) = \left(\sum_{a=1}^{d} \left(\sum_{\alpha \in (\mathbb{N}^d)^*} \frac{\alpha_a}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} \right)^{1/r}$ and use Lemma 15 with $u_a = \sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2$. □

We can then rewrite (3.8) as

$$\begin{aligned} f_{\text{var}}^{\lambda,\mu},\ \eta_{\text{var}}^{\lambda,\mu} = \underset{f \in \mathcal{F},\ \eta \in \mathbb{R}^d_+}{\arg\min} \quad & \widehat{\mathcal{R}}(f) + \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 (\lambda + \mu \alpha^\top \eta^{-1}) \\ \text{subject to} \quad & f = \sum_{\alpha \in \mathbb{N}^d} \hat{f}(\alpha) H_\alpha, \quad \|\eta\|_{r/(2-r)} = 1. \end{aligned} \tag{3.13}$$

Recall that $\Omega_{\text{var}}(f) = \left(\sum_{a=1}^{d} \left(\sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} \right)^{1/2}$. Each term $\left(\sum_{\alpha \in (\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2}$ quantifies the dependency of f on the variable x_a. We then

remark from the definition of $\eta_{\text{var}}^{\lambda,\mu}$ in Equation (3.12), that

$$\forall a \in [d],\ \left(\eta_{\text{var}}^{\lambda,\mu}\right)_a^{r/(2-r)} = \frac{\left(\sum_{\alpha\in(\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}_{\text{var}}^{\lambda,\mu}(\alpha)^2\right)^{r/2}}{\sum_{b=1}^d \left(\sum_{\alpha\in(\mathbb{N}^d)^*} \alpha_b \frac{1}{c_{|\alpha|}} \hat{f}_{\text{var}}^{\lambda,\mu}(\alpha)^2\right)^{r/2}}.$$

Hence $\left(\eta_{\text{var}}^{\lambda,\mu}\right)_a$ represents the variation of $f_{\text{var}}^{\lambda,\mu}$ which is due to x_a. We can use $\eta_{\text{var}}^{\lambda,\mu}$ to estimate the relevant underlying variables by using conventional techniques such as thresholding. Specifically, we can consider a variable x_a to be relevant only if η_a is above some predetermined threshold, i.e. $\hat{S} := \{a \in [d],\ \left(\eta_{\text{var}}^{\lambda,\mu}\right)_a > t\}$ for some $t > 0$.

We can proceed in a similar manner for the feature learning setting.

Lemma 17 (Variational Formulation of Feature Learning Penalty)**.** *Let $f \in \mathcal{F}$, M_f from Equation (3.10), with $M_f = UDU^\top$ its eigendecomposition and $r \in (0,2)$, then*

$$\begin{aligned}\Omega_{\text{feat}}^r(f) = \min_{\Lambda\in\mathbb{R}^{d\times d}} &\operatorname{tr}(\Lambda^{-1}M_f)\\ \text{subject to } &\Lambda = R\operatorname{Diag}(\eta)R^\top\\ &R \in O_d,\ \eta \in \mathbb{R}_+^d,\ \|\eta\|_{r/(2-r)} = 1,\end{aligned}$$

where the minimum is attained for

$$\begin{aligned}\Lambda &= U\operatorname{Diag}(\eta)U^\top\\ \forall a \in [d],\ \eta_a &= \frac{D_a^{(2-r)/2}}{(\sum_{b=1}^d D_b^{r/2})^{(2-r)/r}}.\end{aligned} \tag{3.14}$$

This allows us to rewrite Equation (3.11) as

$$\begin{aligned}f_{\text{feat}}^{\lambda,\mu},\ \Lambda_{\text{feat}}^{\lambda,\mu} = \operatorname*{arg\,min}_{f\in\mathcal{F},\ \Lambda\in\mathbb{R}^{d\times d}} &\widehat{\mathcal{R}}(f) + \lambda\Omega_0^2(f) + \mu\operatorname{tr}\left(\Lambda^{-1}M_f\right)\\ \text{subject to } &\Lambda = R\operatorname{Diag}(\eta)R^\top\\ &R \in O_d,\ \eta \in \mathbb{R}_+^d,\ \|\eta\|_{r/(2-r)} = 1.\end{aligned}$$

Moreover, with $\Lambda = R\operatorname{Diag}(\eta)R^\top$ as above, if we write f in the rotated basis as $f = \sum_{\alpha\in\mathbb{N}^d} \hat{f}(\alpha)H_\alpha(R^\top\cdot)$, and $g = f(R\cdot) = \sum_{\alpha\in\mathbb{N}^d} \hat{f}(\alpha)H_\alpha$, we have $M_f = RM_gR^\top$ (Lemma 14). Therefore

$$\begin{aligned}\operatorname{tr}\left(\Lambda^{-1}M_f\right) = \operatorname{tr}\left(\operatorname{Diag}(\eta^{-1})M_g\right) &= \sum_{a=1}^d \eta_a^{-1} \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{\alpha_a}{c_{|\alpha|}} \hat{f}(\alpha)^2\\ &= \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2\alpha^\top\eta^{-1}.\end{aligned}$$

We can then rewrite Equation (3.15) as

$$f^{\lambda,\mu}_{\text{feat}},\ \Lambda^{\lambda,\mu}_{\text{feat}} = \underset{f\in\mathcal{F},\ \Lambda\in\mathbb{R}^{d\times d}}{\arg\min}\ \widehat{\mathcal{R}}(f) + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{1}{c_{|\alpha|}}\hat{f}(\alpha)^2(\lambda + \mu\alpha^\top\eta^{-1}) \tag{3.15}$$
$$\text{subject to } \Lambda = R\,\text{Diag}(\eta)R^\top$$
$$R \in O_d,\ \eta\in\mathbb{R}^d_+,\ \|\eta\|_{r/(2-r)} = 1$$
$$f = \sum_{\alpha\in\mathbb{N}^d}\hat{f}(\alpha)H_\alpha(R^\top\cdot).$$

We see then that the feature learning problem can be viewed as an extension of the variable selection problem, where we additionally optimise over any possible data rotation. Conversely, the variable selection problem can be seen as a particular case of the feature learning problem, where the rotation matrix R is fixed to the identity matrix.

To estimate the dimension of the underlying feature space P and the features themselves, we use the eigendecomposition of $\Lambda^{\lambda,\mu}_{\text{feat}} = (R^{\lambda,\mu}_{\text{feat}})^\top\,\text{Diag}(\eta^{\lambda,\mu}_{\text{feat}})R^{\lambda,\mu}_{\text{feat}}$. By using the columns of $R^{\lambda,\mu}_{\text{feat}}$ corresponding to the selected features, denoted as $\hat{S} := \{a \in [d] \mid \left(\eta^{\lambda,\mu}_{\text{feat}}\right)_a > t\}$ for some threshold $t > 0$, we construct our feature estimator $\hat{P}$, i.e., $\hat{P} := (R^{\lambda,\mu}_{\text{feat}})_{\hat{S}}$. We see that by employing alternating minimisation, we are able to simultaneously learn the regression function and the underlying features.

3.2 Optimisation Procedure

We now discuss how to solve the optimisation problem using alternative minimisation, drawing on techniques described in Bach et al. [2012]. In the following discussion, we will focus on the feature learning setting. However, it is important to note that by simply fixing $R = I_d$ in each equation, we can easily revert back to the variable selection case.

To solve Equation (3.15), we have observed that when the function f is fixed, the optimal Λ can be determined using Equation (3.14), which involves the matrix M_f.[4]

When Λ is fixed, we seek to solve the optimisation problem

$$\underset{f\in\mathcal{F}}{\arg\min}\ \widehat{\mathcal{R}}(f) + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{1}{c_{|\alpha|}}\hat{f}(\alpha)^2(\lambda + \mu\alpha^\top\eta^{-1})$$
$$\text{subject to } f = \sum_{\alpha\in\mathbb{N}^d}\hat{f}(\alpha)H_\alpha(R^\top\cdot),$$

where $\Lambda = R\,\text{Diag}(\eta)R^\top$. However, this can only be solved if $\widehat{\mathcal{R}}$ is known, i.e., for some chosen loss function ℓ. Until the end of Section 3, we consider the quadratic loss which is commonly used in regression problems and allows for closed-form solutions. Otherwise, iterative optimisation algorithms need to be employed. The problem is then

$$\underset{f\in\mathcal{F}}{\arg\min}\ \frac{1}{n}\sum_{i=1}^n \left(y^{(i)} - f(x^{(i)})\right)^2 + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{1}{c_{|\alpha|}}\hat{f}(\alpha)^2(\lambda + \mu\alpha^\top\eta^{-1}) \tag{3.16}$$
$$\text{subject to } f = \sum_{\alpha\in\mathbb{N}^d}\hat{f}(\alpha)H_\alpha(R^\top\cdot).$$

[4] If $f = \sum_\alpha \hat{f}(\alpha)H_\alpha(R^\top\cdot)$, to compute M_f, we can remark that with $g = f(R\cdot) = \sum_\alpha \hat{f}(\alpha)H_\alpha$, we have the usual formula for M_g from Equation (3.10) and $M_f = RM_gR^\top$.

If we write for any $x, x' \in \mathbb{R}^d$

$$k_\Lambda(x, x') = \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|} H_\alpha(R^\top x) H_\alpha(R^\top x')}{\lambda + \mu \alpha^\top \eta^{-1}}, \tag{3.17}$$

the function k_Λ verifies all properties required to be a reproducing kernel [Aronszajn, 1950]. The condition for a function to be a reproducing kernel is that it is symmetric and that the associated kernel matrix is positive definite for any set of points. Specifically, for any $n \in \mathbb{N}$ and $x^{(1)}, \dots, x^{(n)}$, the matrix $K_\Lambda = (k_\Lambda(x^{(i)}, x^{(j)}))_{i,j \in [n]}$ must be positive definite (where $\lambda > 0$ is useful in this context). We can then apply the theory of reproducing kernel Hilbert spaces (RKHS). In this case, k_Λ serves as the reproducing kernel for the space $\mathcal{F}$, with associated norm $\|\cdot\|_\Lambda$, given by

$$\|f\|_\Lambda^2 = \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 (\lambda + \mu \alpha^\top \eta^{-1})$$

(note that $\hat{f}$ depends on Λ through R). We can interpret the problem as a standard kernel ridge regression, which we refer to as the "kernel point of view." By applying the representer theorem [Aronszajn, 1950], we know that the solution to Equation (3.16) takes the form

$$f = \sum_{i=1}^n \delta_i^\Lambda k_\Lambda(x^{(i)}, \cdot) + \delta_0^\Lambda,$$

where δ^Λ and δ_0^Λ can be obtained in closed form using $Y = (y^{(1)}, \dots, y^{(n)})^\top$ and $K = (k_\Lambda(x^{(i)}, x^{(j)}))_{i,j \in [n]}$ as the minimisers of

$$\delta^\Lambda,\ \delta_0^\Lambda = \underset{\delta \in \mathbb{R}^n,\ \delta_0 \in \mathbb{R}}{\arg\min} \frac{1}{n} \|Y - K_\Lambda \delta - \delta_0 \mathbb{1}\|_2^2 + \delta^\top K_\Lambda \delta. \tag{3.18}$$

It is worth noting that the shape of the kernel defined in Equation (3.17) implies that features corresponding to $\alpha \in \mathbb{N}^d$ with large values of $\alpha^\top \eta^{-1}$ are penalised more. If η_a is small, indicating that it has been pushed down in the previous optimisation steps, it suggests that variable x_a or the direction $(R^\top x)_a$ may not be particularly useful for prediction. In such cases, for these variables/directions to be retained, they would need to contribute significantly more to the fit compared to others.

Furthermore, we observe that the parameter λ serves the purpose of ensuring numerical stability when solving linear systems, particularly when $\alpha^\top \eta^{-1}$ can be null. We recommend setting λ to a significantly smaller value than μ to achieve this desired stability (e.g, $\lambda = 10^{-8}/d^{(2-r)/r}$ in our experiments). In fact, it is possible to fix λ as a predetermined value, eliminating the need for it to be treated as a hyper-parameter.

3.3 Sampling Approximation of the Kernel

We remark that the kernel described in Equation (3.17) is defined as an infinite sum, which means it is not computable in practice. To overcome this challenge, we adopt an approximation approach using sampling.

Let us define $C(\eta) = \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{\lambda + \mu \alpha^\top \eta^{-1}}$. By defining $h(\alpha) = \frac{1}{C(\eta)} \frac{c_{|\alpha|}}{\lambda + \mu \alpha^\top \eta^{-1}}$, for all $\alpha \in (\mathbb{N}^d)^*$ we obtain a probability distribution on $(\mathbb{N}^d)^*$. Consequently, we can express the kernel $k_\Lambda(x, x')$ as $C(\eta) \mathbb{E}_{\alpha \sim h}(H_\alpha(R^\top x) H_\alpha(R^\top x'))$.

Sampling from the distribution h can be challenging, particularly in high-dimensional settings. Therefore, we employ importance sampling techniques. For the first choice $c_{|\alpha|} = \mathbb{1}_{|\alpha|\leq M}$, the kernel $k_\Lambda(x, x')$ can be expressed as

$$\begin{aligned} k_\Lambda(x,x') &= \sum_{\alpha\in(\mathbb{N}^d)^*,|\alpha|\leq M} \frac{H_\alpha(R^\top x)H_\alpha(R^\top x')}{\lambda+\mu\alpha^\top\eta^{-1}} \\ &= \binom{M+d}{d}\mathbb{E}_{\alpha\sim\mathcal{U}\{|\alpha|\leq M\}}\left(\frac{H_\alpha(R^\top x)H_\alpha(R^\top x')}{\lambda+\mu\alpha^\top\eta^{-1}}\right), \end{aligned}$$

where $\mathcal{U}\{|\alpha| \leq M\}$ is the uniform distribution over $\{\alpha \in (\mathbb{N}^d)^*,\ |\alpha| \leq M\}$. Sampling from this uniform distribution can be achieved by selecting a subset B of size d uniformly from the set $[M + d]$, sorting the subset into $B_1 < \cdots < B_d$, setting $B_0 = 0$, and using the differences between consecutive values to construct α. Specifically, for each $a \in [d]$, we set $\alpha_a = B_a - B_{a-1} - 1$. If the resulting α is the null tuple, it is rejected, and the sampling process is repeated.

For the choice $c_{|\alpha|} = \rho^{|\alpha|}$ the kernel is

$$k_\Lambda(x,x') = \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{\rho^{|\alpha|}}{\lambda+\mu\alpha^\top\eta^{-1}} H_\alpha(R^\top x)H_\alpha(R^\top x').$$

We have developed a methodology called "group sampling" that addresses the challenges of sampling from the distribution h. To initialise the sampling, we set all components of η to be equal. This choice ensures unbiasedness among the possible directions while satisfying the constraint $\|\eta\|_{r/(2-r)} = 1$. As a result, the kernel takes the form

$$k_\Lambda(x,x') = \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{\rho^{|\alpha|}}{\lambda+\mu|\alpha|d^{(2-r)/r}} H_\alpha(R^\top x)H_\alpha(R^\top x').$$

We can directly sample from the distribution proportional to $\frac{\rho^{|\alpha|}}{\lambda+\mu|\alpha|d^{(2-r)/r}}$. The sampling process involves two steps. First, we sample an integer k from the distribution

$$k \sim \binom{k+d-1}{d-1}\frac{\rho^k}{\lambda+\mu d^{(2-r)/r}k}.$$

To perform this sampling, we can precompute a table of probabilities for different values of k up to a chosen maximum value (e.g., 40). We then normalise these probabilities and use them to sample the value of k. Once we have obtained k, it represents the cardinality of α. In the second step, we sample α uniformly from the set $\alpha \in (\mathbb{N}^d)^*,\ |\alpha| = k$. This sampling procedure is exact, except for the controlled approximation introduced by the choice of the maximum value.

We can develop an importance sampling scheme for the other optimisation steps when the components of η are not equal. Here are the steps.

1. Sort the components of η in ascending order and find the largest gap between consecutive values. Divide the set $[d]$ into two groups: Group 1, containing the components above the top of the gap, with size d_1, and Group 2, containing the remaining components, with size d_2.

2. Define $\tilde{\eta}_1$ as the minimum value among the components in Group 1, and $\tilde{\eta}_2$ as the

maximum value among the components in Group 2.

3. Sample k_1 and k_2 from the distribution

$$k_1, k_2 \sim \binom{k_1 + d_1 - 1}{d_2 - 1}\binom{k_2 + d_2 - 1}{d_2 - 1}\frac{\rho^{k_1+k_2}}{\lambda + \mu\left(\frac{k_1}{\tilde{\eta}_1} + \frac{k_2}{\tilde{\eta}_2}\right)},$$

where k_1 and k_2 represent $|\alpha^{(1)}|$ and $|\alpha^{(2)}|$ respectively, and $\alpha^{(1)}$ corresponds to the components in Group 1.

4. Sample $\alpha^{(1)}$ uniformly from the set $\alpha \in (\mathbb{N}^{d_1})$, $|\alpha| = k_1$, and sample $\alpha^{(2)}$ uniformly from the set $\alpha \in (\mathbb{N}^{d_2})$, $|\alpha| = k_2$.

5. This yields

$$\begin{aligned} k_\Lambda(x, x') &= \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{C(\tilde{\eta})}{C(\tilde{\eta})} \frac{\rho^{|\alpha|}}{\lambda + \mu\alpha^\top \tilde{\eta}^{-1}} \frac{\lambda + \mu\alpha^\top \tilde{\eta}^{-1}}{\lambda + \mu\alpha^\top \eta^{-1}} H_\alpha(R^\top x) H_\alpha(R^\top x') \\ &= \mathbb{E}_{\alpha \sim \text{Group sampling}}\left(C(\tilde{\eta}) \frac{\lambda + \mu\alpha^\top \tilde{\eta}^{-1}}{\lambda + \mu\alpha^\top \eta^{-1}} H_\alpha(R^\top x) H_\alpha(R^\top x')\right), \end{aligned}$$

with $C(\tilde{\eta})$ a normalising constant.

By using this importance sampling scheme, we can approximate the desired distribution accurately, even when the components of η are not equal.

We observe that with the group sampling approach, the distribution of α is influenced by η through the grouping process, as well as through the values of $\tilde{\eta}_1$ and $\tilde{\eta}_2$. As the optimisation progresses, the sampled tuples exhibit specific patterns: in directions that are deemed unimportant (corresponding to small values of η_a), α_a tends to be close to zero, while in directions that are considered important (corresponding to large η_a), α_a is more widely distributed.[5]

No matter the sampling scheme, we sample $\alpha^{(1)}, \ldots, \alpha^{(m)}$ from some distribution with importance weight $w(\alpha)$, yielding

$$k_\Lambda(x, x') \approx \sum_{j=1}^{m} w(\alpha^{(i)}) H_{\alpha^{(j)}}(R^\top x) H_{\alpha^{(j)}}(R^\top x').$$

We use this formula to compute the kernel matrix $K_\Lambda = \left(k_\Lambda(x^{(i)}, x^{(j)})\right)_{i,j \in [n]}$. Instead of approximating the matrix K_Λ to use in Equation (3.18), we can also consider the equivalent explicit "feature point of view" by writing f in the form

$$f = \sum_{j=1}^{m} \theta_j w(\alpha^{(j)}) H_{\alpha^{(j)}}(R^\top \cdot) + \theta_0 H_0,$$

where

$$\theta^\Lambda,\ \theta_0^\Lambda = \underset{\theta \in \mathbb{R}^m,\ \theta_0 \in \mathbb{R}}{\arg\min} \frac{1}{n}\|Y - \Phi\theta - \theta_0 \mathbb{1}\|_2^2 + \|\theta\|_2^2, \tag{3.19}$$

[5] It is worth noting that using the geometric distribution independently on each dimension of α would have been a simpler approach. However, this method becomes highly inefficient as the dimensionality increases, since it would involve sampling numerous α tuples with low importance weights (as determined by $\frac{H_\alpha(R^\top x) H_\alpha(R^\top x')}{\lambda + \mu\alpha^\top \eta^{-1}}$) due to their alignment with directions where η is very small (i.e., $\alpha^\top \eta^{-1}$ is small).

with $\Phi \in \mathbb{R}^{n\times m}$ the matrix filled with $w(\alpha^{(j)})H_{\alpha^{(j)}}(R^\top x^{(i)})$. This is computationally advantageous when $n > m$. Otherwise, we use the kernel point of view. In both cases, we can use (θ, θ_0) or (δ, δ_0) to rewrite f as $\sum_{\alpha\in\mathbb{N}^d} \hat{f}(\alpha)H_\alpha(R^\top\cdot)$. We remark that $\hat{f}(\alpha) = 0$ when α has not been sampled.

The pseudo-code for the RegFeaL method is provided in Algorithm 2.

for $i \in [n_{\text{iter}}]$ **do**
  if $i = 0$ **then**
    $\eta \leftarrow \mathbb{1}/d^{(2-r)/r}$;
    $R \leftarrow I_d$;
  else
    if feature learning **then**
      Update R and η as in Equation (3.14);
    else
      Update η as in Equation (3.12);
    end
  end
  Sample $\alpha^{(1)}, \dots, \alpha^{(m)}$ using group sampling as in Section 3.3 with η ;
  Compute importance weights $w(\alpha^{(1)}), \dots, w(\alpha^{(m)})$;
  Compute Hermite features $\Phi \in \mathbb{R}^{n\times m}, \Phi_{i,j} = w(\alpha^{(j)})H_{\alpha^{(j)}}(R^\top x^{(i)})$;
  if $n > m$ **then**
    Update θ as in Equation (3.19);
  else
    Update δ as in Equation (3.18);
  end
end

Algorithm 2: RegFeaL pseudocode.

In terms of numerical complexity, each iteration has a cost of

$$\mathcal{O}\Big(\underbrace{nm'd + nd^2}_{\text{Hermite features}} + \underbrace{d^2(m')^2 + d^3}_{M_f \text{ and its eigendecomposition}} + \underbrace{md}_{\text{Sampling}} + \underbrace{nm'\max(n, m')}_{\text{Computing } \theta \text{ or } \delta}\Big),$$

where m' is the number of unique tuples sampled (which is necessarily smaller than m, and can be much smaller when η is sparse). The parameter m can be chosen to achieve a balance between computational cost and performance, but selecting an excessively small value for m may adversely affect performance. In practice, the number of iterations required for convergence is typically very small (less than 10), as demonstrated in Section 5. Additionally, it is worth noting that the computation cost of δ in the feature point of view could be reduced through the use of the Nyström approximation [Rudi et al., 2015].

4 Statistical Properties

We now consider the statistical properties of RegFeaL. We always take $r = 1$ and we do not consider the approximation due to the computation of the estimators in this section. Our goal is to provide a high-probability bound on the expected risk of RegFeaL to gain insights into its generalisation properties under minimal assumptions to obtain a very general result. We do not consider the consistency of the e.d.r. space estimation, as this usually requires much stronger assumptions, such as the linearity condition, the gradient

along the relevant directions to be large enough in norm, or constraint on the loss to be the square loss, for example Cook and Weisberg [1991], Jing Zeng and Zhang [2024].

We leverage the results presented in Bach [2024], which provide bounds on the maximum difference between empirical and expected risk, in terms of the expectation over the class of functions with bounded norm. These bounds are expressed in terms of the Rademacher complexity of the set $\{f \in \mathcal{F},\ \Omega(f) \leq D\}$, where $D > 0$ is a fixed bound. By employing these results, we can obtain a probabilistic bound on the constrained estimator and apply McDiarmid's inequality [Boucheron et al., 2013] to establish a result in probability. Ultimately, Theorem 6 provides a probabilistic bound for the RegFeaL estimator, leveraging the aforementioned results as well as the optimality conditions satisfied by the estimator.

4.1 Setup

We start by making assumptions about the data used to train the model.

Assumption 7 (Data)**.** *$\mathcal{D} = (x^{(i)}, y^{(i)})_{i\in[n]}$ is a set of i.i.d data, with (X, Y) a pair of random variables such that $\forall i \in [n],\ (x^{(i)}, y^{(i)}) \sim (X, Y)$.*

Notice that we do not make strong assumptions on the distribution of the data, such as independence of the covariates or constraint to be elliptically contoured, nor do we require is to be known a priori.

Let us introduce some definitions. Let $(c_k)_{k>0}$ be a non-null sequence of positive reals. We define the function space $\mathcal{F}$ as Span $\left(\{H_0\} \cup \{H_\alpha,\ \text{for } \alpha \in (\mathbb{N}^d)^* \text{such that } c_{|\alpha|>0}\}\right)$. Let ℓ be a loss function on $\mathbb{R} \times \mathbb{R}$, and let the expected risk $\mathcal{R}$ and the empirical risk $\widehat{\mathcal{R}}$ be

$$\mathcal{R}(f) = \mathbb{E}_{X,Y}(\ell(Y, f(X))) \quad \text{and} \quad \widehat{\mathcal{R}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y^{(i)}, f(x^{(i)})).$$

We define the functional norm $\Omega(f)$ for any $f \in \mathcal{F}$ as $\Omega(f) := \Omega_{\text{feat}}(f) + |\hat{f}(0)|$ or $\Omega(f) := \Omega_{\text{var}}(f) + |\hat{f}(0)|$, where $\hat{f}(0)$ represents the constant coefficient of f. It is important to note that the constraint on the constant coefficient is not necessary in practice, but we include it for the purpose of theoretical analysis (we could also add a small weight on $|\hat{f}(0)|$ to this effect). We define the regularised empirical risk $\widehat{\mathcal{R}}\mu(f)$ for $\mu > 0$ as follows

$$\widehat{\mathcal{R}}\mu(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y^{(i)}, f(x^{(i)})) + \mu\Omega(f).$$

We denote our estimator as $f^\mu := \arg\min_{f\in\mathcal{F}} \widehat{\mathcal{R}}_\mu$. In order to establish theoretical results, we will rely on the following assumptions.

Assumption 8 (Problem Assumptions)**.**

1. *The true regression function $f^* := \arg\min_{f\in\mathcal{F}} \mathcal{R}(f)$ exists.*
2. *For some $D > 0$, the loss function ℓ is G-Lipschitz continuous in its second argument for any value of its first argument, i.e., $\forall y \in \mathcal{Y},\ \forall x, x' \in \mathcal{X},\ \forall f \in \mathcal{F}$ such that $\Omega(f) \leq D,\ |\ell(y, f(x)) - \ell(y, f(x'))| \leq G \cdot |f(x) - f(x')|$.*
3. *For some $D > 0$, $\ell_\infty := \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}, f\in\mathcal{F}, \Omega(f)\leq D} \ell(y, f(x))$ is finite.*
4. *The loss ℓ is convex on $\mathbb{R} \times \mathbb{R}$.*

For our main result, we will use $D = 2\Omega(f^*)$. These assumptions are commonly used in the analysis of nonparametric regression [Györfi et al., 2002]. Many commonly used loss functions in regression problems, such as the quadratic loss, absolute mean error, Huber loss, or logistic loss, are convex. The Lipschitz continuity condition holds for all of these losses, except for the quadratic loss, which we handle separately, for example by exploiting the boundedness of the data. If the data is bounded (i.e., $\mathcal{X}\times\mathcal{Y}$ is bounded in $\mathbb{R}^d\times\mathbb{R}$), then $\sup_{x\in\mathcal{X},f\in\mathcal{F},\Omega(f)\leq D}|f(x)|$ is bounded for any $D>0$.[6] We can then use the convexity of the loss ℓ and boundedness of $\mathcal{Y}$ to justify that ℓ_∞ is well-defined. For the quadratic loss, in this setting, it satisfies Assumption 8.2 because $(y-f(x))^2-(y-f(x'))^2=(f(x')-f(x))(y-f(x)+y-f(x'))$, and we can then take $G:=\sup_{(x,y)\in\mathcal{X}\times\mathcal{Y},f\in\mathcal{F},\Omega(f)\leq D}|y-f(x)+y-f(x')|$.

4.2 Rademacher Complexity

First, we apply the Lipschitz continuity assumption to bound the supremum over a set of functions of the difference between the empirical risk and expected risk, in expectation over the dataset.

Lemma 18 (Use of Gaussian Complexity)**.** *Let $\mathcal{G}$ be any set of functions, then under Assumption 7, and Assumption 8.2,*

$$\mathbb{E}_{\mathcal{D}}\left(\sup_{f\in\mathcal{G}}\left(\mathcal{R}(f)-\widehat{\mathcal{R}}(f)\right)+\sup_{f\in\mathcal{G}}\left(\widehat{\mathcal{R}}(f)-\mathcal{R}(f)\right)\right)\leq 4\sqrt{\frac{\pi}{2}}G\cdot G_n(\mathcal{G}),$$

where

$$G_n(\mathcal{G}):=\mathbb{E}_{\mathcal{D},\varepsilon\sim\mathcal{N}(0,I_n)}\left(\sup_{f\in\mathcal{G}}\frac{1}{n}\sum_{i=1}^n\varepsilon_i f(x^{(i)})\right)$$

is the Gaussian complexity of the set $\mathcal{G}$ [see Bartlett and Mendelson, 2002].

See Appendix A.2 for the proof, which we include for the sake of completeness. This is a close variation of the work presented in Bach [2024]. We now need to bound the Gaussian complexity, when we consider subsets of the working space $\mathcal{F}$ with bounded norm.

Lemma 19 (Bound on Gaussian Complexity)**.** *Let $D>0$, with $\mathcal{G}:=\{f\in\mathcal{F},\ \Omega(f)\leq D\}$ with Ω defined as in Section 4.1, under Assumption 7, we have*

$$G_n(\mathcal{G})\leq D\cdot\sqrt{\frac{1}{n}\left(1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)\right)}.$$

We remark that the result depends heavily on the data distribution through the expectations $\mathbb{E}_X(H_\alpha(X)^2)$ and the design of the norm through $(c_k)_{k>0}$. We discuss these in more details in Section 4.4.

Proof of Lemma 19. We first consider the norm Ω_{var}. Let $f\in\mathcal{G}$, we have

$$\frac{1}{n}\sum_{i=1}^n\varepsilon_i f(x^{(i)})=\sum_{\alpha\in\mathbb{N}^d}\hat{f}(\alpha)\left(\frac{1}{n}\sum_{i=1}^n\varepsilon_i H_\alpha(x^{(i)})\right)=\sum_{\alpha\in\mathbb{N}^d}\hat{f}(\alpha)\hat{\xi}(\alpha),$$

[6] This can be seen for Ω_{var} by noticing that $\Omega(f)$ can be written as $|\hat{f}(0)|+\sum_{a=1}^d\Theta_a(f)\geq\left(|\hat{f}(0)|^2+\sum_{a=1}^d\Theta_a(f)^2\right)^{1/2}$, with the latter being an RKHS norm with reproducing kernel $k(x,x')=1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}H_\alpha(x)H_\alpha(x')$. It follows that $f(x)=\langle f,k(X,\cdot)\rangle\leq\hat{f}(0)+\Omega(f)\sqrt{k(x,x)}$ which is bounded if x is bounded.

with ξ an infinite vector indexed by $(\mathbb{N}^d), \hat{\xi}(\alpha) = \frac{1}{n}\sum_{i=1}^n \varepsilon_i H_\alpha(x^{(i)})$. Therefore

$$\sup_{f\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(x^{(i)}) = \sup_{f\in\mathcal{G}} \sum_{\alpha\in\mathbb{N}^d} \hat{f}(\alpha)\hat{\xi}(\alpha) = D\cdot\Omega^*_{\text{var}}(\xi).$$

Now since Ω_{var} is the sum of $d+1$ semi-norms $\Theta_0, \Theta_1, \ldots, \Theta_d$, with

$$\Theta_0(f) = |\hat{f}(0)|$$
$$\Theta_a(f) = \Big(\sum_{\alpha\in(\mathbb{N}^d)^*} \frac{\alpha_a}{c_{|\alpha|}}\hat{f}(\alpha)^2\Big)^{1/2}, \forall a\in[d],$$

we have

$$\Omega^*_{\text{var}}(\xi) = \inf_{\xi=\sum_{a=0}^d \xi_a} \sup_{a\in\{0,\ldots,d\}} \Theta^*_a(\xi_a).$$

This is an extension of the fact that the set $\Omega^*_{\text{var}}(\xi)\le 1$ is the subdifferential of Ω_{var} at $f=0$, and thus the sum of the d subdifferentials of $\Omega_0,\ldots,\Omega_d$ at $f=0$. We consider $a\in[d]$, we have

$$\Omega^*_a(\xi_a)^2 = \sum_{\alpha\in\mathbb{N}^d,\alpha_a>0} \hat{\xi}_a(\alpha)^2\frac{c_{|\alpha|}}{\alpha_a},$$

and $\Omega^*_0(\xi)^2 = \hat{\xi}(0)^2$.

If we choose $\forall\alpha\in(\mathbb{N}^d)^*$, $\hat{\xi}_a(\alpha) = \frac{\sqrt{\alpha_a}}{\sum_b\sqrt{\alpha_b}}\hat{\xi}(\alpha)$, $\hat{\xi}_0(\alpha)=0$, $\hat{\xi}_a(0)=0$ and $\hat{\xi}_0(0)=\hat{\xi}(0)$, we have:

$$\begin{aligned}
\Omega^*_{\text{var}}(\xi)^2 &\le \sup\Big(\sup_{a\in[d]}\sum_{\alpha\in\mathbb{N}^d,\alpha_a>0}\hat{\xi}_a(\alpha)^2\frac{c_{|\alpha|}}{\alpha_a}, \hat{\xi}_0(0)^2\Big)\\
&\le \sup\Big(\sup_{a\in[d]}\sum_{\alpha\in\mathbb{N}^d,\alpha_a>0}\hat{\xi}(\alpha)^2\frac{c_{|\alpha|}}{\big(\sum_b\sqrt{\alpha_b}\big)^2}, \hat{\xi}(0)^2\Big)\\
&\le \sum_{\alpha\in\mathbb{N}^d}\hat{\xi}(\alpha)^2\Big(\frac{c_{|\alpha|}}{|\alpha|}\mathbb{1}_{|\alpha|>0}+\mathbb{1}_{|\alpha|=0}\Big).
\end{aligned}$$

Let $W^2 = \text{Diag}\,(\frac{c_{|\alpha|}}{|\alpha|}\mathbb{1}_{|\alpha|>0}+\mathbb{1}_{|\alpha|=0})$ and Φ the design matrix of all $H_\alpha(x^{(i)})$ (with n rows and infinitely many columns indexed with $\alpha\in\mathbb{N}^d$). We have $\hat{\xi} = \frac{1}{n}\Phi^\top\varepsilon$, and

$$\Omega^*_{\text{var}}(\xi)^2 \le \hat{\xi}^\top W^2\hat{\xi} = \frac{1}{n^2}\varepsilon^\top\Phi W^2\Phi^\top\varepsilon.$$

We compute the expectation of $\Omega^*_{\text{var}}(\xi)^2$ for $\varepsilon\sim\mathcal{N}(0,I_n)$, and get

$$\begin{aligned}
\mathbb{E}_\varepsilon(\Omega^*_{\text{var}}(\xi)^2) \le \mathbb{E}_\varepsilon\Big(\frac{1}{n^2}\varepsilon^\top\Phi W^2\Phi^\top\varepsilon\Big) &= \frac{1}{n^2}\,\text{tr}\,(\Phi W^2\Phi^\top)\\
&= \frac{1}{n}+\frac{1}{n^2}\sum_{\alpha\in(\mathbb{N}^d)^*}\sum_{i=1}^n\frac{c_{|\alpha|}}{|\alpha|}H_\alpha(x^{(i)})^2.
\end{aligned}$$

We now take expectations with regards to the data $\mathcal{D}$ and get

$$\mathbb{E}_{\mathcal{D},\varepsilon}(\Omega^*_{\text{var}}(\xi)^2) \leq \frac{1}{n} \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|} \mathbb{E}_X(H_\alpha(X)^2) + \frac{1}{n}.$$

Using Cauchy-Schwartz, $\mathbb{E}_{\mathcal{D},\varepsilon}(\Omega^*_{\text{var}}(\xi)) \leq \sqrt{\frac{1}{n}(1 + \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|} \mathbb{E}_X(H_\alpha(X)^2))}$.

From Lemma 14, we have

$$\Omega_{\text{feat}}(f) \geq \inf_{R \in O_d} \Omega_{\text{var}}(f(R\cdot)).$$

Then, for an infinite vector ξ indexed by $\mathbb{N}^d$, with $\hat{\xi}(\alpha) = \frac{1}{n}\sum_{i=1}^n \varepsilon_i H_\alpha(x^{(i)})$ and ξ_R an infinite vector indexed by $\mathbb{N}^d$ with $\hat{\xi}_R(\alpha) = \frac{1}{n}\sum_{i=1}^n \varepsilon_i H_\alpha(Rx^{(i)})$, we have $\Omega^*_{\text{feat}}(\xi) \leq \sup_{R \in O_d} \Omega^*_{\text{var}}(\xi_R)$.

Therefore

$$\sup_{R \in O_d} \Omega^*_{\text{var}}(\xi_R) \leq \sup_{R \in O_d} \frac{1}{n^2} \varepsilon^\top \Phi_R W^2 \Phi_R^\top \varepsilon,$$

with Φ_R the design matrix of all $H_\alpha(Rx^{(i)})$ (with n rows and infinitely many columns indexed with $\alpha \in \mathbb{N}^d$). Therefore using Lemma 13,

$$\begin{aligned}
\varepsilon^\top \Phi_R W^2 \Phi_R^\top \varepsilon &= \sum_{i,j} \varepsilon_i \varepsilon_j \left(1 + \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|} H_\alpha(Rx^{(i)}) H_\alpha(Rx^{(j)})\right) \\
&= \sum_{i,j} \varepsilon_i \varepsilon_j \left(1 + \sum_{k=1}^{+\infty} \frac{c_k}{k} \sum_{\alpha \in \mathbb{N}^d,\ |\alpha|=k} H_\alpha(Rx^{(i)}) H_\alpha(Rx^{(j)})\right) \\
&= \sum_{i,j} \varepsilon_i \varepsilon_j \left(1 + \sum_{k=1}^{+\infty} \frac{c_k}{k} \sum_{|\alpha|=k} H_\alpha(x^{(i)}) H_\alpha(x^{(j)})\right) = \varepsilon^\top \Phi W^2 \Phi^\top \varepsilon,
\end{aligned}$$

which is independent of R, therefore yielding exactly the same result as for Ω_{var} once expectation with regards to ε and the data is taken. □

4.3 Statistical Convergence

To gain insight into the proof technique, we initially establish an expectation-based result for the constrained estimator instead of the regularised estimator. We bound the expected risk of the function that minimises the empirical risk over the set of functions with a bounded norm, in expectation over the dataset. To accomplish this, we use Lemma 18 and Lemma 19.

Lemma 20 (Expected risk of Constrained Estimator)**.** *Let $D > \Omega(f^*)$ and let $f^D = \arg\min_{f \in \mathcal{F},\ \Omega(f) \leq D} \widehat{\mathcal{R}}(f)$, under Assumptions 7, 8.1 and 8.2,*

$$\mathbb{E}_{\mathcal{D}}(\mathcal{R}(f^D)) \leq \mathcal{R}(f^*) + \frac{4GD}{\sqrt{n}} \sqrt{\frac{\pi}{2}} \sqrt{1 + \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|} \mathbb{E}_X(H_\alpha(X)^2)}.$$

Proof of Lemma 20. With $\mathcal{G} := \{f \in \mathcal{F},\ \Omega(f) \leq D\}$, we have the classical decomposition

of the excess risk

$$\begin{aligned}\mathcal{R}(f^D) - \mathcal{R}(f^*) &= \mathcal{R}(f^D) - \widehat{\mathcal{R}}(f^D) + \widehat{\mathcal{R}}(f^D) - \widehat{\mathcal{R}}(f^*) + \widehat{\mathcal{R}}(f^*) - \mathcal{R}(f^*) \\ &\leq \mathcal{R}(f^D) - \widehat{\mathcal{R}}(f^D) + \widehat{\mathcal{R}}(f^*) - \mathcal{R}(f^*) \\ &\leq \sup_{f\in\mathcal{G}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f\in\mathcal{G}} \widehat{\mathcal{R}}(f) - \mathcal{R}(f).\end{aligned}$$

We then take the expectation over the data on both sides and use Lemma 18 and Lemma 19

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}(\mathcal{R}(f^D)) - \mathcal{R}(f^*) &\leq \mathbb{E}_{\mathcal{D}}\big(\sup_{f\in\mathcal{F}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big) \\ &\leq 4\sqrt{\frac{\pi}{2}} G \cdot G_n(\mathcal{G}) \\ &\leq \frac{4GD}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1 + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\big(H_\alpha(X)^2\big)},\end{aligned}$$

hence the desired result. □

In addition to the expectation-based result, obtaining a result with high probability for the constrained estimator is also of interest. This is achieved in Lemma 23, presented in Appendix A.3, by using McDiarmid's inequality [Boucheron et al., 2013]. To apply this inequality, an additional assumption is required: the boundedness of the loss (Assumption 8.3). However, the most significant and relevant result is the one obtained for the estimator that minimises the regularised empirical risk. This result is more realistic and imposes the additional requirement of convexity of the loss function.

Theorem 6 (High-Probability Bound on Expected Risk of Regularised Estimator)**.** *Under Assumption 7 and Assumptions 8.1, 8.2, 8.3 with $D = 2\Omega(f^*)$, 8.4, then for any $\delta \in (0,1)$, with the choice of regularising parameter*

$$\mu = \frac{8G}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1 + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\big(H_\alpha(X)^2\big)} + \frac{\ell_\infty 2\sqrt{2}}{\Omega(f^*)\sqrt{n}}\sqrt{\log\frac{2}{\delta}},$$

with probability larger than $1-\delta$

$$\begin{aligned}&\mathcal{R}(f^\mu) \leq \mathcal{R}(f^*) \\ &+ \Omega(f^*)\Big(\frac{16G}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1 + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\big(H_\alpha(X)^2\big)}\Big) + \frac{\ell_\infty 4\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}\end{aligned}$$

and $\Omega(f^\mu) \leq 2\Omega(f^*)$.

We now discuss the meaning of Theorem 6. The theorem states that with high probability, under the appropriate choice of the regularisation parameter, the norm of the estimator f^μ, $\Omega(f^\mu)$, is bounded by twice the norm of the true regression function f^*, $\Omega(f^*)$. We remark that the choice of regularisation parameter depends on $\Omega(f^*)$, however, this is not the case in the bounded setting, see the discussion in Section 4.4. Under Assumption 5 (feature learning setting) or Assumption 6 (variable selection setting), we know that $\Omega(f^*)$ does not depend explicitly on d but only on s, the underlying number of variables or dimension of the linear subspace.

The norm $\Omega(f^*)$ also helps us bound the difference between the expected risk of the estimator $\mathcal{R}(f^\mu)$ and the expected risk of the true regression function $\mathcal{R}(f^*)$. This difference, denoted as $\mathcal{R}(f^\mu) - \mathcal{R}(f^*)$, has a dependency on the number of samples n, with a convergence rate of $n^{-1/2}$, as expected for a Lipschitz loss and a well-specified model. However, the dependency on the dimension d of the original data is somewhat concealed in $\sqrt{1 + \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|} \mathbb{E}_X(H_\alpha(X)^2)}$. We provide a detailed analysis of this dependency for specific choices of the data distribution X and the sequence $(c_k)_{k>0}$ in Section 4.4.

Proof of Theorem 6. The proof is adapted from Bach [2024]. Define $f^{\mu*}$ as the minimiser of $\mathcal{R}_\mu := \mathcal{R} + \mu\Omega$ over $\mathcal{F}$. Now, for $D > 0, \tau > 0$ define the following convex set

$$\mathcal{C}_{D,\tau} = \{f \in \mathcal{F}, \Omega(f) \leq D, \mathcal{R}_\mu(f) - \mathcal{R}_\mu(f^{\mu*}) \leq \tau\}.$$

It has boundary

$$\partial\mathcal{C}_{D,\tau} = \{f \in \mathcal{F}, \Omega(f) \leq D, \mathcal{R}_\mu(f) - \mathcal{R}_\mu(f^{\mu*}) = \tau\},$$

i.e., the second constraint is the saturated one, for well chosen D and τ. This is because, if we consider a f such that $\Omega(f) = D$, since the optimality conditions for $f^{\mu*}$ give that $\Omega^*(\mathcal{R}'(f^{\mu*})) \leq \mu$, (with $\mathcal{R}'$ any subgradient of $\mathcal{R}$ which necessarily exists because $\mathcal{R}$ is convex since ℓ is convex) we have

$$\begin{aligned}
\mathcal{R}_\mu(f) - \mathcal{R}_\mu(f^{\mu*}) &= \mathcal{R}(f) + \mu\Omega(f) - \mathcal{R}(f^{\mu*}) - \mu\Omega(f^{\mu*}) \\
&\geq \langle \mathcal{R}'(f^{\mu*}), (f - f^{\mu*}) \rangle + \mu\Omega(f) - \mu\Omega(f^{\mu*}) \\
&\quad \text{by convexity with } \langle \cdot, \cdot \rangle \text{ associated to } \Omega \\
&\geq -\Omega^*(\mathcal{R}'(f^{\mu*}))\Omega(f - f^{\mu*}) + \mu\Omega(f) - \mu\Omega(f^{\mu*}) \\
&\quad \text{by Holder's inequality} \\
&\geq -\mu\Omega(f - f^{\mu*}) + \mu\Omega(f) - \mu\Omega(f^{\mu*}) \text{ by optimality of } f^{\mu*} \\
&\geq 2\mu\Omega(f) - 2\mu\Omega(f^{\mu*}) \text{ by the triangular inequality} \\
&\geq 2\mu D - 2\mu\Omega(f^{\mu*}) \text{ since } \Omega(f) = D, \\
&\geq 2\mu\Omega(f^*) \text{ by choosing } D = 2\Omega(f^*), \text{ since } \Omega(f^*) \geq \Omega(f^{\mu*}) \\
&\geq \tau, \text{ by choosing } \tau = \mu\Omega(f^*),
\end{aligned}$$

hence the desired result on the active constraint of the boundary. We now fix $\tau = \mu\Omega(f^*)$ and $D = 2\Omega(f^*)$.

Now if f^μ does not belong to $\mathcal{C}_{D,\tau}$, since $f^{\mu*}$ does, there is an element f in the segment $[f^\mu, f^{\mu*}]$ that belongs to $\partial\mathcal{C}_{D,\tau}$, i.e, $\Omega(f) \leq D$ and $\mathcal{R}_\mu(f) - \mathcal{R}_\mu(f^{\mu*}) = \tau$. Because the loss is convex, we have that $\widehat{\mathcal{R}}_\mu(f) \leq \max\{\widehat{\mathcal{R}}_\mu(f^\mu), \widehat{\mathcal{R}}_\mu(f^{\mu*})\} = \widehat{\mathcal{R}}_\mu(f^{\mu*})$. Therefore

$$\begin{aligned}
\tau = \mathcal{R}_\mu(f) - \mathcal{R}_\mu(f^{\mu*}) &\leq \mathcal{R}_\mu(f) - \mathcal{R}_\mu(f^{\mu*}) + \widehat{\mathcal{R}}_\mu(f^{\mu*}) - \widehat{\mathcal{R}}_\mu(f) \\
&\leq \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \widehat{\mathcal{R}}(f^{\mu*}) - \mathcal{R}(f^{\mu*}). \qquad (3.20)
\end{aligned}$$

From the proof of Lemma 23, for all $\delta \in (0, 1)$

$$\begin{aligned}
&\sup_{f \in \mathcal{F},\ \Omega(f) \leq D} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f \in \mathcal{F},\ \Omega(f) \leq D} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) \\
&\qquad \leq \frac{4GD}{\sqrt{n}} \sqrt{\frac{\pi}{2}} \sqrt{1 + \sum_{\alpha \in (\mathbb{N}^d)*} \frac{c_{|\alpha|}}{|\alpha|} \mathbb{E}_X(H_\alpha(X)^2)} + \frac{\ell_\infty 2\sqrt{2}}{\sqrt{n}} \sqrt{\log \frac{1}{\delta}}
\end{aligned}$$

with probability larger than $1-\delta$.

We apply this to the RHS of Equation (3.20) (as $\Omega(f) \leq D$ and $\Omega(f^{\mu*}) \leq D$), which is smaller than $\frac{4GD}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)} + \frac{\ell_\infty 2\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}$ with probability larger than $1-\delta$.

Now if τ is such that

$$\frac{4GD}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)} + \frac{\ell_\infty 2\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}} \leq \tau, \text{ i.e.,}$$

$$\Omega(f^*)\frac{8G}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)} + \frac{\ell_\infty 2\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}} \leq \mu\Omega(f^*)$$

$$\frac{8G}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)} + \frac{\ell_\infty 2\sqrt{2}}{\sqrt{n}\Omega(f^*)}\sqrt{\log\frac{1}{\delta}} \leq \mu$$

then f^μ belongs to $\mathcal{C}_{D,\tau}$ with probability larger than $1-\delta$.

If we choose $\mu = \frac{8GD}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)} + \frac{\ell_\infty 2\sqrt{2}}{\Omega(f^*)\sqrt{n}}\sqrt{\log\frac{1}{\delta}}$, then

$$\begin{aligned}
\mathcal{R}_\mu(f^\mu) &\leq \mathcal{R}_\mu(f^{\mu*}) + \tau \\
&\leq \mathcal{R}_\mu(f^{\mu*}) + \tau \\
&\leq \mathcal{R}_\mu(f^*) + \tau \\
&\leq \mathcal{R}(f^*) + \mu\Omega(f^*) + \tau \\
&\leq \mathcal{R}(f^*) + 2\mu\Omega(f^*) \\
&\leq \mathcal{R}(f^*) \\
&\quad + \Omega(f^*)\left(\frac{16G}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)}\right) + \frac{\ell_\infty 4\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}
\end{aligned}$$

and $\Omega(f^\mu) \leq D = 2\Omega(f^*)$ with probability larger than $1-\delta$. □

4.4 Dependence on Problem Parameters

As we have seen, Theorem 6 depends on some quantities we detail now. First, we provide a definition of subgaussian real variables, as given by Vershynin [2018].

Definition 3 (Subgaussian Variables)**.** *Let Z be a real-valued (not necessarily centred) random variable. Z is subgaussian with variance proxy σ^2 if and only if*

$$\forall t > 0, \max\left(\mathbb{P}(Z \geq t), \mathbb{P}(Z \leq -t)\right) \leq e^{-\frac{t^2}{2\sigma^2}}.$$

Data distribution. To begin, we aim to establish an upper bound for the expectation of the squared Hermite polynomials over the covariates.

Lemma 21 (Analysis of Data-Dependent Terms in Theorem 6)**.** *Let $\alpha \in \mathbb{N}^d$.*

1. *If $X \sim \mathcal{N}(0, I_d)$, then*
$$\mathbb{E}_X(H_\alpha(X)^2) = 1.$$

2. *If X is such that $\|X\|_2 \leq R$ a.s., then*

$$\mathbb{E}_X(H_\alpha(X)^2) \leq e^{\frac{R^2}{2}}.$$

3. *If X is such that $\|X\|_2$ is a subgaussian variable with variance proxy bounded by $\sigma^2 < 1/(36e)$, then*

$$\mathbb{E}_X(H_\alpha(X)^2) \leq e^{36e\sigma^2} \leq e.$$

The proof of this lemma is provided in Appendix A.4. Note that independence between the coordinates is not required, except in the first case, which is an illustration of the definition of the Hermite polynomials. It is worth noting that except in the Gaussian case, the bounds may not be ideal with respect to their dependency on d. However, these bounds rely heavily on the bound for Hermite polynomials in Equation (3.3), which is valid for all points on the real line and for all one-dimensional Hermite polynomials. Thus, it is expected that better bounds in expectation are possible.

Choice of $(c_k)_{k>0}$. The quantities in Theorem 6 are influenced by the design of the penalty, which is determined by the choice of the sequence $(c_k)_{k>0}$. This dependency is observed in $\Omega(f^*)$, ℓ_∞, and $\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X(H_\alpha(X)^2)}$. It is worth noting that the bounds provided in Lemma 21 do not rely on the specific value of α. Therefore, our focus is now on bounding the summation term $\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}$.

Lemma 22 (Analysis of Terms Depending on $(c_k)_{k>0}$ in Theorem 6)**.** *If $c_{|\alpha|} = \rho^{|\alpha|}$, with $\rho \in (0,1)$*

$$\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|} \leq \frac{1}{(1-\rho)^d}$$

and if $c_{|\alpha|} = \mathbb{1}_{|\alpha|\leq M}$,

$$\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|} \leq \frac{M+1}{d}\binom{M+d}{M+1}.$$

The proof of this result can be found in Appendix A.5. By combining the different results, in the case of bounded data, for example, we can derive a corollary of Theorem 6 as follows

Corollary 1 (High-Probability Bound on Expected Risk of Regularised Estimator for Bounded Data)**.** *Under Assumption 7 and Assumptions 8.1, 8.2, 8.3 with $D = 2\Omega(f^*)$, 8.4, if $\|X\|_2 \leq R$ a.s., $(c_k)_{k>0} = (\rho^k)_{k>0}$, then for any $\delta \in (0,1)$, with the choice of regularising parameter*

$$\mu = \frac{G}{\sqrt{n}}\sqrt{1+\frac{e^{R^2/2}}{(1-\rho)^d}}\left(8\sqrt{\frac{\pi}{2}}+2\sqrt{2}\sqrt{\log\frac{2}{\delta}}\right),$$

with probability larger than $1-\delta$

$$\mathcal{R}(f^\mu) \leq \mathcal{R}(f^*) + \Omega(f^*)\frac{G}{\sqrt{n}}\sqrt{1+\frac{e^{R^2/2}}{(1-\rho)^d}}\left(16\sqrt{\frac{\pi}{2}}+4\sqrt{2}\sqrt{\log\frac{2}{\delta}}\right)$$

and $\Omega(f^\mu) \leq 2\Omega(f^*)$.

The proof is provided in Appendix A.6. We note that the choice of the regularisation parameter is independent of the unknown norm $\Omega(f^*)$ or the distribution of X, as long

as R is known. In the derived bound, the value of G can be independent of d for certain loss functions such as the logistic loss. We observe that $\Omega(f^*)$ does not depend on the dimension d, but solely on the number of variables or the dimension of the linear subspace s. It is important to note that the method exhibits a strong dependence on the dimension, which does not overcome the curse of dimensionality. However, this is merely the first step towards solving the multi-index model through regularised empirical risk minimisation, leaving room for future work and improvements.

5 Numerical Study

In this section, we present the numerical results that demonstrate the behaviour and performance of RegFeaL. The implementation of the estimator, as well as the code to run the experiments, can be accessed online at `https://github.com/BertilleFollain/RegFeaL`. The RegFeaL estimator class is designed to be compatible with the Scikit-learn API [Pedregosa et al., 2011], ensuring seamless integration with existing machine learning workflows.

5.1 Setup

We describe the experiment setup, which includes data simulation, training procedure and metrics for evaluation.

Data. In each generated dataset, depending on whether we consider feature learning or variable selection, we construct the linear subspace P differently. In the feature learning case, we sample a matrix from the set of $d \times d$ orthogonal matrices O_d and select its first s columns to form P. For variable selection, we simply consider the first s variables to be the relevant ones. Note that while our experiments were conducted with independently generated covariates, our method is invariant to rotations (in the feature learning case) and sign changes of the data (in both feature learning and variable selection). As such, it is robust to potential correlation between the covariates. The i.i.d dataset $(x^{(i)}, y^{(i)})_{i\in[n]}$ is then generated as follows

$$
\begin{aligned}
&X \sim \mathcal{U}\{[-\sqrt{3}, \sqrt{3}]\}^d \\
&f^*(x) = \sin(2(P^\top x)_1) + \sin(2(P^\top x)_2), \forall x \in \mathbb{R}^d \text{ (sinus dataset)} \\
&f^*(x) = (P^\top x)_1 + (P^\top x)_2 - (P^\top x)_1^2 - (P^\top x)_2^2 + 2(P^\top x)_1 (P^\top x)_2^3 - 4, \\
&\forall x \in \mathbb{R}^d \text{ (polynomial dataset)} \\
&Y = f^*(X) + \sigma\varepsilon,\ \varepsilon \sim \mathcal{N}(0, 1).
\end{aligned}
$$

Each component of X has mean 0 and variance 1. Notably, in both datasets, the true regression function f^* depends on $s = 2$ linear combinations of the original variables. The importance of the noise can be controlled through the parameter σ. The test set $(x^{(i)}_{\text{test}}, y^{(i)}_{\text{test}})_{i\in[n_{\text{test}}]}$ is generated in a similar manner as the training set.

Training. The loss that we consider is the quadratic loss. We train RegFeaL on the training set with fixed values of λ and r, and we cross-validate on μ and ρ. The number of iterations n_{iter} depends on the experiment. Some of the parameters are the same in all experiments, such as $n_{\text{test}} = 5000, s = 2, \lambda = 10^{-8}/d^{(2-r)/r}, r = 0.33$.

The values of the grid used for cross-validation can be found in Appendix B. The training pipeline differs between Experiment 1 and Experiments 2 and 3.

In Experiment 1, for each parameter tuple (ρ, μ), we estimate the number of relevant dimensions $\hat{s}$ using $\hat{s} := |\{a \in [d], (\eta_{\text{feat}}^{\lambda,\mu})_a^{r/(2-r)} \geq 1/d\}|$. Recall that $\eta_a^{r/(2-r)}$, represents the importance of feature a, and at initialisation, it is set to $1/d$ for all $a \in [d]$. We then select $\hat{P}$ as the set of $\hat{s}$ eigenvectors of $\Lambda_{\text{feat}}^{\lambda,\mu}$ corresponding to the $\hat{s}$ largest eigenvalues. Finally, we train a final regressor using Multivariate Adaptive Regression Splines (MARS) [Friedman, 1991] on the dataset $(\hat{P}^\top x^{(i)}, y^{(i)})_{i\in[n]}$.

In Experiments 2 and 3, we simply use the output $f_{\text{feat}}^{\lambda,\mu}$ of Algorithm 2 as the prediction function. In both cases, the R^2 score is used as the evaluation metric, which is described in Equation (3.21).

Metrics. We evaluate the performance of RegFeaL using two metrics: the R^2 score [Wright, 1921] for regression performance and an adapted Grassmannian distance for feature learning performance.

The R^2 score is computed as

$$1 - \frac{\sum_{i=1}^{n_{\text{test}}} (y_{\text{test}}^{(i)} - y_{\text{pred}}^{(i)})^2}{\sum_{i=1}^{n_{\text{test}}} (y_{\text{test}}^{(i)} - \bar{y}_{\text{test}})^2}, \tag{3.21}$$

where $\bar{y}_{\text{test}}$ is the mean of the test response values. The R^2 score can be computed on both the training and test sets. A score of 1 indicates the best possible performance, while a score of $-\infty$ indicates the worst performance. A constant estimator that predicts the average response value corresponds to a score of 0.

For the feature learning score, we compute the Grassmannian distance between the true subspace P and the estimated subspace $\hat{P}$, which corresponds to the s largest eigenvalue for the score computation. Note that the knowledge of s is only necessary to compute this score and not necessary for training. Note also that this is not the same $\hat{P}$ that was used to retrain MARS in Experiment 1, as the dimension of that one is estimated. The score is defined as

$$\begin{aligned} &\|P(P^\top P)^{-1}P^\top - \hat{P}(\hat{P}^\top \hat{P})^{-1}\hat{P}^\top\|^2/(2s) \text{ if } s \leq d/2 \\ &\|P(P^\top P)^{-1}P^\top - \hat{P}(\hat{P}^\top \hat{P})^{-1}\hat{P}^\top\|^2/(2(d-s)) \text{ if } s > d/2, \end{aligned}$$

where s is the number of relevant dimensions. The best possible score is 1, indicating a perfect match between the true and estimated subspaces, while a score of 0 indicates no correspondence between the subspaces.

In the setting of variable selection, this discussion can be adapted as discussed in Section 3. The omitted details of the experiments can be found in Appendix B.

5.2 Results

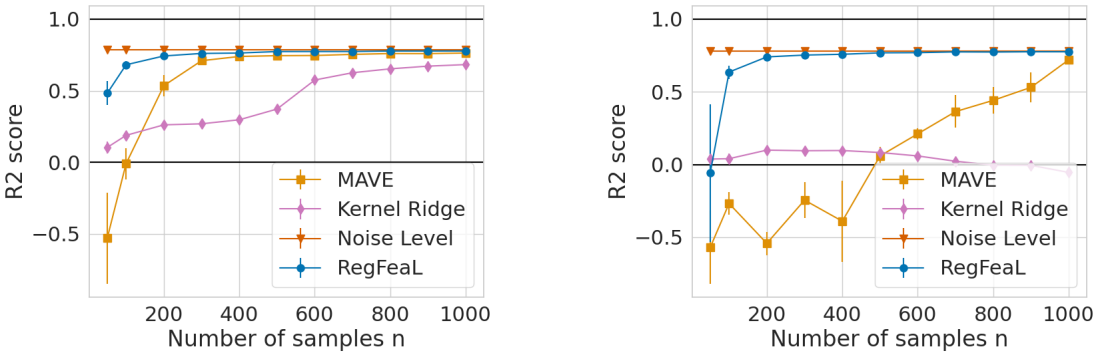
We now provide the results of the experiments.

Experiment 1. In this experiment, we investigate the dependence on the dimension of the variables d and the number of samples n. We perform the training procedure described earlier, including the retraining step using MARS [Friedman, 1991] on the projected data. We evaluate the performance on both the sinus dataset and the polynomial dataset with noise levels $\sigma = 0.5$ and $\sigma = 2.5$ respectively. For the sinus dataset, we consider both the

variable selection and feature learning settings. We conduct a total of $n_{\text{iter}} = 5$ iterations, and the grid used for cross-validation can be found in Appendix B.

To provide a comparison, we also include the performance of the state-of-the-art method MAVE [Xia et al., 2002], which is based on local averaging and does not use regularisation. In our implementation, we follow the recommended procedure for MAVE, which involves first training the Outer Product of Gradients (OPG) method to determine the effective dimensionality reduction (e.d.r) space. We use cross-validation to select the underlying dimension of the space and then retrain the model using MARS on the projected data. This allows us to compute the R^2 score. For the feature learning score, we compute it based on the learned effective dimensionality reduction (e.d.r) space. Specifically, we choose $s = 2$ as the dimension of the subspace to compute the score, following the same approach as RegFeaL.

Additionally, we include the R^2 score for Kernel Ridge, which uses kernel ridge regression with the kernel $k(x, x') = \sum_{\alpha \in (\mathbb{N}^d)^*} c_{|\alpha|} H_\alpha(x) H_\alpha(x')$ and the hyperparameter λ, which we cross-validate over. To provide a comprehensive analysis, we also display the noise level, which represents the best achievable score considering the noise level σ. We repeat the entire experiment five times, each time with different data, and present the average results with error bars of $\pm \sigma_{\text{exp}}/\sqrt{5}$, where σ_{exp} is the standard deviation of the scores across the repetitions. The results of the experiment can be found in Figures 3.1, 3.2, and 3.3.



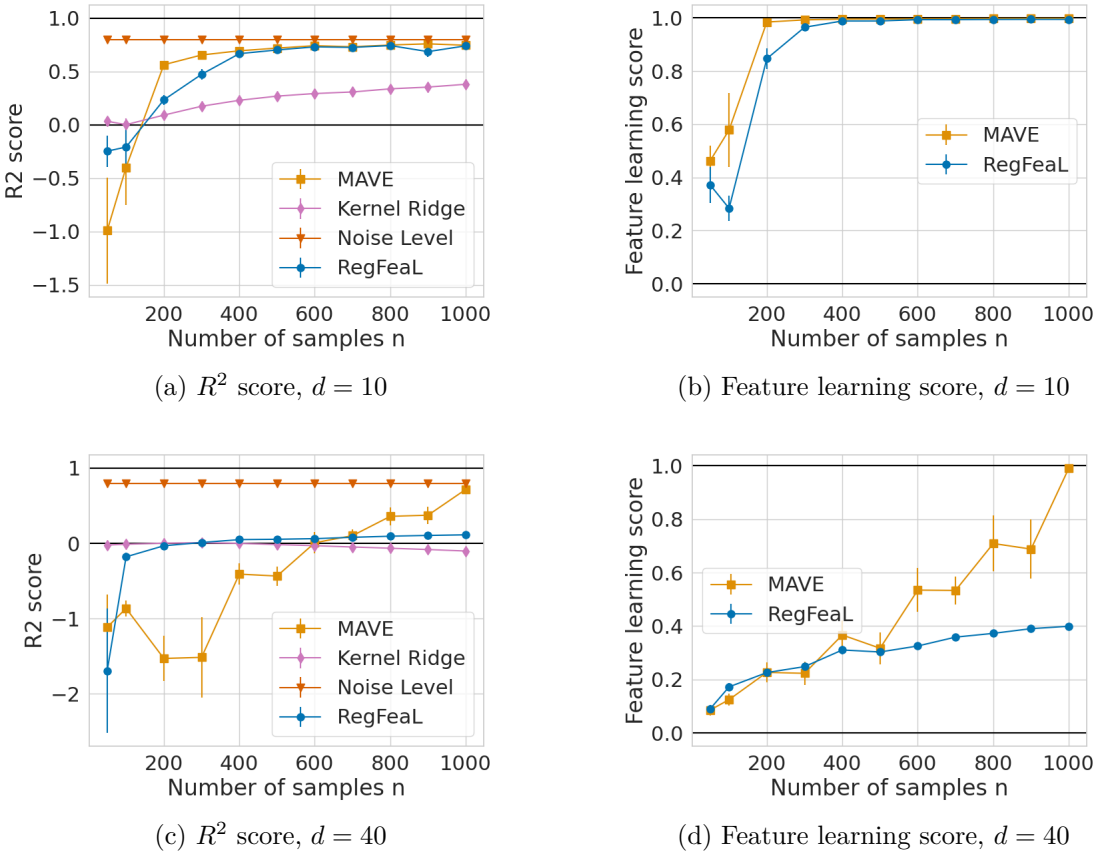
(a) R^2 score $d = 10$ (b) R^2 score, $d = 40$

Figure 3.1: Performance dependency on d and n for the sinus dataset in the variable selection setting.

In all figures, we observe that the performance improves with a higher number of samples (n), which is expected, while it deteriorates with a larger dimension (d), which is typical behaviour.

In Figure 3.1, we focus on the R^2 score for the sinus dataset in the variable regression setting. We observe that RegFeaL performs well in both dimensions (10 and 40) without requiring a large number of samples. However, Kernel Ridge fails in dimension 40 as the kernel cannot effectively capture the dependency on only 2 variables. As for MAVE, it does not benefit from the knowledge that this is a variable selection problem, unlike RegFeaL, resulting in a higher sample requirement, particularly in dimension 40.

In Figure 3.2, we examine the R^2 score and the feature learning score for the sinus dataset in the feature learning setting. We observe that MAVE and RegFeaL exhibit similar behaviour in dimension 10, reaching the noise level for the R^2 score and achieving a perfect feature learning score with enough samples. However, in dimension 40, MAVE



(a) R^2 score, $d = 10$

(b) Feature learning score, $d = 10$

(c) R^2 score, $d = 40$

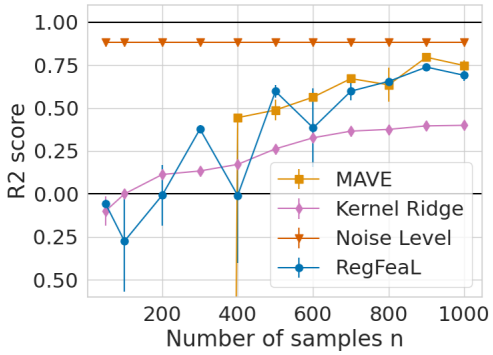
(d) Feature learning score, $d = 40$

Figure 3.2: Performance dependency on d and n for the sinus dataset in the feature learning setting.

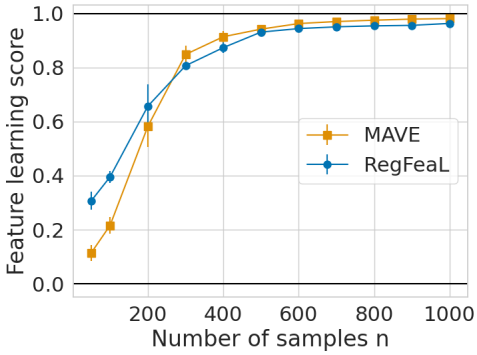
struggles significantly when the number of samples is low, while RegFeaL requires a substantially larger sample size to accurately learn the e.d.r. space. Our interpretation is that in this setting, where the true regression function uses a sinus, RegFeaL is hindered by its definition using a basis of polynomials.

In Figure 3.3, we investigate the R^2 score and the feature learning score for the polynomial dataset in the feature learning setting. The feature learning performance of MAVE and RegFeaL is similar in this scenario. Regarding the R^2 score, Kernel Ridge encounters difficulties in dimension 40 as it does not benefit from the underlying hidden structure. In dimension 10, RegFeaL performs similarly to MAVE, but in dimension 40, it outperforms MAVE as MAVE tends to be overly restrictive and consistently underestimates the number of linear features required to provide a good fit when the e.d.r. space is not perfectly learnt. In contrast, RegFeaL is less conservative, allowing us to leverage more features when the number of samples is too low to accurately estimate them.

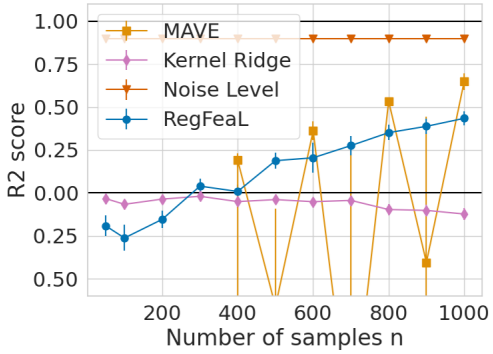
Experiment 2. In this experiment, we investigate the impact of the number of random features m (as discussed in Section 3.3) on the R^2 score and feature learning score for different values of n. The dimension d is fixed at 10, while the true underlying dimension s is 2. We consider the noiseless setting $\sigma = 0$ and use the sinus dataset. The same methodology is applied for error bar computation as in Experiment 1. The results are presented in Figure 3.4.



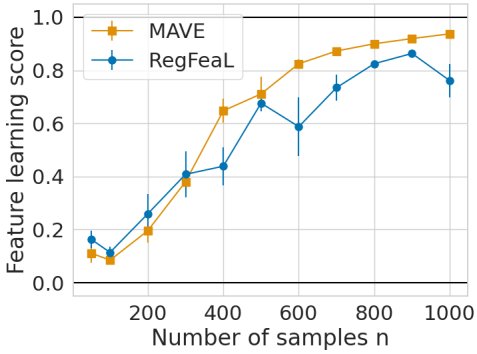
(a) R^2 score, $d = 20$



(b) Feature learning score, $d = 20$

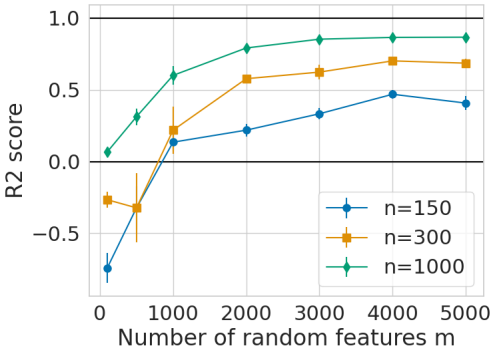


(c) R^2 score, $d = 40$

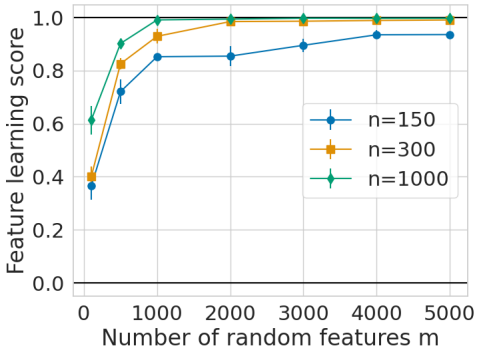


(d) Feature learning score, $d = 40$

Figure 3.3: Performance dependency on d and n for the polynomial dataset in the feature learning setting.



(a) R^2 score

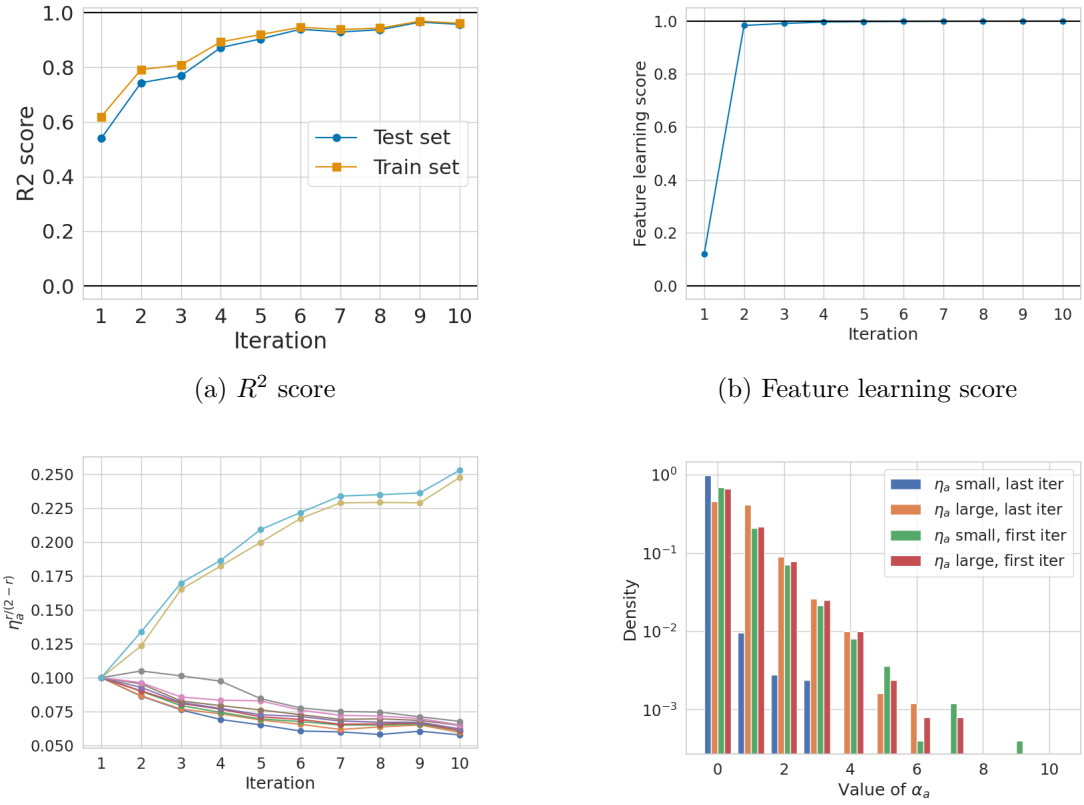


(b) Feature learning score

Figure 3.4: Influence of the number of random features.

We observe that both the R^2 score and feature learning score improve with an increase in the number of random features m. This observation aligns with the discussion in Section 3.3, where a larger value of m leads to a better approximation of the kernel k_Λ, and allows for a wider range of α and H_α, resulting in enhanced descriptive power and improved fit and prediction of the subspace. However, we note that beyond a certain value of m, the performance improvement levels off while computational costs continue to rise. This suggests that choosing excessively large values of m does not provide any significant benefit.

Experiment 3. In this experiment, we maintain the number of samples $n = 5000$, the number of random features $m = 2500$, the dimension $d = 10$, and the underlying dimension $s = 2$ fixed. We work with the noiseless sinus dataset, i.e., $\sigma = 0$, and examine the training behaviour of RegFeaL over the iterations. We train the model using cross-validation based on the R^2 score and set $n_{\text{iter}} = 10$. The results are depicted in Figure 3.5.



(a) R^2 score

(b) Feature learning score

(c) $\eta_a^{r/(2-r)}, \forall a \in [d]$

(d) Empirical distribution of α

Figure 3.5: Training behaviour.

In Figure 3.5a, we observe that the R^2 score improves across the iterations on both the test set and the training set. However, the behaviour is not strictly increasing on the training set. This can be attributed to the fact that the kernel approximation differs at each iteration, leading to variations in the fit.

Figure 3.5b demonstrates that the features are learned more rapidly than the fit. It is important to note that the feature learning score assumes knowledge of the underlying dimension $s = 2$. Hence, an important question is whether the estimated value of s is

accurate.

In Figure 3.5c, we observe the values of $\eta_a^{r/(2-r)}$ for all $a \in [d]$ across the iterations. Recall that $\sum_{a=1}^{d} \eta_a^{r/(2-r)} = 1$ and that $\eta_a^{r/(2-r)}$ represents the relative importance of feature $(R^\top x)_a$. Initially, all $\eta_a^{r/(2-r)}$ are equal to $1/d$. As the training progresses, most of the components of $\eta^{r/(2-r)}$ decrease, while two components increase, surpassing the threshold of $1/d$. These two components correspond to the relevant dimensions, indicating that the correct number of dimensions would be easily predicted. Additionally, we observe that these two components of η have relatively similar values, which aligns with the symmetry of the regression function in this example.

Figure 3.5d displays the empirical density (in log scale) of α_a for two different values of $a \in [d]$ (specifically, $a_{\text{small}} := \arg\min_{a\in[d]} \eta_a$ and $a_{\text{large}} := \arg\max_{a\in[d]} \eta_a$ for the final η) at two different iterations: the first and last iteration. During the first iteration, the distributions of α_a for a_{large} and a_{small} are equal, which aligns with the initialisation discussed in Section 3.3 (all components of η are equal). However, at the end of the optimisation, we observe that the distribution of $\alpha_{a_{\text{small}}}$, corresponding to a non-important linear feature, remains almost constant at 0. Conversely, the distribution of $\alpha_{a_{\text{large}}}$, representing an important linear feature, is more widely spread, which is beneficial to the fit.

6 Conclusion

We addressed the challenge of prediction function estimation in multi-index models by proposing a novel approach RegFeaL. Our method combines empirical risk minimisation with derivative-based regularisation to simultaneously estimate the prediction function, the relevant linear transformation, and its dimension. By leveraging the orthogonality and rotation invariance properties of Hermite polynomials, RegFeaL captures the underlying structure of the data. Through alternative minimisation, we iteratively rotate the data to better align it with the leading dimensions.

Theoretical results support the statistical consistency of the expected risk of our estimator and provide explicit rates of convergence. We demonstrated the performance and effectiveness of our method through extensive empirical experiments on diverse datasets. One of the strengths of our approach is that it does not rely on strong assumptions about the distribution shape or prior knowledge of the subspace dimension.

However, we acknowledge that our method is still subject to the curse of dimensionality, as indicated by the theoretical results showing an exponential dependence on the dimension of the covariates. Nonetheless, we believe that our findings will contribute to further developments in representation learning and high-dimensional data analysis. Regularisation is a versatile approach that can be applied to a wide range of problems where an empirical risk can be formulated, foregoing the limitations of some methods solely based on the square loss in supervised learning.

There are several interesting directions for future research. One possibility is exploring alternative bases other than Hermite polynomials. Additionally, investigating more efficient algorithms and strategies for handling high-dimensional data could be valuable. Furthermore, examining the applicability of our approach to various types of problems and datasets would also be worth pursuing.

Appendix

A Additional Proofs and Results

A.1 Proof of Lemma 13

Proof of Lemma 13. We denote by $\mathcal{N}(0, I_d)$ the d-dimensional normal distribution with mean $0 \in \mathbb{R}^d$ and covariance matrix I_d. For any $k \in \mathbb{N}$, $x, x' \in \mathbb{R}^d$, using $\forall z \in \mathbb{R}$, $h_k(z) = \frac{1}{\sqrt{k!}}\mathbb{E}_{Y\sim\mathcal{N}(0,1)}(z+iY)^k$ (which can be shown by recurrence), we have

$$\begin{aligned}\sum_{|\alpha|=k} H_\alpha(x)H_\alpha(x') &= \sum_{|\alpha|=k}\prod_{a=1}^{d} h_{\alpha_a}(x_a)h_{\alpha_a}(x'_a)\\ &=\mathbb{E}_{Y,Y'\sim\mathcal{N}(0,I_d)}\Big(\sum_{|\alpha|=k}\prod_{a=1}^{d}\frac{1}{\alpha_a!}(x_a+iY_a)^{\alpha_a}(x'_a+iY'_a)^{\alpha_a}\Big)\\ &=\frac{1}{k!}\mathbb{E}_{Y,Y'\sim\mathcal{N}(0,I_d)}\big((x^\top x' - Y^\top Y' + i(x^\top Y' + Y^\top x'))^k\big).\end{aligned}$$

This shows rotational invariance, that is, for any orthogonal matrix $R \in O_d$,

$$\sum_{|\alpha|=k} H_\alpha(x)H_\alpha(x') = \sum_{|\alpha|=k} H_\alpha(Rx)H_\alpha(Rx').$$

□

A.2 Proof of Lemma 18

Proof of Lemma 18. Define $\mathcal{H} = \{h : (x, y) \in \mathcal{X} \times \mathcal{Y} \to \ell(y, f(x)),\ \text{for } f \in \mathcal{G}\}$. We have that

$$\begin{aligned}&\sup_{f\in\mathcal{G}}\big(\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\big) + \sup_{f\in\mathcal{G}}\big(\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big)\\ &= \sup_{h\in\mathcal{H}}\Big(\mathbb{E}(h(z)) - \frac{1}{n}\sum_{i=1}^{n} h(z^{(i)})\Big) + \sup_{h\in\mathcal{H}}\Big(\frac{1}{n}\sum_{i=1}^{n} h(z^{(i)}) - \mathbb{E}(h(z))\Big).\end{aligned}$$

We define the Rademacher complexity of the set $\mathcal{H}$ by

$$R_n(\mathcal{H}) = \mathbb{E}_{\mathcal{D},\varepsilon\sim\left(\mathcal{U}\{-1,1\}\right)^n}\Big(\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i h(z^{(i)})\Big),$$

where $\varepsilon \sim \left(\mathcal{U}\{-1,1\}\right)^n$ means that each component of ε is independent and follows the uniform distribution over the set $\{-1,1\}$.

Using Proposition 4.2 from Bach [2024], we obtain

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\Big(\sup_{h\in\mathcal{H}}\mathbb{E}(h(z)) - \frac{1}{n}\sum_{i=1}^{n} h(z^{(i)})\Big) &\le 2R_n(\mathcal{H})\\ \mathbb{E}_{\mathcal{D}}\Big(\sup_{h\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n} h(z^{(i)}) - \mathbb{E}(h(z))\Big) &\le 2R_n(\mathcal{H}).\end{aligned}$$

Now from Assumption 8.2 and using Proposition 4.3 from Bach [2024]

$$R_n(\mathcal{H}) \leq G \cdot R_n(\mathcal{G}),$$

with

$$R_n(\mathcal{G}) = \mathbb{E}_{\mathcal{D},\varepsilon\sim\left(\mathcal{U}\{-1,1\}\right)^n}\left(\sup_{f\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(x^{(i)})\right).$$

We have from Exercise 4.9 from Bach [2024] that $R_n(\mathcal{G}) \leq \sqrt{\frac{\pi}{2}} G_n(\mathcal{G})$. Combining all inequalities yields the desired result. □

A.3 Lemma 23 and its Proof

Lemma 23. *Under Assumption 7, Assumptions 8.1, 8.2, 8.3, with $D \geq \Omega(f^*)$, and $f^D := \arg\min_{f\in\mathcal{F},\ \Omega(f)\leq D} \widehat{\mathcal{R}}(f)$, for any $\delta \in (0,1)$, with probability larger than $1-\delta$*

$$\mathcal{R}(f^D) \leq \mathcal{R}(f^*) + \frac{4GD}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1 + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)} + \frac{\ell_\infty 2\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}.$$

Proof of Lemma 23. Define $\mathcal{G} := \{f \in \mathcal{F},\ \Omega(f) \leq D\}$. We apply McDiarmid's inequality [Boucheron et al., 2013] to $\sup_{f\in\mathcal{G}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f\in\mathcal{G}} \widehat{\mathcal{R}}(f) - \mathcal{R}(f)$, which has bounded variation with constant $4\ell_\infty/n$, yielding that for all $\delta \in (0,1)$

$$\mathbb{P}_{\mathcal{D}}\Bigg(\sup_{f\in\mathcal{G}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f\in\mathcal{G}} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) \leq \\ \mathbb{E}\big(\sup_{f\in\mathcal{G}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f\in\mathcal{F}} \widehat{\mathcal{R}}(f) - \mathcal{R}(f)\big) + \frac{\ell_\infty 2\sqrt{2}}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}\Bigg) \geq 1-\delta.$$

We recall that

$$\mathcal{R}(f^D) - \mathcal{R}(f^*) \leq \sup_{f\in\mathcal{G}} \mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f\in\mathcal{G}} \widehat{\mathcal{R}}(f) - \mathcal{R}(f)$$

and from the proof of Lemma 20

$$\mathbb{E}\Big(\sup_{f\in\mathcal{G}} \left(\mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right) + \sup_{f\in\mathcal{G}} \left(\widehat{\mathcal{R}}(f) - \mathcal{R}(f)\right)\Big) \\ \leq \frac{4GD}{\sqrt{n}}\sqrt{\frac{\pi}{2}}\sqrt{1 + \sum_{\alpha\in(\mathbb{N}^d)^*} \frac{c_{|\alpha|}}{|\alpha|}\mathbb{E}_X\left(H_\alpha(X)^2\right)},$$

yielding the final result. □

A.4 Proof of Lemma 21

Proof of Lemma 21. The result for centred normal data with identity covariance matrix is by the construction of the Hermite polynomials [Hermite, 2009].

If $\|X\|_2$ is bounded by R, using the bound from Equation (3.3), we get that

$$\mathbb{E}_X(H_\alpha(X)^2) \leq \mathbb{E}(e^{\|X\|^2/2}) \leq \mathbb{E}_X(e^{R^2/2}) \leq e^{\frac{R^2}{2}}.$$

If X is such that $\|X\|$ is subgaussian with variance proxy σ^2, we know that $\forall\lambda \leq$

$1/(6\sqrt{2e}\sigma)$, then $\mathbb{E}_X(e^{\|X\|^2\lambda^2}) \leq e^{72e\lambda^2\sigma^2}$ [Vershynin, 2018, Proposition 2.5.2]. Therefore, using the bound from Equation (3.3), we have

$$\mathbb{E}_X(H_\alpha(X)^2) \leq \mathbb{E}(e^{\|X\|^2/2}) \leq e^{36e\sigma^2} \leq e$$

This concludes the study of $\mathbb{E}_X(H_\alpha(X)^2)$. □

A.5 Proof of Lemma 22

Proof of Lemma 22. Using d-dimensional geometric random variables, we know that

$$\sum_{\alpha\in\mathbb{N}^d}(1-\rho)^d\rho^{|\alpha|} = 1, \text{ and therefore } \sum_{\alpha\in(\mathbb{N}^d)^*}\frac{\rho^{|\alpha|}}{|\alpha|} \leq \frac{1}{(1-\rho)^d}.$$

For the other setting,

$$\sum_{\alpha\in(\mathbb{N}^d)^*,|\alpha|\leq M}\frac{1}{|\alpha|} = \sum_{k=1}^{M}\frac{1}{k}\binom{d-1+k}{d-1} \leq \frac{M+1}{d}\binom{M+d}{M+1},$$

which concludes the proof. □

A.6 Proof of Corollary 1

Proof of Corollary 1. First, we note from Lemma 21 that for any $\alpha \in \mathbb{N}^d$, we have $\mathbb{E}_X(H_\alpha(X)^2) \leq e^{R^2/2}$. Additionally, from Lemma 22, we know that $\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|} \leq \frac{1}{(1-\rho)^d}$.

Next, we aim to improve the use of McDiarmid's inequality by bounding the deviation of $\sup_{f\in\mathcal{F},\Omega(f)\leq D}\mathcal{R}(f) - \widehat{\mathcal{R}}(f) + \sup_{f\in\mathcal{F},\Omega(f)\leq D}\widehat{\mathcal{R}}(f) - \mathcal{R}(f)$ when a single data point $(x^{(i)}, y^{(i)})$ is changed to $(\tilde{x}^{(i)}, \tilde{y}^{(i)})$ changing the dataset from $\mathcal{D}$ to $\tilde{\mathcal{D}}$. In the original proof of Theorem 6, we used $4l_\infty/n$ as our bound, but we can provide a tighter bound. We write $\widehat{\mathcal{R}}_\mathcal{D}(f)$ to specify the dependency on the dataset. We also write $\mathcal{G} := \{f \in \mathcal{F},\ \Omega(f) \leq D\}$. Specifically, we have

$$\begin{aligned}
&\sup_{f\in\mathcal{G}}\mathcal{R}(f) - \widehat{\mathcal{R}}_\mathcal{D}(f) - \sup_{f\in\mathcal{G}}\mathcal{R}(f) - \widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f)\\
&= \sup_{f\in\mathcal{G}}\mathcal{R}(f) - \widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f) + \frac{1}{n}\ell(\tilde{y}^{(i)}, f(\tilde{x}^{(i)})) - \frac{1}{n}\ell(y^{(i)}, f(x^{(i)})) - \sup_{f\in\mathcal{G}}\mathcal{R}(f) - \widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f)\\
&\leq \sup_{f\in\mathcal{G}}\frac{1}{n}\ell(\tilde{y}^{(i)}, f(\tilde{x}^{(i)})) - \frac{1}{n}\ell(y^{(i)}, f(x^{(i)})),
\end{aligned}$$

and similarly

$$\sup_{f\in\mathcal{G}}\widehat{\mathcal{R}}_\mathcal{D}(f) - \mathcal{R}(f) - \sup_{f\in\mathcal{G}}\widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f) - \mathcal{R}(f) \leq \sup_{f\in\mathcal{G}}\frac{1}{n}\ell(y^{(i)}, f(x^{(i)})) - \frac{1}{n}\ell(\tilde{y}^{(i)}, f(\tilde{x}^{(i)}).$$

Combining both and taking the argmax functions f_1 and f_2, we obtain

$$\begin{aligned}
&\sup_{f\in\mathcal{G}}\mathcal{R}(f)-\widehat{\mathcal{R}}_{\mathcal{D}}(f)-\sup_{f\in\mathcal{G}}\mathcal{R}(f)-\widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f)+\sup_{f\in\mathcal{G}}\widehat{\mathcal{R}}_{\mathcal{D}}(f)-\mathcal{R}(f)-\sup_{f\in\mathcal{G}}\widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f)-\mathcal{R}(f)\\
&\quad\leq\frac{1}{n}\ell(\tilde{y}^{(i)},f_1(\tilde{x}^{(i)})-\frac{1}{n}\ell(y^{(i)},f_1(x^{(i)}))+\frac{1}{n}\ell(y^{(i)},f_2(x^{(i)}))-\frac{1}{n}\ell(\tilde{y}^{(i)},f_2(\tilde{x}^{(i)})\\
&\quad\leq\frac{G}{n}(|(f_1-f_2)(x^{(i)}|+|(f_1-f_2)(\tilde{x}^{(i)})|)\\
&\quad\leq\frac{4}{n}G\sup_{f\in\mathcal{F},\Omega(f)\leq D,x\in\mathbb{R}^d,\|x\|_2\leq R}|f(x)|\\
&\quad\leq\frac{4}{n}GD\sup_{x\in\mathbb{R}^d,\|x\|_2\leq R}\Omega^*((H_\alpha(x))_\alpha)\\
&\quad\leq\frac{4}{n}GD\sup_{x\in\mathbb{R}^d,\|x\|_2\leq R}\sqrt{1+\sum_{\alpha\in(\mathbb{N}^d)^*}\frac{c_{|\alpha|}}{|\alpha|}H_\alpha(x)^2}\\
&\quad\leq\frac{4}{n}GD\sqrt{1+\frac{e^{R^2/2}}{(1-\rho)^d}}.
\end{aligned}$$

We can obtain the same exact bound for the opposite quantity of

$$\sup_{f\in\mathcal{G}}\mathcal{R}(f)-\widehat{\mathcal{R}}_{\mathcal{D}}(f)-\sup_{f\in\mathcal{G}}\mathcal{R}(f)-\widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f)+\sup_{f\in\mathcal{G}}\widehat{\mathcal{R}}_{\mathcal{D}}(f)-\mathcal{R}(f)-\sup_{f\in\mathcal{G}}\widehat{\mathcal{R}}_{\tilde{\mathcal{D}}}(f)-\mathcal{R}(f)$$

by using the same arguments. We use this bound for $D=2\Omega(f^*)$. The result follows by employing the proof of Theorem 6. □

B Technical Details of the Numerical Experiments

Experiment 1. For MAVE and RegFeaL, the MARS final training used the default parameters provided by the py-earth python package (https://contrib.scikit-learn.org/py-earth/), except for the maximum degree, which was taken as the estimated dimension for both methods. MAVE was run using the provided CRAN package in R (https://cran.r-project.org/web/packages/MAVE/index.html) and the default parameters.

The number of iterations n_{iter} was set to 5. For RegFeaL, the cross-validation for $\rho\times\mu$ was done over the grid defined by $(0.01,0.05,0.1,0.2,0.4,0.6,0.8)$ for ρ and $(1000,100,10,1,0.1,0.01,0.001)/d^{((2-r)/r)}$ for μ.

The cross-validation for Kernel Ridge was done on parameter λ, with the set of values $(1000,100,10,1,0.1,0.01,0.001)/d^{((2-r)/r)}$. The score of the noise level was estimated by $1-\frac{n\sigma^2}{\sum_{i=1}^n(y_{\text{test}}^{(i)}-\bar{y}_{\text{test}})^2}$.

Experiment 2. For each value of n, we used cross-validation for the largest value of m and then used the selected ρ and λ for all other values of m. The cross-validation was done over the grid defined by $(0.2,0.4,0.6,0.8,1.0)$ for ρ and $(100,1,0.1,0.01,0.001)/d^{((2-r)/r)}$ for μ. The number of iterations n_{iter} was 3.

Experiment 3. The cross-validation for $\rho\times\mu$ was done over the grid defined by the sequences $(0.2,0.4,0.6,0.8,1.0)$, for ρ and $(100,1,0.1,0.01,0.001)/d^{((2-r)/r)}$, for μ.

CHAPTER 4

Integrating Neural Networks and Kernel Methods

The contents of this chapter are available in the preprint (under review by the Journal of Machine Learning Research) B. Follain and F. Bach. Enhanced Feature Learning via Regularisation: Integrating Neural Networks and Kernel Methods, 2024. URL https://arxiv.org/abs/2407.17280 while the code is available at https://github.com/BertilleFollain/BKerNN.

Abstract

We propose a new method for feature learning and function estimation in supervised learning via regularised empirical risk minimisation. Our approach considers functions as expectations of Sobolev functions over all possible one-dimensional projections of the data. This framework is similar to kernel ridge regression, where the kernel is $\mathbb{E}_w(k^{(B)}(w^\top x, w^\top x'))$, with $k^{(B)}(a,b) := \min(|a|,|b|)\mathbb{1}_{ab>0}$ the Brownian kernel, and the distribution of the projections w is learnt. This can also be viewed as an infinite-width one-hidden layer neural network, optimising the first layer's weights through gradient descent and explicitly adjusting the non-linearity and weights of the second layer. We introduce an efficient computation method for the estimator, called Brownian kernel neural network (BKerNN), using particles to approximate the expectation. The optimisation is principled due to the positive homogeneity of the Brownian kernel. Using Rademacher complexity, we show that BKerNN's expected risk converges to the minimal risk with explicit high-probability rates of $O(\min((d/n)^{1/2}, n^{-1/6}))$ (up to logarithmic factors). Numerical experiments confirm our optimisation intuitions, and BKerNN outperforms kernel ridge regression, and favourably compares to a one-hidden layer neural network with ReLU activations in various settings and real data sets.

Contents

1 Introduction

In the era of high-dimensional data, effective feature selection methods are crucial. Representation learning aims to automate this process, extracting meaningful information from complex data sets. Non-parametric methods often struggle in high-dimensional settings, making the multi-index model, which assumes a few relevant linear features explain the relationship between response and factors, an attractive alternative. Formally, the multiple index model [Xia, 2008] is expressed as $Y = f^*(X) + \text{noise} = g^*(P^\top X) + \text{noise}$, with Y the response, X the d-dimensional covariates, g^* the unknown link function, $P \in \mathbb{R}^{d\times k}$ the features and $k \leq d$, the number of such relevant linear features[1]. The components $P^\top X$ are linear features of the data that need to be learnt, reducing the dimensionality of the problem, which may allow to escape the curse of dimensionality, while the more general function g increases the capacity of the model.

Multiple index models have been extensively studied, leading to various methods for estimating the feature space. Brillinger [2012] introduced the method of moments for Gaussian data and one feature, by using specific moments to eliminate the unknown function. For features of any dimension, several methods have been proposed. Sliced inverse regression (SIR) [Li, 1991] uses second-order moments to identify effective dimensions by slicing the response variable and finding linear combinations of predictors, while improvements have been proposed [Yang et al., 2017], these methods heavily rely on assumptions about the covariate distribution shape and prior knowledge of the distribution. Iterative improvements have been an interesting line of work [Dalalyan et al., 2008], while optimisation-based methods like local averaging minimise an objective function to estimate the subspace [Fukumizu et al., 2009, Xia et al., 2002]. Despite their practical performance, particularly the MAVE method [Xia et al., 2002], the theoretical guarantees show exponential dependence on the original data dimension, making them less suitable for high-dimensional settings.

In this work, we tackle feature learning and function estimation jointly through the paradigm of empirical risk minimisation. We consider a classical supervised learning problem. We have i.i.d. samples $(x_i, y_i)_{i\in[n]}$ from a random variable $(X, Y) \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^d \times \mathbb{R}$. Our goal is to minimise the expected risk, which is defined as $\mathcal{R}(f) := \mathbb{E}_{X,Y}[\ell(Y, f(X))]$ over some class of functions $\mathcal{F}$, where ℓ is a loss function mapping from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}$. This can be achieved through the framework of regularised empirical risk minimisation, where the empirical risk is defined as $\widehat{\mathcal{R}}(f) := \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i))$. Our interest in regularised empirical risk minimisation stems from its flexibility, allowing it to be applied to a wide range of problems as long as the objective can be defined as the optimisation of an expected loss. Our primary objective is to achieve the lowest possible risk, which we study in theory and in practice, while we explore the recovery of underlying features in numerical experiments. Our method draws inspiration from several lines of work, namely positive definite kernels and neural networks with their mean field limit, which we briefly review, together with the main limitations we aim to alleviate.

Kernel methods and multiple kernel learning. A well-known method in supervised learning is kernel ridge regression [KRR, Vovk, 2013], which implicitly maps data into high-dimensional feature spaces using kernels. It benefits from dimension-independent rates of convergence if the model is well-specified, i.e., if the target function belongs to the related Hilbert space. However, KRR does not benefit from the existence of linear features in terms of convergence rates of the risk when the model is misspecified [Bach,

[1] Note that in all other chapters, this was called s.

2024, Section 9.3.5], as it relies on pre-specified features. To address the limitations of single-kernel methods, multiple kernel learning (MKL) optimally combines multiple kernels to capture different data aspects [Bach et al., 2004, Gönen and Alpaydın, 2011]. However, MKL suffers from significant computational complexity and the critical choice of base kernels, which can introduce biases if not selected properly. Furthermore, MKL does not resolve the issue of leveraging hidden linear features effectively.

Neural networks. Now consider another type of supervised learning methods, namely neural networks with an input layer of size d, a hidden layer with m neurons, an activation function σ, followed by an output layer of size 1. Functions which can be represented are of the form $f(x) = \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j)$, where σ can be the ReLU, $\sigma(z) = \max(0, z)$ or the step function $\sigma(z) = \mathbb{1}_{z>0}$. Neural networks benefit from hidden linear features, achieving favourable rates dependent on k, the number of relevant features, rather than on the data dimension [Bach, 2024, Section 9.3.5]. However, this formulation requires multiple b values to fit a function with the same w, particularly in single-index models $f^*(x) = g^*(w^\top x)$, which is inefficient.

Regularising by adding a penalty term to the empirical risk minimisation objective guides specific estimator behaviours. In the context of feature learning, Rosasco et al. [2013] used derivatives for regularisation in nonparametric models, focusing on variable selection. While their method reduces to classical regularisation techniques for linear functions, it faces limitations: functions depending on a single variable do not belong in the chosen RKHS, using derivatives at data points limits the exploitation of regularity, and there is no benefit from hidden variables in the misspecified case. An improvement over this method was studied for both feature learning and variable selection by Follain and Bach [2024b], where a trace norm penalty [Koltchinskii et al., 2011, Giraud, 2014] on the derivatives was used for the feature learning case. However, the dependency on the dimension of the rate did not allow high-dimensional learning. We can justify the use of trace norm penalties by considering the structure of neural networks. Under the multiple-index model, the weights $w_1, \dots, w_m$ of the first layer are expected to lie in a low-rank subspace of rank at most k. However, directly enforcing a rank constraint is not practical for optimisation. Therefore, we could use a relaxation such as $\Omega(f) = \operatorname{tr}\left(\left(\sum_{j=1}^{m} w_j w_j^\top\right)^{1/2}\right)$, which is the trace norm of a matrix containing the weights, to approximate the rank constraint effectively. However, there is still the issue of multiple constant terms for a single weight. We will see specialised penalties for feature learning for a different family of functions.

Mean-field limit. To apply a similar framework to our future estimator, we introduce the mean-field limit of an over-parameterised one-hidden layer neural network [Nitanda and Suzuki, 2017, Mei et al., 2019, Chizat and Bach, 2022, Sirignano and Spiliopoulos, 2020]. When the number of neurons m is very large, the network can be rescaled as follows

$$f(x) = \frac{1}{m} \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j), \text{ which approximates } \int \eta \sigma(w^\top x + b) \, \mathrm{d}\mu(\eta, w, b), \tag{4.1}$$

where μ is a probability distribution, and we can take the weights and constant terms (w, b) to be constrained when the activation is 1-homogeneous,[2] such as the ReLU or step function. This approach is valuable because, as noted by Chizat and Bach [2022], under certain conditions (convexity of the loss and penalty functions, homogeneity of the activation function), the regularised empirical risk problem optimised via gradient descent

[2] A function Φ is positively 1-homogeneous if, for any $\kappa > 0$, $\Phi(\kappa w) = \kappa \Phi(w)$.

in the infinitely small step-size limit converges to the minimiser of the corresponding problem with infinitely many particles. This allows us to use a finite number of particles m in practice while still leveraging the theoretical benefits derived from the continuous framework.

Plan of the chapter and notations. In this chapter, we introduce the Brownian kernel neural network (BKerNN), a novel model for feature learning and function estimation. Our approach combines kernel methods and neural networks using regularised empirical risk minimisation. Section 2 presents the theoretical foundations and formulation of BKerNN. Section 3 details the practical implementation, including the optimisation algorithm and convergence insights. Section 4 provides a statistical analysis using Rademacher complexity to show high-probability convergence to the minimal risk with explicit rates. Section 5 evaluates BKerNN through experiments on simulated and real data sets, comparing it with neural networks and kernel methods. Finally, Section 6 summarises the findings and suggests future research directions.

We use the following notations. For a positive integer m, we define $[m] := \{1, \dots, m\}$. For a d-dimensional vector α and $i \in [d]$, α_i denotes its i-th element. For a matrix A, $\operatorname{tr} A$ denotes its trace when A is square, A^{-1} its inverse when well defined, while $A_{i,j}$ the element in its i-th row and j-th column, and $A^\top$ its transpose. I_d is the $d \times d$ identity matrix. We use $\mathcal{S}^{d-1}$ to denote the unit sphere in $\mathbb{R}^d$ for $\|\cdot\|$ a generic norm and $\|\cdot\|_*$ its dual norm. The ℓ_2, ℓ_1, and ℓ_∞ norms are denoted as $\|\cdot\|_2$, $\|\cdot\|_1$, and $\|\cdot\|_\infty$ respectively. We use $O(\cdot)$ to denote the asymptotic behaviour of functions, indicating the order of growth. The set of probability measures on a given space S is denoted by $\mathcal{P}(S)$. A normal random variable is denoted as following the law $\mathcal{N}(\text{mean}, \text{variance})$. $\mathbb{1}$ is the indicator function. For two spaces S_1, S_2, $S_1^{S_2}$ is the set of functions from S_2 to S_1.

2 Neural Networks and Kernel Methods Fusion

Building on the limitations of current methods discussed in the introduction, we propose a novel architecture that integrates neural networks with kernel methods. This approach can be interpreted in two ways: as learning with a kernel that is itself learned during training, or as employing a one-hidden layer neural network where the weights from the input layer to the hidden layer are learned through gradient descent, while the weights and non-linearity from the hidden layer to the output are optimised explicitly. In this section, we introduce the custom function space we propose, revisit key properties of reproducing kernel Hilbert spaces (RKHS), and explore the connections between BKerNN model, kernel methods, and neural networks. Additionally, we present the various regularisation penalties we consider throughout our analysis.

2.1 Custom Space of Functions

We begin by considering the continuous setting, which mirrors the mean-field limit of over-parameterised one-hidden layer neural networks discussed in Section 1.

Definition 4 (Infinite-Width Function Space)**.** *Let*

$$\mathcal{F}_\infty := \left\{ f \mid f(\cdot) = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top \cdot)\, \mathrm{d}\mu(w), \quad \Omega_0(f) < \infty \right\},$$

where c is a constant in $\mathbb{R}$, $\mathcal{S}^{d-1}$ is the unit sphere for some norm $\|\cdot\|$ on $\mathbb{R}^d$ (typically either ℓ_2 or ℓ_1), $\mu \in \mathcal{P}(\mathcal{S}^{d-1})$ is a probability measure on $\mathcal{S}^{d-1}$, and $\forall w \in \mathcal{S}^{d-1}$, $g_w : \mathbb{R} \to \mathbb{R}$

belongs to a space of functions $\mathcal{H}$. We define $\mathcal{H}$ as $\{g : \mathbb{R} \to \mathbb{R} \mid g(0) = 0,\ g$ has a weak derivative $g',\ \int_{\mathbb{R}}(g')^2 < \infty\}$. $\mathcal{H}$ is a Hilbert space and a Sobolev space, with the inner product defined as $\langle \tilde{g}, g\rangle_{\mathcal{H}} = \int \tilde{g}'g'$. Note that g_w vary for each w and that

$$\Omega_0(f) := \inf_{c\in\mathbb{R},(g_w)_w\in\mathcal{H}^{\mathcal{S}^{d-1}},\mu\in\mathcal{P}(\mathcal{S}^{d-1})} \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}}\, \mathrm{d}\mu(w), \tag{4.2}$$

such that $f = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top \cdot)\, \mathrm{d}\mu(w)$, where $\|g_w\|_{\mathcal{H}}^2 := \int_{-\infty}^{+\infty}(g_w')^2$.

$\mathcal{F}_\infty$ is well-defined using a "variation norm" on couples (g, w) integrated w.r.t a Borel measure on $\mathcal{H} \times \mathcal{S}^{d-1}$. The details of the equivalence with the version presented above are available in Appendix A.1. It follows from similar arguments to those of Kurkova and Sanguineti [2001] and Bach [2024, Section 9.3.2].

The function space $\mathcal{F}_\infty$ is inspired by infinite-width single hidden layer neural networks: with the addition of the intercept c, each function in this space can be seen as the integral of a linear part w and a non-linearity g_w over some probability distribution, as in Equation (4.1) where the non-linearity is $\eta\sigma(\cdot)$. Thus here the activation functions are learnt.

The approximation of $\mathcal{F}_\infty$ with m particles can then be obtained as follows.

Definition 5 (Finite-Width Function Space)**.** *Let*

$$\mathcal{F}_m := \left\{ f \mid f(\cdot) = c + \frac{1}{m}\sum_{j=1}^{m} g_j(w_j^\top \cdot), w_j \in \mathcal{S}^{d-1}, g_j \in \mathcal{H}, c \in \mathbb{R} \right\}.$$

Remark that $\forall m \in \mathbb{N}^*, \mathcal{F}_m \subset \mathcal{F}_\infty$, by taking the discrete probability measure uniformly supported by the particles $w_1, \dots, w_m$.

We now consider regularised empirical risk minimisation starting with the basic penalty Ω_0. This penalty enforces the regularity of the function and, because we use penalisation with non-squared norms, limits the number of non-zero g_w. While this penalty is not specifically aimed at feature learning, by limiting the number of non-zero particles, it indirectly promotes feature learning to some extent. This serves as a starting point, and we introduce more targeted penalties in Section 2.5 with a stronger feature learning behaviour. For $f \in \mathcal{F}_m$ written as in Definition 5, the penalty simplifies to $\Omega_0(f) = \frac{1}{m}\sum_{j=1}^m \|g_j\|_{\mathcal{H}}$. The learning objective is thus defined as

$$\hat{f}_\lambda := \arg\min_{f\in\mathcal{F}}\ \widehat{\mathcal{R}}(f) + \lambda\Omega(f), \tag{4.3}$$

where $\lambda > 0$ is a regularisation parameter and Ω is currently Ω_0 from Equation (4.2). The function space $\mathcal{F}$ is either $\mathcal{F}_\infty$ or $\mathcal{F}_m$. For statistical analysis in Section 4, we consider $\mathcal{F}_\infty$, while in practice, we compute the estimator using $\mathcal{F}_m$ as discussed in Section 3. The rationale for using $\mathcal{F}_m$ and expecting the statistical properties of $\mathcal{F}_\infty$ is elaborated in Section 3.2.

In the continuous setting, Equation (4.3) corresponds to

$$\min_{c\in\mathbb{R},(g_w)_w\in\mathcal{H}^{\mathcal{S}^{d-1}},\mu\in\mathcal{P}(\mathcal{S}^{d-1})} \frac{1}{n}\sum_{i=1}^{n} \ell\left(y_i, c + \int_{\mathcal{S}^{d-1}} g_w(w^\top x_i)\mathrm{d}\mu(w)\right) + \lambda\int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}}\mathrm{d}\mu(w), \tag{4.4}$$

while in the m-particles setting, Equation (4.3) becomes

$$\min_{c\in\mathbb{R},w_1,\dots,w_m\in\mathcal{S}^{d-1},g_1,\dots,g_m\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\ell\Big(y_i, c+\frac{1}{m}\sum_{j=1}^{m} g_j(w_j^\top x_i)\Big)+\lambda\frac{1}{m}\sum_{j=1}^{m}\|g_j\|_{\mathcal{H}}. \tag{4.5}$$

2.2 Properties of Reproducing Kernel Hilbert Space $\mathcal{H}$ and Kernel k

In this subsection, we succinctly present some properties of reproducing kernel Hilbert spaces (RKHS) that are essential for our analysis. See Aronszajn [1950], Berlinet and Thomas-Agnan [2011] for an introduction to RKHS. Recall that we defined the Hilbert space $\mathcal{H}$ as

$$\mathcal{H} := \left\{ g : \mathbb{R}\to\mathbb{R} \mid g(0)=0, \int_{\mathbb{R}}(g')^2 < +\infty \right\},$$

with the inner product $\langle \tilde{g}, g\rangle = \int_{\mathbb{R}} \tilde{g}'g'$. This space is a reproducing kernel Hilbert space with the reproducing kernel $k^{(B)}(a,b) = (|a|+|b|-|a-b|)/2 = \min(|a|,|b|)\mathbb{1}_{ab>0}$. This kernel, which can be referred to as the "Brownian" kernel, corresponds to the covariance of the Brownian motion at times a and b [Mishura and Shevchenko, 2017, Chapter 3]. Consequently, we have the reproducing property

$$\forall g\in\mathcal{H}, \forall a\in\mathbb{R}, g(a) = \langle g, k_a^{(B)}\rangle,$$

where $k_a^{(B)} : b\in\mathbb{R}\to k^{(B)}(a,b)\in\mathbb{R}$. As a reproducing kernel, it is positive definite, meaning that for any $n\in\mathbb{N}$, $\alpha\in\mathbb{R}^n$, and $a\in\mathbb{R}^n$, we have $\sum_{i,j=1}^n \alpha_i k^{(B)}(a_i,a_j)\alpha_j \geq 0$. Additionally, we observe that $\|k_a^{(B)}\|_{\mathcal{H}}^2 = |a|$ and $\|k_a^{(B)} - k_b^{(B)}\|_{\mathcal{H}}^2 = |a-b|$. It is also noteworthy that by definition, the functions in $\mathcal{H}$ are necessarily continuous, in fact even 1/2-Hölder continuous as we see in Lemma 24.

The usual Hilbert/Sobolev space is $W^{1,2}(\mathbb{R})$ (also written as $\mathcal{H}^1$) with inner product equal to $\langle f,g\rangle = \int fg + \int f'g'$. This space is also an RKHS for the reproducing kernel $k^{\exp}(a,b) = \exp(-|a-b|)$ [see, e.g., Williams and Rasmussen, 2006]. We demonstrate that for optimisation purposes, the Brownian kernel is more advantageous due to its positive homogeneity in Section 3.2.

2.3 Characterisation of $\mathcal{F}_\infty$

In this subsection, we discuss the properties of the function space $\mathcal{F}_\infty$ and its relationship to other relevant spaces, such as the space of functions of one-hidden-layer neural networks presented in Section 1. We first present the following lemma.

Lemma 24 (Properties of Functions in $\mathcal{F}_\infty$)**.** *$\mathcal{F}_\infty$ is a vector space and* $\max(f(0),\Omega_0(f))$ *is a norm on $\mathcal{F}_\infty$. For $f\in\mathcal{F}_\infty$, the function f is 1/2-Hölder continuous with constant $\Omega_0(f)$, i.e., $|f(x)-f(x')|\leq\Omega_0(f)\sqrt{\|x-x'\|^*}$.*

The proof can be found in Appendix A.2.1. This lemma indicates that the space of functions $\mathcal{F}_\infty$ is contained within the space of 1/2-Hölder continuous functions. Recall that on a compact, all Lipschitz functions are Hölder continuous functions, indicating that the Hölder condition is less restrictive.

Now, we consider the relationship of $\mathcal{F}_\infty$ to other function spaces. Starting with the one-dimensional case, BKernNN reduces to kernel ridge regression with the Brownian kernel, which is also equivalent to learning with natural cubic splines [for an introduction to splines, see Wahba, 1990]. For multi-dimensional data, we use the Fourier decomposition of functions to bound the defining norms of function spaces, enabling us to make comparisons.

Lemma 25 (Functions Spaces Included in $\mathcal{F}_\infty$)**.** *Assume we only consider functions f with support on the ball centred at 0 with radius R and norm $\|\cdot\|^*$. Assume f has a Fourier transform and can be written using the inverse transform*[3] *as*

$$f(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \hat{f}(\omega) e^{i\omega^\top x} \,\mathrm{d}\omega,$$

then, it follows that

$$\Omega_0(f) \leq \frac{\sqrt{2R}}{(2\pi)^d} \int_{\mathbb{R}^d} |\hat{f}(\omega)| \cdot \|\omega\| \,\mathrm{d}\omega.$$

Hence, if $\int_{\mathbb{R}^d} |\hat{f}(\omega)| \cdot \|\omega\| \,\mathrm{d}\omega < \infty$, then f belongs to $\mathcal{F}_\infty$.

The proof is given in Appendix A.2.2. We remark that the condition $\int_{\mathbb{R}^d} |\hat{f}(\omega)| \cdot \|\omega\| \,\mathrm{d}\omega < \infty$ is a form of constraint on the regularity of the first-order derivatives.

According to Bach [2024, Section 9.3.4], the space of one-hidden-layer neural networks with ReLU activations in the mean-field limit with $\|w\|_2 = 1, |b| \leq R$ can be equipped with the Banach norm $\gamma_1(f) = \int |\eta| \mathrm{d}\mu(\eta, w, b)$, which can be then bounded as in Lemma 25 by

$$\frac{2}{(2\pi)^d R} \int_{\mathbb{R}^d} |\hat{f}(\omega)|(1 + 2R^2\|\omega\|_2^2) \,\mathrm{d}\omega. \tag{4.6}$$

Now remark that the bound on Ω_0 contains a factor $\|w\|$ in the integral, whereas for ReLU neural networks with γ_1 norm it is $1 + 2R^2\|w\|_2^2$. Hence, the constraint is stronger on the neural network space, no matter what norm $\|\cdot\|$ corresponds to, suggesting that $\mathcal{F}_\infty$ is a larger space of functions.

Also note that the bound from Equation (4.6) can be shown to be smaller (up to a constant) than the norm defining the Sobolev space penalising derivatives up to order $s := d/2 + 5/2$, which is $\int_{\mathbb{R}^d} |\hat{f}(\omega)|^2(1 + 2R^2\|\omega\|_2^2)^s \,\mathrm{d}\omega$. [Bach, 2024, Section 9.3.5]. This space is an RKHS because $s > d/2$, and the inequality on norms yields that the space of neural networks with ReLU activations equipped with the norm γ_1 (which is a Banach space) contains this RKHS. Another interesting remark is that if we used the norm $\gamma_2(f) = \int \eta^2 \mathrm{d}\mu(\eta, w, b)$ instead of γ_1, the space that we would obtain is an RKHS and is strictly included in the one defined by γ_1 [Bach, 2024, Section 9.5.1]

For neural networks with step activations, i.e., $\sigma(z) = \mathbb{1}_{z>0}$ in the mean-field limit, a similar bound holds for the γ_1 norm

$$\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} |\hat{f}(\omega)|(1 + R\|\omega\|_2) \,\mathrm{d}\omega. \tag{4.7}$$

This can be seen by applying the same proof technique as for Equation (4.6) from Bach [2024, Section 9.3.4][4]. Learning with this space is not practically feasible due to optimisation issues as the step function is incompatible with gradient descent methods. However, the bound from Equation (4.7) is similar to the one on Ω_0, hinting that $\mathcal{F}_\infty$ is comparably large even though learning is possible with $\mathcal{F}_\infty$, as discussed in Section 3. For a discussion on this topic, see Bach [2024, Chapter 9] and Liu et al. [2024].

[3] A sufficient condition is that both f and $\hat{f}$ belong to $L^1(\mathbb{R}^d)$.

[4] The only difference being that we use $e^{iu\|w\|_2} = 1 + \int_0^R i\|w\|_2 e^{it\|w\|_2} \mathbb{1}_{t \leq u} \mathrm{d}t$ instead of Taylor's formula, yielding $\gamma_1(x \to e^{i\omega^\top x}) \leq 1 + R\|w\|_2$.

2.4 Learning the Kernel or Training a Neural Network?

We first transform the optimisation problem before considering our setup from two different perspectives: one through kernel learning and the other through neural networks. To transform the optimisation problem, we use the representer theorem, a well-known result in RKHS that allows us to replace the optimisation over functions in the RKHS with optimisation over a finite weighted sum of the kernel at the data points.

Lemma 26 (Kernel Formulation of Finite-Width)**.** *Equation (4.5) is equivalent to*

$$\min_{w_1,\dots,w_m\in\mathbb{R}^d, c\in\mathbb{R}, \alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2}\alpha^\top K\alpha + \frac{\lambda}{2}\frac{1}{m}\sum_{j=1}^{m} \|w_j\|, \tag{4.8}$$

where $K = \frac{1}{m}\sum_{j=1}^{m} K^{(w_j)}$*, and* $K^{(w_j)} \in \mathbb{R}^{n\times n}$ *is the kernel matrix for kernel* $k^{(B)}$ *and projected data* $(w_j^\top x_1, \dots, w_j^\top x_n)$*, i.e.,* $K^{(w_j)}_{i,i'} = (|w_j^\top x_i| + |w_j^\top x_{i'}| - |w_j^\top (x_i - x_{i'})|)/2$*. Notice that there are no constraints on the particles* $(w_j)_{j\in[m]}$ *to belong to the unit sphere anymore.*

The proof is provided in Appendix A.3.1. This lemma shows that we only need to solve a problem over finite-dimensional quantities. For computational complexity considerations, see Section 3.1. We can view Equation (4.8) using kernels. In a classical kernel supervised learning problem with an unregularised intercept, we would have a fixed kernel matrix K and consider

$$\min_{c\in\mathbb{R}, \alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2}\alpha^\top K\alpha.$$

For infinitely many particles, the analogue of Lemma 26 is Lemma 27.

Lemma 27 (Kernel Formulation of Infinite-Width)**.** *Equation (4.4) is equivalent to:*

$$\min_{\nu\in\mathcal{P}(\mathbb{R}^d), c\in\mathbb{R}, \alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2}\alpha^\top K\alpha + \frac{\lambda}{2}\int_{\mathbb{R}^d} \|w\| \,\mathrm{d}\nu(w), \tag{4.9}$$

with $K = \int_{\mathbb{R}^d} K^{(w)} \,\mathrm{d}\nu(w)$ *and and* $K^{(w)} \in \mathbb{R}^{n\times n}$ *is the kernel matrix for kernel* $k^{(B)}$ *and data* $(w^\top x_1, \dots, w^\top x_n)$*. At the optimum, the support of* μ *from Equation (4.4) can be obtained from that of* ν *from Equation (4.9) by normalising all vectors.*

Notice that there is shift in spaces, as ν is a probability distribution on $\mathbb{R}^d$, whereas μ was a probability distribution on $\mathcal{S}^{d-1}$. The proof is provided in Appendix A.3.2.

2.4.1 Kernel Perspective

Lemma 26 shows that we are solving a regularised kernel ridge regression problem where the kernel $\frac{1}{m}\sum_{j=1}^{m}(|w_j^\top x| + |w_j^\top x'| - |w_j^\top (x - x')|)/2$ is also learnt through the weights $(w_j)_{j\in[m]}$, and the third term $\frac{\lambda}{2}\frac{1}{m}\sum_{j=1}^{m}\|w_j\|$ serves as a penalty to improve kernel learning.

The homogeneity of the kernel $k^{(B)}$ leads to well-behaved optimisation, as we discuss in Section 3.2 and see in Experiment 1 in Section 5.2. The kernel matrix K is indeed positively 1-homogeneous in the particles $(w_j)_{j\in[m]}$. If we had chosen $\mathcal{H}$ to be the RKHS corresponding to the exponential kernel (or the Gaussian kernel), we would have faced the challenge of learning the kernel $\sum_{j=1}^{m} e^{-|w_j^\top (x-x')|}$, which exhibits a complex and non-homogeneous dependency on the weights $(w_j)_{j\in[m]}$. By using the Brownian kernel instead

of the exponential kernel, we only slightly change the regularisation, regularising with $\int_{\mathbb{R}}(g')^2$ instead of $\int_{\mathbb{R}} g^2 + \int_{\mathbb{R}}(g')^2$ while making the optimisation more tractable.

Compared to multiple kernel learning, BKerNN offers notable advantages. MKL involves combining several predefined kernels, which is prone to overfitting as the number of kernels increases. Additionally, selecting the optimal kernel combination is challenging and often requires sophisticated algorithms. In contrast, BKerNN adapts the kernel through the learned weights $(w_j)_{j\in[m]}$, making the optimisation process simpler and more efficient, as discussed in Section 3.

2.4.2 Neural Network Perspective

Our architecture can also be interpreted as a special type of neural network with one hidden layer. Recall that $\mathcal{F}_\infty$ is inspired by neural networks as it involves linear components w followed by a non-linear part. In neural networks, this non-linear part is typically $\eta\sigma(\cdot)$, which we replaced with $g_w(\cdot) \in \mathcal{H}$ in our setting. The functions in $\mathcal{F}_m$ are expressed similarly with the number of particles m equivalent to the number of neurons in the hidden layer.

As we discuss in Section 3.1, we learn the weights $(w_j)_{j\in[m]}$ through gradient descent, while the functions $(g_j)_{j\in[m]}$ are learned explicitly, leveraging a closed-form solution. This approach resonates with the work of Marion and Berthier [2023] and Bietti et al. [2023]. Marion and Berthier [2023] examine a one-hidden layer neural network where the step-sizes for the inner layer are much smaller than those for the outer layer. They prove that the gradient flow converges to the optimum of the non-convex optimisation problem in a simple univariate setting and that the number of neurons does not need to be asymptotically large, which is a stronger result than the usual study of mean-field regimes or neural tangent kernel. Bietti et al. [2023] consider learning the link function in a non-parametric way infinitely faster than the low-rank projection subspace, which resonates with our method, although they focus Gaussian data.

We have also established that the function space $\mathcal{F}_\infty$ is more extensive than the space of neural networks with ReLU activations in Section 2.3. In Section 3.2, we demonstrate that this enlargement is compatible with efficient optimisation.

2.5 Other Penalties

We now present other penalties designed to achieve different effects. The three terms in Equation (4.8) correspond to the empirical risk, the standard penalty from KRR on the RKHS norm of the function, and an extra regularisation term on the learnt kernel weights. This additional term, $\frac{\lambda}{2m}\sum_{j=1}^m \|w_j\|$, originates from the penalty $\Omega_0(f)$ in Equation (4.2). However, we can explore other penalties on $w_1, \dots, w_m$ that induce various additional sparsity effects, even if they do not directly correspond to penalties on $f \in \mathcal{F}_m$. Let $W \in \mathbb{R}^{d\times m}$ be the matrix with $(w_1, \dots, w_m)$ as columns, denote by $W^{(a)}$ the a-th row of W, and let $W = USV^\top$ be its singular value decomposition, with S a diagonal matrix composed of $S_1, \dots, S_{\min(m,d)}$. Recall that ν is a probability distribution on $\mathbb{R}^d$.

1. **Basic penalty:** $\Omega_{\text{basic}}(w_1, \dots, w_m) = \frac{1}{2m}\sum_{j=1}^m \|w_j\|$, which we discussed in Section 2.1. In the continuous setting, it corresponds to $\frac{1}{2}\int_{\mathbb{R}^d} \|w\| \mathrm{d}\nu(w)$. This penalty, which does not target any specific pattern in the data-generating mechanism, is the one for which we provide theoretical results in Section 4. However, it does not work as well in practive as the following penalties.

2. **Variable penalty:** $\Omega_{\text{variable}}(w_1, \dots, w_m) = \frac{1}{2}\sum_{a=1}^{d}\left(\frac{1}{m}\sum_{j=1}^{m}(w_j)_a^2\right)^{1/2}$, which is also equal to $\frac{1}{2\sqrt{m}}\sum_{a=1}^{\min(m,d)}\|W^{(a)}\|_2$. This penalty, inspired by the group Lasso [Yuan and Lin, 2006], is designed for variable selection, pushing quantities $\|W^{(a)}\|_2$ towards zero, thus encouraging dependence on a few variables. In the continuous setting, it corresponds to $\frac{1}{2}\sum_{a=1}^{\min(m,d)}\left(\int_{\mathbb{R}^d}|w_a|^2 \mathrm{d}\nu(w)\right)^{1/2}$.

3. **Feature penalty:** $\Omega_{\text{feature}}(w_1, \dots, w_m) = \frac{1}{2}\operatorname{tr}\left(\left(\frac{1}{m}\sum_{j=1}^{m} w_j w_j^\top\right)^{1/2}\right)$, which is also equal to $\frac{1}{2}\sum_{a=1}^{\min(m,d)}\frac{S_a}{\sqrt{m}}$ and to the nuclear norm of W divided by $2\sqrt{m}$. It is used for feature learning as it is a convex relaxation of the rank, encouraging W to have low rank and thus dependence on only a few linear transformations of the data. Regularisation using the nuclear norm in the context of feature learning is well-established in the literature, as demonstrated by Argyriou et al. [2008]. It corresponds to $\frac{1}{2}\operatorname{tr}\left(\left(\int_{\mathbb{R}^d} ww^\top \mathrm{d}\nu(w)\right)^{1/2}\right)$ in the continuous setting.

4. **Concave variable penalty:** The concave version of the penalty for variable selection, $\Omega_{\text{concave variable}}(w_1, \dots, w_m) = \frac{1}{2s}\sum_{a=1}^{d}\log\left(1 + \frac{s}{\sqrt{m}}\|W^{(a)}\|_2\right)$, with $s \geq 0$. The appeal of the added concavity is discussed below. In the continuous setting, it corresponds to $\frac{1}{2s}\sum_{a=1}^{d}\log\left(1 + s\int_{\mathbb{R}^d}(w_a)^2 \mathrm{d}\nu(w)\right)^{1/2}$.

5. **Concave feature penalty:** The concave version of the penalty intended for feature learning, $\Omega_{\text{concave feature}}(w_1, \dots, w_m) = \frac{1}{2s}\sum_{a=1}^{\min(m,d)}\log\left(1 + \frac{s}{\sqrt{m}}S_a\right)$ for feature selection, with $s \geq 0$. The appeal of the added concavity is discussed below. In the continuous setting it corresponds to $\frac{1}{2s}\sum_{a=1}^{d}\log\left(1 + s\left(\left(\int_{\mathbb{R}^d} ww^\top \mathrm{d}\nu(w)\right)^{1/2}\right)_{a,a}\right)$.

The first penalty is convex in both ν and W, making it straightforward to optimise. The second and third penalties, while not convex in ν, are convex in W due to the presence of squared and square root terms on the components of W, easing optimisation in the m particles setting. The fourth and fifth penalties are neither convex in ν nor W, instead, they are concave in W. As s approaches zero, these penalties revert to their non-concave versions. Convex penalties, while easier to handle, can be detrimental by diminishing relevant variables or features to achieve sparsity. Mitigating this effect can involve retraining with the selected variables/features or employing concave penalties, which is the choice we made here. Although concave penalties are more complex to analyse, they often yield better performance because they drive the solution towards the boundary, promoting sparsity [Fan and Li, 2001, Bach et al., 2012]. We discuss the impact of the choice of regularisation in Experiment 3 in Section 5.3.

3 Computing the Estimator

In this section, we detail the process of computing the estimator for each of the penalties presented in Section 2.5. We then discuss the importance of the homogeneity of the Brownian kernel and how the optimisation with particles relates to the continuous setting.

3.1 Optimisation Procedure

In this section, we focus on the square loss $\ell(y, y') = \frac{1}{2}(y - y')^2$, which allows for explicit computations. However, the method can be extended to other loss functions using gradient-based techniques, [see Bach, 2024, Chapter 5]. Recalling Equation (4.8) and the

penalties described in Section 2.5, the optimisation problem we aim to solve is

$$\min_{w_1,\dots,w_m\in\mathbb{R}^d, c\in\mathbb{R}, \alpha\in\mathbb{R}^n} \frac{1}{2n}\|Y - K\alpha - c\mathbb{1}_n\|_2^2 + \frac{\lambda}{2}\alpha^\top K\alpha + \lambda\Omega_{\text{weights}}(w_1,\dots,w_m), \tag{4.10}$$

where $K = \frac{1}{m}\sum_{j=1}^m K^{(w_j)}$ and Ω_{weights} represents any of the penalties from Section 2.5.

To solve this problem, we alternate between minimisation with respect to α and c, which is done in closed-form, and minimisation with respect to $w_1,\dots,w_m$ which is done using one step of proximal gradient descent.

3.1.1 Fixed Particles $w_1,\dots,w_m$

When the weights $w_1,\dots,w_m$ are fixed, the kernel matrix K is also fixed, allowing us to find the solution for the constant c and the coefficients α in closed-form. By centring both the kernel matrix and the response Y, we transform the problem into a classical kernel ridge regression problem, for which explicit solutions are well-known.

Lemma 28 (Optimisation for Fixed Particles)**.** *For fixed $w_1,\dots,w_m$ and hence a fixed K, define*

$$G(w_1,\dots,w_m) := \min_{\alpha\in\mathbb{R}^n, c\in\mathbb{R}} \frac{1}{2n}\|Y - K\alpha - c\mathbb{1}_n\|_2^2 + \frac{\lambda}{2}\alpha^\top K\alpha.$$

The optimisation problem defining G is solved by

$$\alpha = (\tilde{K} + n\lambda I)^{-1}\tilde{Y} \quad \text{and} \quad c = \frac{\mathbb{1}^\top Y}{n} - \frac{\mathbb{1}^\top K\alpha}{n},$$

where $\tilde{K} := \Pi K\Pi$ and $\tilde{Y} := Y - \frac{\mathbb{1}\mathbb{1}^\top Y}{n}$, with $\Pi = I - \frac{\mathbb{1}\mathbb{1}^\top}{n}$ being the centring matrix. The objective value is then

$$G(w_1,\dots,w_m) = \frac{\lambda}{2}\tilde{Y}^\top(\tilde{K} + \lambda n I)^{-1}\tilde{Y}.$$

The proof is provided in Appendix A.4.1. Lemma 28 allows us to optimise α and c explicitly during the optimisation process. The complexity of this step is $O(n^3 + n^2 d)$, which can be challenging when the sample size n is large, a common drawback of kernel methods. However, techniques like the Nyström method [Drineas and Mahoney, 2005], which approximates the kernel matrix, can help mitigate this issue. Alternatively, we could use gradient descent techniques, but as shown in Marion and Berthier [2023], it may be beneficial to learn the weights from the hidden layer to the output layer (corresponding to learning $g_1,\dots,g_m$ and hence α) with a much larger step-size than the weights from the input layer to the hidden layer (corresponding to learning $w_1,\dots,w_m$). Learning α and c explicitly represents the limit of this two-timescale regime.

3.1.2 Proximal Step to Optimise the Weights $w_1,\dots,w_m$

Next, we focus on optimising $w_1,\dots,w_m$ while keeping c and α fixed. The goal is to solve

$$\min_{w_1,\dots,w_m\in\mathbb{R}^d} G(w_1,\dots,w_m) + \lambda\Omega_{\text{weights}}(w_1,\dots,w_m), \tag{4.11}$$

where the dependence on $(w_j)_{j\in[m]}$ in the first term is through the kernel matrix K. Note that G is convex in K but not in $w_1,\dots,w_m$. Additionally, G is differentiable

almost everywhere, except where $w_j^\top(x_i - x_{i'})$ for some $j \in [m], i \neq i' \in [n]$. However, standard practice assumes that these non-differentiabilities average out with many data points. Meanwhile, the penalties Ω_{weights} are not differentiable at certain fixed points, independently of the data, similarly to the Lasso penalty. Therefore, we use proximal gradient descent to solve Equation (4.11). With a step-size $\gamma > 0$, this involves minimising

$$\sum_{j=1}^{m} \frac{\partial G}{\partial w_j}(w^{\text{old}})^\top(w_j - w_j^{\text{old}}) + \frac{1}{2\gamma}\sum_{j=1}^{m}\|w_j - w_j^{\text{old}}\|_2^2 + \lambda\Omega_{\text{weights}}(w_1, \dots, w_m),$$

over $w_1, \dots, w_m \in \mathbb{R}^d$. This corresponds to the simultaneous proximal gradient descent steps $w_j \leftarrow \text{prox}_{\lambda\gamma\Omega}(w_j - \gamma\frac{\partial G}{\partial w_j})$. We therefore compute the gradient and the proximal operator. For the gradient, we have the following lemma.

Lemma 29 (Gradient of G)**.** *Let $j \in [m]$, then*

$$\frac{\partial G}{\partial w_j} = \frac{\lambda}{4}\frac{1}{m}\sum_{i,i'=1}^{n} z_i z_{i'} \operatorname{sign}(w_j^\top(x_i - x_{i'}))(x_i - x_{i'}),$$

where $z = (\tilde{K} + n\lambda I)^{-1}\tilde{Y}$.

The proof is in Appendix A.4.2. Note that G is not differentiable around 0, which is also the case of common activation functions in neural networks such as the ReLU, but this is not an issue in practice.

Next, we compute the proximal operator for the described penalties. Recall the definition of the proximal operator

$$\text{prox}_\Omega(W) = \underset{(u_1,\dots,u_m)\in\mathbb{R}^{d\times m}}{\arg\min}\ \frac{1}{2}\sum_{j=1}^{m}\|w_j - u_j\|_2^2 + \Omega(u_1, \dots, u_m).$$

We use $W \in \mathbb{R}^{d\times m}$ and $(w_1, \dots, w_m)$ interchangeably, with $W = USV^\top$ (SVD). We denote the rows of W by $W^{(a)}$ as before. The following lemma provides the proximal operators.

Lemma 30 (Proximal Operators)**.** *We describe the proximal operators.*

1. *For $\Omega_{\text{basic}}(W) = \frac{1}{2m}\sum_{j=1}^{m}\|w_j\|$, then $\left(\text{prox}_{\lambda\gamma\Omega}(W)\right)_j = \left(1 - \frac{\lambda\gamma}{2m}\frac{1}{\|w_j\|}\right)_+ w_j$.*
2. *For $\Omega_{\text{variable}}(W) = \frac{1}{2\sqrt{m}}\sum_{a=1}^{d}\|W^{(a)}\|_2$, $(\text{prox}_{\lambda\gamma\Omega}(W))^{(a)} = \left(1 - \frac{\lambda\gamma}{2\sqrt{m}}\frac{1}{\|W^{(a)}\|_2}\right)_+ W^{(a)}$.*
3. *For $\Omega_{\text{feature}}(W) = \frac{1}{2}\text{trace}\left(\left(\frac{1}{m}\sum_{j=1}^{m} w_j w_j^\top\right)^{1/2}\right)$, then we have $\text{prox}_{\lambda\gamma\Omega}(W) = U\tilde{S}V^\top$ with $\tilde{S} = \left(1 - \frac{\lambda\gamma}{2\sqrt{m|S|}}\right)_+ S$.*
4. *For $\Omega_{\text{concave variable}}(W) = \frac{1}{2s}\sum_{a=1}^{d}\log\left(1 + \frac{s}{\sqrt{m}}\|W^{(a)}\|_2\right)$, then with c obtained from $(\|W^{(a)}\|_2)_{a\in[d]}$ by an explicit (albeit lengthy) formula $(\text{prox}_{\lambda\gamma\Omega}(W))^{(a)} = cW^{(a)}$.*
5. *For $\Omega_{\text{concave feature}}(W) = \frac{1}{2s}\sum_{a=1}^{d}\log\left(1 + \frac{s}{\sqrt{m}}S_a\right)$, then with c which obtained from S by an explicit (albeit lengthy) formula $\text{prox}_{\lambda\gamma\Omega}(W) = U\tilde{S}V^\top$ with $\tilde{S} = cS$.*

The proof is in Appendix A.4.3. Each proximal step is easy to compute using the explicit formulas above, with complexities $O(md)$ for the basic, variable, and concave variable cases, and $O(md\min(m, d))$ for the feature and concave feature cases, due to the SVD computation.

3.1.3 Algorithm Pseudocode

We now have all the components necessary to provide the pseudocode of the proposed method BKerNN, specifically for the square loss. For other losses, the main difference is that α and c might not be solvable in closed-form and would need to be computed through alternative methods such as gradient descent.

Data: $X, Y, m, \lambda, \gamma, \Omega_{\text{weights}}$
Result: $w_1, \dots, w_m, c, \alpha$
$W = (w_1, \dots, w_m) \in \mathbb{R}^{d \times m} \leftarrow (\mathcal{N}(0, 1/d))^{d \times m}$;
for $i \in [n_{\text{iter}}]$ **do**
  Compute K;
  $\alpha \leftarrow (\tilde{K} + n\lambda I)^{-1}\tilde{Y}, c \leftarrow \frac{\mathbb{1}^\top Y}{n} - \frac{\mathbb{1}^\top}{n} K\alpha$;
  Compute $\frac{\partial G}{\partial W}$;
  $\gamma \leftarrow \gamma \times 1.5$;
  while $G(\text{prox}_{\lambda\gamma\Omega}(W - \gamma\frac{\partial G}{\partial W}) > G(W) - \gamma\frac{\partial G}{\partial W} \cdot G_\gamma(W) + \frac{\gamma}{2}\|G_\gamma(W)\|_2^2$ **do**
    $\gamma \leftarrow \gamma/2$;
  end
  $W \leftarrow \text{prox}_{\lambda\gamma\Omega}(W - \gamma\frac{\partial G}{\partial W})$;
end

Algorithm 3: BKerNN pseudocode.

To select the step-size γ for the proximal gradient descent step appropriately, we use a backtracking line search, assuming G is locally Lipschitz. Starting with the previous step-size, we multiply it by 1.5. If the backtracking condition is not satisfied, we divide γ by 2 and repeat. The backtracking condition is that $G(\text{prox}_{\lambda\gamma\Omega}(W - \gamma\frac{\partial G}{\partial W}))$ should be smaller than $G(W) - \gamma\frac{\partial G}{\partial W} \cdot G_\gamma(W) + \frac{\gamma}{2}\|G_\gamma(W)\|_2^2$, where $G_\gamma(W) = (W - \text{prox}_{\lambda\gamma\Omega}(W - \gamma\frac{\partial G}{\partial W}))/\gamma$. This method was taken from Beck [2017].

With the outputted $w_1, \dots, w_m$, c, and α from the algorithm, the estimator is the function $\hat{f}_\lambda$ defined as $\hat{f}_\lambda(x) = c + \sum_{i=1}^n \alpha_i \sum_{j=1}^m \frac{1}{m}(|w_j^\top x_i| + |w_j^\top x| - |w_j^\top (x - x_i)|)/2$. This formulation enables us to perform predictions on new data points and facilitates the extraction of meaningful linear features through the learned weights $(w_j)_{j\in[m]}$. Remark that we do not take into account the optimisation error in the rest of the chapter.

3.2 Convergence Guarantees on Optimisation Procedure

In this section, we discuss the convergence properties of the optimisation procedure. Although we do not provide a formal proof due to differentiability issues, we highlight the importance of the homogeneity of the Brownian kernel and present arguments suggesting the robustness of the optimisation process.

We aim to apply the insights from Chizat and Bach [2022] and Chizat and Bach [2018], which state that under certain assumptions, in the limit of infinitely many particles and an infinitely small step-size, gradient descent optimisation converges to the global optimum of the infinitely-many particles problem. Key assumptions include convexity with respect to the probability distribution in the and homogeneity of a specific quantity Ψ, which we define below. We reformulate our problem in line with Chizat and Bach [2022].

Considering the square loss with the basic penalty Ω_{basic}, the optimisation problem

with m particles from Equation (4.10) can be rewritten as

$$\min_{w_1,\dots,w_m\in\mathbb{R}^d}\left(\inf_{\alpha\in\mathbb{R}^n,c\in\mathbb{R}}\frac{1}{2n}\|Y-K\alpha-c\mathbb{1}_n\|_2^2+\frac{\lambda}{2}\alpha^\top K\alpha+\frac{\lambda}{2m}\sum_{j=1}^m\|w_j\|\right)$$
$$=\min_{w_1,\dots,w_m\in\mathbb{R}^d}\left(\frac{\lambda}{2}\tilde{Y}^\top\left(\tilde{K}+\lambda nI\right)^{-1}\tilde{Y}+\frac{\lambda}{2m}\sum_{j=1}^m\|w_j\|\right),$$

where $K=\frac{1}{m}\sum_{j=1}^m K^{(w_j)}$ is the final kernel matrix, $K^{(w)}\in\mathbb{R}^{n\times n}$ is the kernel matrix for kernel $k^{(B)}$ with projected data $(w^\top x_1,\dots,w^\top x_n)$, $\Pi=I_n-\mathbb{1}_n\mathbb{1}_n^\top$ is the centring matrix, while $\tilde{Y}=\Pi Y$ is the centred output, and $\tilde{K}=\Pi K\Pi$ is the centred kernel matrix. We solve this using proximal gradient descent. For the continuous case, the problem is

$$\min_{\nu\in\mathcal{P}(\mathbb{R}^d)}\left(\inf_{\alpha\in\mathbb{R}^n,c\in\mathbb{R}}\frac{1}{2n}\|Y-K\alpha-c\mathbb{1}_n\|_2^2+\frac{\lambda}{2}\alpha^\top K\alpha+\frac{\lambda}{2}\int_{\mathbb{R}^d}\|w\|\,\mathrm{d}\nu(w)\right)$$
$$=\min_{\nu\in\mathcal{P}(\mathbb{R}^d)}\left(\frac{\lambda}{2}\tilde{Y}^\top\left(\tilde{K}+\lambda nI\right)^{-1}\tilde{Y}+\frac{\lambda}{2}\int_{\mathbb{R}^d}\|w\|\,\mathrm{d}\nu(w)\right),$$

where $K=\int_{\mathbb{R}^d}K^{(w)}\,\mathrm{d}\nu(w)$ and ν is a probability measure on $\mathbb{R}^d$.

In both cases we minimise $F(\nu)$ (defined right below) over $\mathcal{P}(\mathbb{R}^d)$ for the continuous case and over $\mathcal{P}_n(\mathbb{R}^d)$, which is the set of probability distributions anchored at n points on $\mathbb{R}^d$, in the m-particles case. F is defined as

$$F(\nu):=Q\left(\int_{\mathbb{R}^d}\Psi(w)\,\mathrm{d}\nu(w)\right),$$

where $Q:\mathbb{R}^{n\times n}\times\mathbb{R}\to\mathbb{R}$, $Q(K,c')=\frac{\lambda}{2}\tilde{Y}^\top\left(\tilde{K}+\lambda nI\right)^{-1}\tilde{Y}+\frac{\lambda}{2}c'$, and $\Psi:\mathbb{R}^d\to\mathbb{R}^{n\times n}\times\mathbb{R}$, $\Psi(w)=(K^{(w)},\|w\|)$. Note that Ψ is indeed positively 1-homogeneous, as the necessary condition $\forall w\in\mathbb{R}^d,\forall\kappa>0$, $\Psi(\kappa w)=\kappa\Psi(w)$ is verified. Moreover, Q is convex in ν, indicating the optimisation is well-posed (while we perform computations for the square loss, we would also obtain a convex function for any convex loss).

Our method employs proximal gradient descent instead of basic gradient descent, which is acceptable as both methods approximate the differential equation arising in the infinitesimal step-size limit. Gradient descent is an explicit method, whereas proximal gradient descent combines implicit and explicit updates [Süli and Mayers, 2003]. Moreover, it allows to deal efficiently with the non-smoothness of the sparsity-inducing penalties (no additional cost and improved convergence behaviour).

While our framework aligns with that of Chizat and Bach [2022], we cannot directly apply their results due to the non-differentiability of Ψ around zero, a common issue in such analyses. Despite this, our setup meets the crucial assumptions of convexity in Q and the homogeneity of Ψ. See Experiment 1 in Section 5.2 for a numerical evaluation of the practical significance of the homogeneity assumption.

4 Statistical Analysis

In this section our objective is to obtain high-probability bounds on the expected risk of the BKernNN estimator to understand its generalisation capabilities. To achieve this, we bound the Gaussian complexity (a similar concept to the Rademacher complexity,) of the sets $\{f\in\mathcal{F}_\infty\mid\max(f(0),\Omega_0(f))\leq D\}$ for $D>0$. Recall that $\mathcal{F}_\infty$ is defined in

Definition 4. We begin by introducing the Gaussian complexity in Definition 6, followed by Lemma 31, which is used to simplify the quantities for subsequent bounding. We then bound the Gaussian complexities using two distinct techniques in Sections 4.1.1 and 4.1.2. The first technique yields a dimension-dependent bound with better complexity in sample size, while the second provides a dimension-independent bound. Finally, in Section 4.2, we derive the high-probability bound on the expected risk of BKerNN with explicit rates for data with subgaussian square-rooted norm, using an extension of McDiarmid's inequality from Meir and Zhang [2003], before detailing the data-dependent quantities of the rates. All of these results require few assumptions on the problem, and on the data-generating mechanism in particular.

While our method resembles multiple kernel learning, the theoretical results from MKL, which are often related to Rademacher chaos [e.g., Lanckriet et al., 2004, Ying and Campbell, 2010] are not directly applicable. This is because, in our approach, the learned weights are multi-dimensional and embedded within the kernel, rather than being simple scalar weights used to combine predefined kernels. Thus, the unique structure of our model requires different theoretical considerations.

4.1 Gaussian Complexity

Recall that the estimator BKerNN is defined as

$$\hat{f}_\lambda := \operatorname*{arg\,min}_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)) + \lambda\Omega(f),$$

where $\mathcal{F}$ is $\mathcal{F}_m := \{f \mid f(x) = c + \frac{1}{m}\sum_{j=1}^{m} g_j(w_j^\top x), w_j \in \mathcal{S}^{d-1}, g_j \in \mathcal{H}, c \in \mathbb{R}\}$ in practice for optimisation and $\mathcal{F}_\infty := \{f \mid f(x) = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top x)\,\mathrm{d}\mu(w), g_w \in \mathcal{H}, \mu \in \mathcal{P}(\mathcal{S}^{d-1}), c \in \mathbb{R}, \quad \Omega_0(f) < \infty\}$ for statistical analysis. Although we considered various penalties in Section 2.5, here we focus on $f \in \mathcal{F}_\infty$ with $\Omega(f) = \max(\Omega_0(f), |c|) = \max(\Omega_0(f), |f(0)|)$, where $\Omega_0(f)$ was defined as

$$\Omega_0(f) = \inf_{c\in\mathbb{R},\mu\in\mathcal{P}(\mathcal{S}^{d-1}),(g_w)_w\in\mathcal{H}^{\mathcal{S}^{d-1}}} \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}}\,\mathrm{d}\mu(w),$$

such that $f(\cdot) = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top\cdot)\mathrm{d}\mu(w)$ and corresponds to the basic penalty for Ω_{weights}. This is made possible through a well-defined mean-field limit; we leave the other penalties to future work.

We now introduce the concept of Gaussian complexity [for more details, see Bartlett and Mendelson, 2002].

Definition 6 (Gaussian Complexity)**.** *The Gaussian complexity of a set of functions $\mathcal{G}$ is defined as*

$$G_n(\mathcal{G}) := \mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\sup_{f\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i f(x_i)\right),$$

with ε a centred Gaussian vector with identity covariance matrix, and $\mathcal{D}_n := (x_1, \ldots, x_n)$ the data set consisting of i.i.d. samples drawn from the distribution of the random variable X. Note that it only contains the covariates, not the response.

We aim to bound $G_n(\{f \in \mathcal{F}_\infty \mid \Omega(f) \le D\})$ for some $D > 0$. The discussion on the Gaussian complexity of the space $\mathcal{F}_\infty$ would yield the same bounds if $\mathcal{F}_m$ were considered instead. However, since we demonstrated in Section 3.2 that optimisation in $\mathcal{F}_m$ and optimisation in $\mathcal{F}_\infty$ are closely related, we focus exclusively on $\mathcal{F}_\infty$ in this section.

First, we note that we can study the Gaussian complexity of a simpler class of functions, as indicated by the following lemma, which allows us to deal with the constant and remove the integral present in the definition of $\mathcal{F}_\infty$.

Lemma 31 (Simplification of Gaussian Complexity)**.** *Let $D > 0$. Then,*

$$G_n(\{f \in \mathcal{F}_\infty \mid \Omega(f) \leq D\}) \leq D\left(\frac{1}{\sqrt{n}} + G_n\right)$$

with $G_n := G_n(\{f \mid f(\cdot) = g(w^\top \cdot), \|g\|_{\mathcal{H}} \leq 1, w \in \mathcal{S}^{d-1}\})$.

The proof can be found in Appendix A.5.1. We now need to bound G_n, which we approach in two different ways. First, in Section 4.1.1, we use covering balls on the sphere $\mathcal{S}^{d-1}$, resulting in a dimension-dependent bound. Then, in Section 4.1.2, we approximate functions in $\mathcal{F}_\infty$ by Lipschitz functions, before using a covering argument, leading to a dimension-independent bound at the cost of worst dependency in the sample size n. With these bounds on G_n, we will derive results on the expected risk of the BKernNN estimator, providing explicit rates depending on the upper bounds of G_n, without exponential dependence on dimension.

4.1.1 Dimension-Dependent Bound

First, we note that the supremum over the functions g with $\|g\|_{\mathcal{H}} \leq 1$ can be obtained in closed-form (see Lemma 34 in Appendix A.5.2). This reduces the problem to considering the expectation of a supremum over the sphere, which we address using a covering of $\mathcal{S}^{d-1}$.

Theorem 7 (Dimension-Dependent Bound)**.** *We have*

$$G_n \leq 8\sqrt{\frac{d}{n}}\sqrt{\log(n+1)}\sqrt{\mathbb{E}_X\|X\|^*},$$

where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$. Recall that $\|\cdot\|$ defines the sphere $\mathcal{S}^{d-1}$.

The bound on the Gaussian complexity obtained here is dimension-dependent due to the covering of the unit ball in $\mathbb{R}^d$, but it has a favourable dependency on the sample size. For ease of exposition, we have replaced the original factor $\sqrt{\log(1+n/(2d))+1/(2d)+1}$ with $8\sqrt{\log(n+1)}$. Recall that $\|\cdot\|^* = \|\cdot\|_2$ for the ℓ_2 sphere and $\|\cdot\|^* = \|\cdot\|_\infty$ for the ℓ_1 sphere. Note that the dependency on the data distribution is explicit and can be easily bounded in different data-generating mechanisms, as discussed in Lemma 33 at the end of Section 4.

Proof of Theorem 7. First, using Lemma 34, we have

$$G_n = \mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\sup_{w\in\mathcal{S}^{d-1}} \frac{\sqrt{\varepsilon^\top K^{(w)}\varepsilon}}{n}\right),$$

where $K^{(w)}$ is the kernel matrix for the Brownian kernel with data $(w^\top x_1, \ldots, w^\top x_n)$.

We bound the supremum inside of the expectation using covering balls. Let $M \in \mathbb{N}^*$ and $\mathcal{W}^M$ be such that $\forall w \in \mathcal{S}^{d-1}, \exists \tilde{w} \in \mathcal{W}^M \subset \mathcal{S}^{d-1}$ such that $\|w - \tilde{w}\| \leq \zeta$, i.e., we have a ζ-covering of the sphere with its own norm in d dimensions. Fix $w \in \mathcal{S}^{d-1}$ and $\tilde{w}$ such

that $\|w - \tilde{w}\| \leq \zeta$. We then have

$$\begin{aligned}
|\sqrt{\varepsilon^\top K^{(w)}\varepsilon} - \sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon}| &= \left| \left\| \sum_{i=1}^n \varepsilon_i k_{w^\top x_i} \right\|_{\mathcal{H}} - \left\| \sum_{i=1}^n \varepsilon_i k_{\tilde{w}^\top x_i} \right\|_{\mathcal{H}} \right| \\
&\leq \left\| \sum_{i=1}^n \varepsilon_i (k_{w^\top x_i} - k_{\tilde{w}^\top x_i}) \right\|_{\mathcal{H}} \leq \sum_{i=1}^n |\varepsilon_i| \cdot \|k_{w^\top x_i} - k_{\tilde{w}^\top x_i}\|_{\mathcal{H}} \\
&= \sum_{i=1}^n |\varepsilon_i| \sqrt{|w^\top x_i - \tilde{w}^\top x_i|} \leq \sum_{i=1}^n |\varepsilon_i| \sqrt{\|w - \tilde{w}\| \|x_i\|^*} \\
&\leq \sqrt{\|w - \tilde{w}\|} \sum_{i=1}^n |\varepsilon_i| \sqrt{\|x_i\|^*} \leq \zeta^{1/2} \sum_{i=1}^n |\varepsilon_i| \sqrt{\|x_i\|^*}.
\end{aligned}$$

Next, we get

$$\sqrt{\varepsilon^\top K^{(w)}\varepsilon} = \sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon} + \sqrt{\varepsilon^\top K^{(w)}\varepsilon} - \sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon} \leq \sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon} + \zeta^{1/2} \sum_{i=1}^n |\varepsilon_i| \sqrt{\|x_i\|^*}.$$

Taking the supremum and dividing by the sample size n,

$$\sup_{w \in \mathcal{S}^{d-1}} \frac{\sqrt{\varepsilon^\top K^{(w)}\varepsilon}}{n} \leq \sup_{\tilde{w} \in \mathcal{W}^M} \frac{\sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon}}{n} + \zeta^{1/2} \sum_{i=1}^n |\varepsilon_i| \sqrt{\|x_i\|^*}. \tag{4.12}$$

Considering the expectation over ε of Equation (4.12), we get

$$\mathbb{E}_\varepsilon \left(\sup_{w \in \mathcal{S}^{d-1}} \frac{\sqrt{\varepsilon^\top K^{(w)}\varepsilon}}{n} \right) \leq \mathbb{E}_\varepsilon \left(\sup_{\tilde{w} \in \mathcal{W}^M} \frac{\sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon}}{n} \right) + \zeta^{1/2} \mathbb{E}_\varepsilon \left(\frac{1}{n} \sum_{i=1}^n |\varepsilon_i| \sqrt{\|x_i\|^*} \right).$$

We now handle $\mathbb{E}_\varepsilon \left(\sup_{\tilde{w} \in \mathcal{W}^M} \frac{\sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon}}{n} \right)$ using standard concentration tools for supremum of infinitely many random variables. Consider $t > 0$, then

$$\begin{aligned}
\mathbb{E}_\varepsilon \left(\sup_{\tilde{w} \in \mathcal{W}^M} \sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon} \right) &\leq \sqrt{\mathbb{E}_\varepsilon \left(\sup_{\tilde{w} \in \mathcal{W}^M} \varepsilon^\top K^{(\tilde{w})}\varepsilon \right)} \\
&\leq \sqrt{\frac{1}{t} \log \left(\mathbb{E}_\varepsilon \left(e^{t \sup_{\tilde{w} \in \mathcal{W}^M} \varepsilon^\top K^{(\tilde{w})}\varepsilon} \right) \right)} \\
&= \sqrt{\frac{1}{t} \log \left(\mathbb{E}_\varepsilon \left(\sup_{\tilde{w} \in \mathcal{W}^M} e^{t\varepsilon^\top K^{(\tilde{w})}\varepsilon} \right) \right)} \\
&\leq \sqrt{\frac{1}{t} \log \left(\mathbb{E}_\varepsilon \left(\sum_{\tilde{w} \in \mathcal{W}^M} e^{t\varepsilon^\top K^{(\tilde{w})}\varepsilon} \right) \right)} \\
&= \sqrt{\frac{1}{t} \log \left(\sum_{\tilde{w} \in \mathcal{W}^M} \mathbb{E}_\varepsilon \left(e^{t\varepsilon^\top K^{(\tilde{w})}\varepsilon} \right) \right)}.
\end{aligned}$$

Fix $\tilde{w} \in \mathcal{W}^M$ and consider $\mathbb{E}_\varepsilon \left(e^{t\varepsilon^\top K^{(\tilde{w})}\varepsilon} \right)$. Diagonalising $K^{(\tilde{w})}$ to $U_{\tilde{w}} D_{\tilde{w}} U_{\tilde{w}}^\top$, we have that $U_{\tilde{w}}^\top \varepsilon$ is still a Gaussian vector with identity covariance matrix. When t is small enough,

i.e., $\forall i \in [n], 2t(D_{\tilde{w}})_i < 1$, or $t < \frac{1}{2\max_i(D_{\tilde{w}})_i}$,

$$\begin{aligned}\mathbb{E}_\varepsilon\left(e^{t\varepsilon^\top K^{(\tilde{w})}\varepsilon}\right) &= \mathbb{E}_\varepsilon\left(e^{t\sum_{i=1}^n (D_{\tilde{w}})_i\varepsilon_i^2}\right) = \prod_{i=1}^n \mathbb{E}_\varepsilon(e^{t(D_{\tilde{w}})_i\varepsilon_i^2}) \\ &= \prod_{i=1}^n \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} e^{(t(D_{\tilde{w}})_i - 1/2)\varepsilon_i^2}\,\mathrm{d}\varepsilon_i \\ &= \prod_{i=1}^n \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} e^{(2t(D_{\tilde{w}})_i - 1)\frac{\varepsilon_i^2}{2}} = \prod_{i=1}^n (1-2t(D_{\tilde{w}})_i)^{-1/2}.\end{aligned}$$

Re-injecting this, we obtain

$$\begin{aligned}\log\left(\mathbb{E}_\varepsilon\left(e^{t\varepsilon^\top K^{(\tilde{w})}\varepsilon}\right)\right) &= \log\left(\prod_{i=1}^n (1-2t(D_{\tilde{w}})_i)^{-1/2}\right) \\ &\leq \frac{-1}{2}\sum_{i=1}^n \log(1-2t(D_{\tilde{w}})_i).\end{aligned}$$

To bound this further, take $t \leq \frac{1}{4\max((D_{\tilde{w}})_i)}$, which implies both $2t(D_{\tilde{w}})_i < 1/2$ and $-\log(1-2t(D_{\tilde{w}})_i) \leq 4t(D_{\tilde{w}})_i$, leading to

$$\log\left(\mathbb{E}_\varepsilon\left(e^{t\varepsilon^\top K^{(\tilde{w})}\varepsilon}\right)\right) \leq 2t\sum_{i=1}^n (D_{\tilde{w}})_i \leq 2t\,\mathrm{tr}(K^{(\tilde{w})}) \leq 2t\sum_{i=1}^n \|x_i\|^*.$$

Taking $t \leq \min_{\tilde{w}\in\mathcal{W}^M} \frac{1}{4\max((D_{\tilde{w}})_i)}$, we obtain

$$\begin{aligned}\mathbb{E}_\varepsilon\left(\sup_{\tilde{w}\in\mathcal{W}^M} \frac{\sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon}}{n}\right) &\leq \frac{1}{n}\sqrt{\frac{1}{t}\log\left(Me^{2t\sum_{i=1}^n \|x_i\|^*}\right)} \\ &\leq \frac{1}{n}\sqrt{\frac{1}{t}\left(\log M + 2t\sum_{i=1}^n \|x_i\|^*\right)}.\end{aligned}$$

Taking $t = \frac{1}{4\sum_{i=1}^n \|x_i\|^*}$, which fulfils the previously required conditions, we get

$$\mathbb{E}_\varepsilon\left(\sup_{\tilde{w}\in\mathcal{W}^M} \frac{\sqrt{\varepsilon^\top K^{(\tilde{w})}\varepsilon}}{n}\right) \leq \frac{1}{\sqrt{n}}\sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}\sqrt{4\log M + 2}.$$

In the end, we obtain

$$\mathbb{E}_\varepsilon\left(\sup_{w\in\mathcal{S}^{d-1}} \frac{\sqrt{\varepsilon^\top K^{(w)}\varepsilon}}{n}\right) \leq \frac{1}{\sqrt{n}}\sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}\sqrt{4\log M + 2} + \zeta^{1/2}\sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}},$$

where we have used $\mathbb{E}_\varepsilon|\varepsilon_i| \leq \sqrt{\mathbb{E}_\varepsilon(\varepsilon_i)^2} = 1$ and $\frac{\sum_{i=1}^n \sqrt{\|x_i\|^*}}{n} \leq \sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}$.

We know that $M \leq (1+2/\zeta)^d$ [Wainwright, 2019, Lemma 5.7], yielding

$$\begin{aligned}
\mathbb{E}_\varepsilon\left(\sup_{w\in\mathcal{S}^{d-1}}\frac{\sqrt{\varepsilon^\top K^{(w)}\varepsilon}}{n}\right) &\leq \sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}\left(\frac{\sqrt{4d\log(1+\frac{2}{\zeta})+2}}{\sqrt{n}}+\zeta^{1/2}\right)\\
&\leq \sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}\left(\frac{\sqrt{4d\log(1+\frac{n}{2d})+2}}{\sqrt{n}}+\sqrt{\frac{4d}{n}}\right)\\
&\leq 2\sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}\frac{\sqrt{d}}{\sqrt{n}}\left(\sqrt{\log\left(1+\frac{n}{2d}\right)+\frac{1}{2d}}+1\right)\\
&\leq 4\sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}\frac{\sqrt{d}}{\sqrt{n}}\left(\sqrt{\log\left(1+\frac{n}{2d}\right)}+1\right)\\
&\leq 8\sqrt{\frac{\sum_{i=1}^n \|x_i\|^*}{n}}\frac{\sqrt{d}}{\sqrt{n}}\sqrt{\log(n+1)},
\end{aligned}$$

where to get the second line, we took $\zeta = 4d/n$. By taking the expectation over the data set $\mathcal{D}_n$, since $\mathbb{E}_{\mathcal{D}_n}(\sqrt{n^{-1}\sum_{i=1}^n \|x_i\|^*}) \leq \sqrt{\mathbb{E}(\|X\|^*)}$, we have the desired result. □

4.1.2 Dimension-Independent Bound

We now bound the Gaussian complexity with a quantity that does not explicitly depend on the dimension of the data. Recall that we aim to bound

$$G_n = \mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\sup_{\|g\|_\mathcal{H}\leq 1, w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^n \varepsilon_i g(w^\top x_i)\right),$$

where ε is a centred Gaussian vector with an identity covariance matrix. First, recall that the functions in $\mathcal{H}$ with norm bounded by 1 are not Lipschitz functions but are instead 1/2-Hölder functions (Lemma 24). Specifically, let $g \in \mathcal{H}$, $\|g\|_\mathcal{H} \leq 1$, then for any $a, b \in \mathbb{R}$, we have $|g(a) - g(b)| \leq \|k_a - k_b\|_\mathcal{H} = \sqrt{|a-b|}$.

An interesting result for a fixed 1-Lipschitz function h is that we can apply the contraction principle [Bach, 2024, Proposition 4.3] to the Rademacher complexity. Informally, this yields

$$\mathbb{E}_\varepsilon\left(\sup_{w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^n \varepsilon_i h(w^\top x_i)\right) \leq \mathbb{E}_\varepsilon\left(\sup_{w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^n \varepsilon_i w^\top x_i\right),$$

where exceptionally ε is composed of independent Rademacher variables. The supremum in the second term can then be taken explicitly. We will make use of this idea by first approximating the functions in the unit ball of $\mathcal{H}$ with Lipschitz functions, before using Slepian's lemma [Ledoux and Talagrand, 1991, Corollary 3.14] to obtain similar results on the Gaussian complexity.

Lemma 32 (Lipschitz Approximation)**.** *Let $g \in \mathcal{H}$ with $\|g\|_\mathcal{H} \leq 1$, and let $\zeta > 0$. There exists a $(1/\zeta)$-Lipschitz function $g_\zeta : \mathbb{R} \to \mathbb{R}$ with $g_\zeta(0) = 0$ such that $\|g - g_\zeta\|_\infty \leq \zeta$.*

The proof can be found in Appendix A.5.3. This lemma indicates that we can approximate functions in the unit ball of the RKHS $\mathcal{H}$ up to any precision in the infinite norm by Lipschitz functions with a Lipschitz constant equal to the inverse of the precision.

Theorem 8 (Dimension-Independent Bound)**.** *If $\mathcal{S}^{d-1}$ is the ℓ_1 or the ℓ_2 sphere, then*

$$G_n \leq \frac{6}{n^{1/6}} \left((\log 2d)^{1/4} \mathbb{1}_{*=\infty} + \mathbb{1}_{*=2} \right) \left(\mathbb{E}_{\mathcal{D}_n} \left(\max_{i \in [n]} (\|X_i\|^*)^2 \right) \right)^{1/4}.$$

Recall that in the ℓ_1 sphere case, $\|\cdot\|^* = \|\cdot\|_\infty$, and in the ℓ_2 case $\|\cdot\|^* = \|\cdot\|_2$. Here, we obtain a bound on the Gaussian complexity that depends only mildly on the data dimension d, either not at all in the case of the ℓ_2 sphere or logarithmically for the ℓ_1 sphere. This means that the estimator BKernNN can be effectively used in high-dimensional settings, where the data dimension may be exponentially large relative to the sample size. This improved dependency on the dimension d comes at the cost of a worse dependency on the sample size n compared to Theorem 7. Note also that there can be an implicit dependency on the dimension through the data distribution, which we discuss in Lemma 33 at the end of Section 4 under different data-generating mechanisms.

Proof of Theorem 8. By applying Lemma 32, we have for any $\zeta_1 > 0$

$$\begin{aligned}
\hat{G}_n &:= \mathbb{E}_\varepsilon \left(\sup_{w \in \mathcal{S}^{d-1}, \|g\|_{\mathcal{H}} \leq 1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i g(w^\top x_i) \right) \\
&= \mathbb{E}_\varepsilon \left(\sup_{w \in \mathcal{S}^{d-1}, \|g\|_{\mathcal{H}} \leq 1} \frac{1}{n} \sum_{i=1}^n \varepsilon_i \left(g_{\zeta_1}(w^\top x_i) + g(w^\top x_i) - g_{\zeta_1}(w^\top x_i) \right) \right) \\
&\leq \mathbb{E}_\varepsilon \left(\sup_{w \in \mathcal{S}^{d-1}, \|g\|_{\mathcal{H}} \leq 1} \left(\frac{1}{n} \sum_{i=1}^n \varepsilon_i g_{\zeta_1}(w^\top x_i) + \|g - g_{\zeta_1}\|_\infty \right) \right).
\end{aligned}$$

We can then change the supremum over the unit ball of $\mathcal{H}$ to a supremum over Lipschitz functions

$$\begin{aligned}
\hat{G}_n &\leq \mathbb{E}_\varepsilon \left(\sup_{w \in \mathcal{S}^{d-1},\ g_{\zeta_1}(1/\zeta_1)-\text{Lip},\ g_{\zeta_1}(0)=0} \frac{1}{n} \sum_{i=1}^n \varepsilon_i g_{\zeta_1}(w^\top x_i) \right) + \zeta_1 \\
&= \frac{1}{\zeta_1} \mathbb{E}_\varepsilon \left(\sup_{h\ 1-\text{Lip},\ h(0)=0} \sup_{w \in \mathcal{S}^{d-1}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i h(w^\top x_i) \right) + \zeta_1 \\
&= 2 \sqrt{\mathbb{E}_\varepsilon \left(\sup_{h\ 1-\text{Lip},\ h(0)=0} \sup_{w \in \mathcal{S}^{d-1}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i h(w^\top x_i) \right)},
\end{aligned}$$

by choosing the best ζ_1. Technically, we can restrict ourselves to the following class of function: $\mathcal{F}_{1-\text{Lip}} := \{h : [-\max_{i\in[n]} \|x_i\|^*, \max_{i\in[n]} \|x_i\|^*] \to \mathbb{R} \mid h(0) = 0, h \text{ is } 1-\text{Lipschitz}\}$.

We then use a covering argument. To cover $\mathcal{F}_{1-\text{Lip}}$ up to precision $\zeta_2 > 0$ in $\|\cdot\|_\infty$ norm with M functions from $\mathcal{F}_{1-\text{Lip}}$, one needs $M \leq \left(\frac{8 \max_{i\in[n]} \|x_i\|^*}{\zeta_2} + 1 \right) 2^{\frac{4 \max_{i\in[n]} \|x_i\|^*}{\zeta_2}}$ [Luxburg and Bousquet, 2004, Theorem 17]. Let $h_1, \dots h_M$ be such a covering. This yields

that

$$\hat{G}_n \leq 2\sqrt{\mathbb{E}_\varepsilon\left(\sup_{h\in\mathcal{F}_{1-\mathrm{Lip}}}\sup_{w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^n \varepsilon_i h(w^\top x_i)\right)}$$
$$\leq 2\sqrt{\mathbb{E}_\varepsilon\left(\sup_{h\in\{h_1,\ldots,h_M\}}\sup_{w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i} \varepsilon_i h(w^\top x_i)\right)+\zeta_2},$$

by proceeding as with the covering of the unit ball of $\mathcal{H}$.

We then use Lemma 35 to bound the expectation on the supremum of the finite set of Lipschitz functions, which is inspired by Bartlett and Mendelson [2002]. This yields

$$\mathbb{E}_\varepsilon\left(\sup_{h\in\{h_1,\ldots,h_M\},w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^n \varepsilon_i h(w^\top x_i)\right)$$
$$\leq \mathbb{E}_\varepsilon\left(\left\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\right\|^* + \sqrt{8\frac{\sum_{i=1}^n(\|x_i\|^*)^2}{n^2}}\sqrt{2\log M}\right). \tag{4.13}$$

We then consider each term of Equation (4.13) separately, while also taking expectation with regards to the data set. For the second term, using the bound on M [Luxburg and Bousquet, 2004, Theorem 17] and basic inequalities to simplify the term, we have

$$\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\sqrt{8\frac{\sum_{i=1}^n(\|x_i\|^*)^2}{n^2}}\sqrt{2\log M}\right)$$
$$\leq \mathbb{E}_{\mathcal{D}_n}\left(\sqrt{8\frac{\sum_{i=1}^n(\|x_i\|^*)^2}{n^2}}\sqrt{\frac{4\max_{i\in[n]}\|x_i\|^*}{\zeta_2}\log 2+\log\left(\frac{8\max_{i\in[n]}\|x_i\|^*}{\zeta_2}+1\right)}\right)$$
$$\leq \mathbb{E}_{\mathcal{D}_n}\left(8\sqrt{\frac{\sum_{i=1}^n(\|x_i\|^*)^2}{n^2}}\sqrt{\frac{\max_{i\in[n]}\|x_i\|^*}{\zeta_2}}\right)$$
$$\leq \frac{8}{\sqrt{n}}\frac{1}{\sqrt{\zeta_2}}\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^{3/2}\right) \leq \frac{8}{\sqrt{n}}\frac{1}{\sqrt{\zeta_2}}\left(\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)\right)^{3/4}.$$

$$\frac{G_n^2}{4} \leq \mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\left\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\right\|^*\right)+\frac{8}{\sqrt{n}}\frac{1}{\sqrt{\zeta_2}}\left(\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)\right)^{3/4}+\zeta_2$$
$$\leq \mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\left\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\right\|^*\right)+2\left(\frac{8}{\sqrt{n}}\left(\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)\right)^{3/4}\right)^{2/3}$$
$$\leq \mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\left\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\right\|^*\right)+2\frac{4}{n^{1/3}}\sqrt{\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)},$$

by taking $\zeta_2^{3/2}=\frac{8}{\sqrt{n}}\left(\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)\right)^{3/4}$ in the second line.

Now for the first term from Equation (4.13) which we have to deal with still, consider

first the case $\|\cdot\|^* = \|\cdot\|_2$ then,

$$\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\|_2\right) \leq \sqrt{\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\|_2^2\right)}$$
$$= \frac{\sqrt{2}}{n}\sqrt{\mathbb{E}_{\mathcal{D}_n}\left(\sum_{i=1}^n \|x_i\|_2^2\right)} = \frac{\sqrt{2}}{\sqrt{n}}\sqrt{\mathbb{E}_X\left(\|X\|_2^2\right)}.$$

In the other case where $\|\cdot\|^* = \|\cdot\|_\infty$, we can use Boucheron et al. [2013, Theorem 2.5], as for a fixed data set $\mathcal{D}_n$, $\sum_{i=1}^n \varepsilon_i s(x_i)_a$ is a centred Gaussian vector with variance equal to $\sum_{i=1}^n ((x_i)_a)^2$ which is smaller than $\max_{a\in[d]}\sum_{i=1}^n ((x_i)_a)^2$. This yields that

$$\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\|_\infty\right) = \frac{\sqrt{2}}{n}\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\max_{a\in[d]}|\sum_{i=1}^n \varepsilon_i (x_i)_a|\right)$$
$$= \frac{\sqrt{2}}{n}\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\max_{a\in[d],s\in\{-1,1\}}\sum_{i=1}^n \varepsilon_i s(x_i)_a\right)$$
$$\leq \frac{\sqrt{2}}{n}\mathbb{E}_{\mathcal{D}_n}\left(\max_{a\in[d]}\sqrt{2\sum_{i=1}^n ((x_i)_a)^2\log(2d)}\right).$$

We then have

$$\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\|\frac{\sqrt{2}}{n}\sum_{i=1}^n \varepsilon_i x_i\|_\infty\right) \leq \frac{2}{n}\sqrt{\log 2d}\mathbb{E}_{\mathcal{D}_n}\left(\sqrt{\max_{a\in[d]}\sum_{i=1}^n ((x_i)_a)^2}\right)$$
$$\leq \frac{2}{n}\sqrt{\log 2d}\mathbb{E}_{\mathcal{D}_n}\left(\sqrt{\sum_{i=1}^n \max_{a\in[d]}((x_i)_a)^2}\right)$$
$$\leq \frac{2}{n}\sqrt{\log 2d}\mathbb{E}_{\mathcal{D}_n}\left(\sqrt{\sum_{i=1}^n \|x_i\|_\infty^2}\right)$$
$$\leq \frac{2}{\sqrt{n}}\sqrt{\log 2d}\sqrt{\mathbb{E}_X\left(\|X\|_\infty^2\right)}.$$

This yields that the last term of Equation (4.13) can be bounded

$$\frac{G_n^2}{4} \leq \left(\frac{2}{\sqrt{n}}\sqrt{\log 2d}\mathbb{1}_{*=\infty} + \frac{\sqrt{2}}{\sqrt{n}}\mathbb{1}_{*=2}\right)\sqrt{\mathbb{E}_X((\|X\|^*)^2)} + 2\frac{4}{n^{1/3}}\sqrt{\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)}$$
$$\leq \left(\sqrt{\log 2d}\mathbb{1}_{*=\infty} + \mathbb{1}_{*=2}\right)\frac{8}{n^{1/3}}\sqrt{\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)},$$

hence

$$G_n \leq (\log 2d\mathbb{1}_{*=\infty} + \mathbb{1}_{*=2})^{1/4}\frac{6}{n^{1/6}}\left(\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}(\|x_i\|^*)^2\right)\right)^{1/4},$$

which concludes the proof. □

4.2 Bound on Expected Risk of Regularised Estimator

We now use the bounds on the Gaussian complexity we have obtained in Section 4.1 to derive a bound on the expected risk of BKerNN. We show that, with explicit rates, the expected risk of our estimator converges with high-probability to that of the minimiser for data with subgaussian norms, which includes both bounded data and data with subgaussian components. First, we provide a definition of subgaussian real variables, as given by Vershynin [2018].

Definition 7 (Subgaussian Variables)**.** *Let Z be a real-valued (not necessarily centred) random variable. Z is subgaussian with variance proxy σ^2 if and only if*

$$\forall t > 0, \max\left(\mathbb{P}(Z \geq t), \mathbb{P}(Z \leq -t)\right) \leq e^{-\frac{t^2}{2\sigma^2}}.$$

We now present the main theoretical result of the chapter.

Theorem 9 (Bound on Expected Risk with High-Probability)**.** *Let the estimator function be $\hat{f}_\lambda := \arg\min_{f \in \mathcal{F}_\infty} \widehat{\mathcal{R}}(f) + \lambda\Omega(f)$. Assume the following:*

1. ***Well-specified model**: The minimiser $f^* := \arg\min_{f \in \mathcal{F}_\infty, \Omega(f) < +\infty} \mathcal{R}(f)$ exists.*
2. ***Convexity of the loss**: For any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $f \in \mathcal{F}_\infty \to \ell(y, f(x))$ is convex.*
3. ***Lipschitz condition**: The loss ℓ is L-Lipschitz in its second (bounded) argument, i.e., $\forall y \in \mathcal{Y}, a \in \{f(x) \mid x \in \mathcal{X},\ f \in \mathcal{F}_\infty,\ \Omega(f) \leq 2\Omega(f^*)\}, a \to \ell(y, a)$ is L-Lipschitz.*
4. ***Data distribution**: The data set $(x_i, y_i)_{i \in [n]}$ consists of i.i.d. samples of the random variable (X, Y) where $1 + \sqrt{\|X\|^*}$ is subgaussian with variance proxy σ^2.*

Then, for any $\delta \in (0, 1)$, with probability larger than $1 - \delta$, for $\lambda = 12L\left(\frac{1}{\sqrt{n}} + G_n\right) + \frac{288L\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}$,

$$\mathcal{R}(\hat{f}_\lambda) \leq \mathcal{R}(f^*) + 24\Omega(f^*)L\left(\frac{1}{\sqrt{n}} + G_n + \frac{24\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}\right).$$

With the bounds on G_n from Theorem 7 and Theorem 8, recall that if $\|\cdot\|$ is either $\|\cdot\|_2$ or $\|\cdot\|_1$, we have

$$G_n \leq \min\left(\frac{6}{n^{1/6}}\left((\log 2d)^{1/4}\mathbb{1}_{*=\infty} + \mathbb{1}_{*=2}\right)\left(\mathbb{E}_{\mathcal{D}_n}\left(\max_{i \in [n]}(\|X_i\|^*)^2\right)\right)^{1/4}, 8\sqrt{\frac{d}{n}}\sqrt{\log(n+1)}\sqrt{\mathbb{E}_X\|X\|^*}\right).$$

Proof of Theorem 9. This proof is primarily based on Bach [2024, Proposition 4.7].

Let f_λ^* be a minimiser of $\mathcal{R}_\lambda := \mathcal{R} + \lambda\Omega$ over $\mathcal{F}_\infty$. Consider the set $\mathcal{C}_\tau := \{f \in \mathcal{F}_\infty \mid \mathcal{R}_\lambda(f) - \mathcal{R}_\lambda(f_\lambda^*) \leq \tau\}$ for some $\tau > 0$ that will be chosen later. $\mathcal{C}_\tau$ is a convex set by the convexity assumption on the loss ℓ.

First, we show that $\mathcal{C}_\tau$ is included in the set $\mathcal{B}_\tau := \{f \in \mathcal{F}_\infty \mid \Omega(f) \leq \Omega(f^*) + \tau/\lambda\}$. This inclusion follows from the optimality of f^* and f_λ^*. Let $f \in \mathcal{C}_\tau$, then

$$\mathcal{R}_\lambda(f) \leq \mathcal{R}_\lambda(f_\lambda^*) + \tau \leq \mathcal{R}_\lambda(f^*) + \tau \leq \mathcal{R}(f) + \lambda\Omega(f^*) + \tau,$$

yielding $f \in \mathcal{B}_\tau$.

Next, set $\tau = \lambda\Omega(f^*)$ with λ to be chosen later. We show that $\hat{f}_\lambda$ belongs to $\mathcal{C}_\tau$ with high probability. If $\hat{f}_\lambda \notin \mathcal{C}_\tau$, since $f^*_\lambda \in \mathcal{C}_\tau$ and $\mathcal{C}_\tau$ is convex, there exists a $\tilde{f}$ in the segment $[\hat{f}_\lambda, f^*_\lambda]$ and which is on the boundary of $\mathcal{C}_\tau$, i.e. such that $\mathcal{R}_\lambda(\tilde{f}) = \mathcal{R}_\lambda(f^*_\lambda) + \tau$. Since the empirical risk is convex, we have $\widehat{\mathcal{R}}_\lambda(\tilde{f}) \leq \max(\widehat{\mathcal{R}}_\lambda(\hat{f}_\lambda), \widehat{\mathcal{R}}_\lambda(f^*_\lambda)) = \widehat{\mathcal{R}}_\lambda(f^*_\lambda)$. Then,

$$\begin{aligned}\widehat{\mathcal{R}}(f^*_\lambda) - \widehat{\mathcal{R}}(\tilde{f}) - \mathcal{R}(f^*_\lambda) + \mathcal{R}(\tilde{f}) &= \widehat{\mathcal{R}}_\lambda(f^*_\lambda) - \widehat{\mathcal{R}}_\lambda(\tilde{f}) - \mathcal{R}_\lambda(f^*_\lambda) + \mathcal{R}_\lambda(\tilde{f}) \\ &\geq -\mathcal{R}_\lambda(f^*_\lambda) + \mathcal{R}_\lambda(\tilde{f}) = \tau. \end{aligned} \tag{4.14}$$

Note that $\Omega(\tilde{f}) \leq 2\Omega(f^*)$ and $\Omega(f^*_\lambda) \leq 2\Omega(f^*)$. Combining Lemma 36 and Lemma 38, for $\delta \in (0,1)$, with probability greater than $1-\delta$, we have for all $f \in \mathcal{F}_\infty$ such that $\Omega(f) \leq 2\Omega(f^*)$:

$$\begin{aligned}&\widehat{\mathcal{R}}(f^*_\lambda) - \widehat{\mathcal{R}}(f) - \mathcal{R}(f^*_\lambda) + \mathcal{R}(f) \\ &\quad\leq \mathbb{E}_{\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty, \Omega(f)\leq 2\Omega(f^*)} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) + \sup_{f\in\mathcal{F}_\infty, \Omega(f)\leq 2\Omega(f^*)} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right) \\ &\qquad + \Omega(f^*)\frac{96\sqrt{2e}L\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}} \\ &\quad\leq 12\Omega(f^*)L\left(\frac{1}{\sqrt{n}} + G_n\right) + \Omega(f^*)\frac{96\sqrt{2e}L\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}.\end{aligned}$$

Now, choose λ such that $\tau = \lambda\Omega(f^*) \geq 12\Omega(f^*)L\left(\frac{1}{\sqrt{n}} + G_n\right) + \Omega(f^*)\frac{96\sqrt{2e}L\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}$. This yields a contradiction with Equation (4.14). Thus, with such a λ, with probability greater than $1-\delta$, we have $\hat{f}_\lambda \in \mathcal{C}_\tau$, hence

$$\begin{aligned}\mathcal{R}_\lambda(\hat{f}_\lambda) &\leq \mathcal{R}_\lambda(f^*_\lambda) + \lambda\Omega(f^*), \\ \mathcal{R}(\hat{f}_\lambda) &\leq \mathcal{R}(f^*) + 2\lambda\Omega(f^*).\end{aligned}$$

For $\lambda = 12L\left(\frac{1}{\sqrt{n}} + G_n\right) + \frac{288L\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}$, this yields

$$\mathcal{R}(\hat{f}_\lambda) \leq \mathcal{R}(f^*) + \Omega(f^*)\left(24L\left(\frac{1}{\sqrt{n}} + G_n\right) + \frac{576L\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}\right).$$

□

We now provide insightful comments on Theorem 9. We first remark that the result could be proven more directly for bounded data using McDiarmid's inequality, resulting in a better constant.

The chosen λ does not depend on unknown quantities such as $\Omega(f^*)$, but only on known quantities such as the Lipschitz constant of the loss, the sample size, or the dimension of the data. This allows λ to be explicitly chosen for a fixed probability δ, although it is usually computed through cross-validation.

Classical losses typically satisfy our assumptions. For instance, the square loss is always convex and L-Lipschitz if the data and response are bounded, with $L = 2\sup_{y\in\mathcal{Y}}|y| + 4\Omega(f^*)\sup_{x\in\mathcal{X}}\|x\|^*$. Similarly, the logistic loss is always convex and L-Lipschitz with $L = 1$ (in the context of outputs in $\{-1, 1\}$).

Our approach stands out by requiring minimal assumptions on the data-generating mechanism, which is less restrictive compared to other methodologies in the multi-index

model domain. This emphasis on general applicability is also why we do not include feature recovery results, as such outcomes typically necessitate strong assumptions about the data and often require prior knowledge of the distribution.

The rates obtained depend explicitly on the dimension of the data through the bound on the Gaussian complexity. Considering the first term in the minimum, we observe that the bound is independent (up to logarithmic factors) of the data dimension, making BKerNN suitable for high-dimensional problems. However, this bound has a less favourable dependency on the sample size compared to the dimension-dependent bound, which is the second term in the minimum. We conjecture that the actual rate has the best of both worlds, achieving an explicit dependency on dimension d and sample size n of $n^{-1/2}$ (up to logarithmic factors).

Comparing the rate between BKerNN, neural networks with ReLU activations, and kernel methods, we find that in well-specified settings (where the Bayes estimator belongs to each function space considered), KRR yields a $O(n^{-1/2})$ rate independent of dimension, but require very smooth functions, for example, a Sobolev space of order s (i.e. the derivatives up to order s are square integrable) is only a RKHS if $s > d/2$ [Bach, 2024, Chapter 7]. Neural networks with ReLU activation achieve a similar rate with fewer constraints, as their function space is typically larger than RKHS spaces [Bach, 2024, Chapter 9].

If the model is not well-specified but we consider the Bayes predictor f^* to be Lipschitz continuous, the rates for neural networks with ReLU activation and bounded corresponding Banach norm γ_1 ($O(n^{-1/(d+5)})$) and kernel methods ($O(n^{-1/(d+1)})$) [Bach, 2024, Section 7.5, Section 9.4] do not beat the curse of dimensionality, and neither does our setup.

However, in the case of linear latent variables, i.e., under the multiple index model where $f^* = g^*(P^\top x)$ with P a $d \times k$ matrix with $k < d$ and orthonormal columns, the RKHS cannot take advantage of this hypothesis and the rates remain unchanged. In contrast, the neural network can, assuming that g^* has bounded Banach norm, then we only pay the price of the k underlying dimensions and not the full d dimensions [Bach, 2024, Section 9.4]. BKerNN also has this property, which is visible by using the simple arguments presented in the discussion in Bach [2024, Section 9.3.5], which show that $\Omega(f^*) \leq \Omega(g^*)$. Moreover, the optimisation process for BKerNN is much easier than that of neural networks, and our function space is larger, underscoring the attractiveness of BKerNN.

There is also an implicit dependency on the dimension in Theorem 9 through data-dependent terms, namely the variance proxy σ^2 or the expectations in the bound of G_n. We now examine these quantities under two data-generating mechanisms: bounded and subgaussian variables.

Lemma 33 (Analysis of Data-Dependent Terms in Theorem 9)**.** *The following inequalities hold.*

1. *If X is bounded, i.e., $\|X\|^* \leq R$ almost surely, then*

$$\sqrt{\mathbb{E}_X \|X\|^*} \leq \sqrt{R}, \quad \left(\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]} (\|X_i\|^*)^2\right)\right)^{1/4} \leq \sqrt{R}.$$

   Moreover, $1 + \sqrt{\|X\|^*}$ is subgaussian with variance proxy $\sigma^2 \leq 1 + \sqrt{R}$.

2. *If X is a vector of subgaussian variables (not necessarily centred or independent)*

with variance proxy σ_a^2 for component X_a, then

$$\sqrt{\mathbb{E}_X(\|X\|_2)} \leq \sqrt{6}\left(\sum_{a=1}^{d}\sigma_a^2\right)^{1/4}, \quad \sqrt{\mathbb{E}_X(\|X\|_\infty)} \leq 4(\log d)^{1/4}\max_{a\in[d]}\sqrt{\sigma_a},$$

$$\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}\|X_i\|_2^2\right)^{1/4} \leq 4(1+\log(n))^{1/4}\left(\sum_{a=1}^{d}\sigma_a^2\right)^{1/4},$$

$$\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]}\|X_i\|_\infty\right)^{1/4} \leq 4(1+\log(nd))^{1/4}\max_{a\in[d]}\sqrt{\sigma_a}.$$

Furthermore, $1+\sqrt{\|X\|_2}$ is subgaussian with variance proxy $\sigma^2 \leq (1+\sum_{a=1}^{d}\sigma_a)^2$, and $1+\sqrt{\|X\|_\infty}$ is subgaussian with variance proxy $\sigma^2 \leq 2+\max_{a\in[d]}\sigma_a^2(1+\sqrt{\log(2d)})^2$.

See the proof in Appendix A.5.5. Note that R usually does not implicitly depend on the dimension in the $\|\cdot\| = \|\cdot\|_\infty$ case, and R can typically be $O(d^{1/2})$ in the $\|\cdot\|_2$ case. For the subgaussian mechanism, each σ_a typically does not depend on the dimension.

5 Numerical Experiments

In this section, we present and analyse the properties of BKerNN. The BKerNN implementation in Python is fully compatible with Scikit-learn [Pedregosa et al., 2011], ensuring seamless integration with existing machine learning workflows. The source code, along with all necessary scripts to reproduce the experiments, is available at `https://github.com/BertilleFollain/BKerNN`. We define the scores and other estimators in the section below.

5.1 Introduction to Scores and Competitors

In the experiments below, we use two scores to assess performance. The prediction score is defined as the coefficient of determination, a classical metric in the statistics literature [Wright, 1921], R^2, which ranges from $-\infty$ to 1, where a score of 1 indicates perfect prediction, a score of 0 indicates that the model predicts no better than the mean of the target values, and negative values indicate that the model performs worse than this baseline. Mathematically, the R^2 score is defined as follows

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}, \tag{4.15}$$

where y_i are the true values, $\hat{y}_i$ are the predicted values, $\bar{y}$ is the mean of the true values, and n is the number of samples.

The feature learning score measures the model's ability to identify and learn the true feature space (1 being the best, 0 the worst). It is computable only when the underlying feature space (in the form of a matrix $P \in \mathbb{R}^{d\times k}$, with k the number of features) is known and relevant only when features are of similar importance, which we have ensured in the experiments below.

Depending on the regularisation type, the estimated feature matrix $\hat{P}$ is computed via singular value decomposition (SVD) for Ω_{feature}, $\Omega_{\text{concave feature}}$ or Ω_{basic} regularisation, or by selecting the top k variables for Ω_{variable} or $\Omega_{\text{concave variable}}$ regularisation. We then compute the projection matrices $\pi_{\hat{P}}$ and π_P and calculate the feature learning error as the

normalised Frobenius norm of their difference

$$\pi_{\hat{P}} = \hat{P}(\hat{P}^\top \hat{P})^{-1}\hat{P}^\top \quad \text{and} \quad \pi_P = P(P^\top P)^{-1}P^\top,$$

$$\text{score} = \begin{cases} 1 - \frac{\|\pi_P - \pi_{\hat{P}}\|_F^2}{2k} & \text{if} k \leq \frac{n_{\text{features}}}{2}, \\ 1 - \frac{\|\pi_P - \pi_{\hat{P}}\|_F^2}{2n_{\text{features}} - 2k} & \text{if} k > \frac{n_{\text{features}}}{2}, \end{cases} \tag{4.16}$$

where the score is 1 if $k = n_{\text{features}}$.

In several experiments, we compare the performance of BKerNN against ReLUNN and BKRR. BKRR refers to Kernel Ridge Regression using the multi-dimensional Brownian kernel $k^{(mdB)}(x, x') = (\|x\| + \|x'\| - \|x - x'\|)/2$. ReLUNN is a simple one-hidden-layer neural network with ReLU activations, trained using batch stochastic gradient descent.

5.2 Experiment 1: Optimisation Procedure, Importance of Positive Homogeneous Kernel

In this experiment, we compare BKerNN with two methods that differ from BKerNN only through the kernel that is used. We wish to illustrate the importance of the homogeneity assumptions discussed in Section 3.2. Specifically, we consider ExpKerNN with the (rescaled) exponential kernel $k^{\exp}(a, b) = e^{-|a-b|/2}$ and GaussianKerNN with the Gaussian kernel $k^{\text{Gaussian}}(a, b) = e^{-|a-b|^2/2}$. Unlike the Brownian kernel used in BKerNN, the exponential and Gaussian kernels are not positively 1-homogeneous.

We trained all three methods on a simulated data set, using cross-validation to select the regularisation parameter λ while keeping other parameters fixed ($m = 100$, basic regularisation, more details are provided in Appendix B.1). The training set consisted of 214 samples and the test set of 1024. The data had $d = 45$ dimensions with $k = 5$ relevant features, and Gaussian additive noise with a standard deviation of 0.5. An orthogonal matrix P of size $d \times d$ was sampled uniformly from the orthogonal group before being truncated to size $d \times k$. The covariates were sampled uniformly from $[-1, 1]^d$, and the target variable y was computed as $y = 2\pi \left|\sum_{a=1}^{k} (P^\top x)_a\right| + \text{noise}$.

We displayed the mean squared error (MSE) on both the training and test sets for the selected λ for each method in Figure 4.1. While all three methods perform very well on the training set, the test set performance of ExpKerNN and GaussianKerNN is significantly worse compared to BKerNN. This discrepancy is not due to suboptimal regularisation choices, as cross-validation was used to select the best λ for each method.

Instead, the superior test performance of BKerNN underscores its effective optimisation process, avoiding the pitfalls of local minima that seem to trap ExpKerNN and GaussianKerNN. Our observations in Figure 4.1 strongly support our discussion in Section 3.2 on the critical role of the positive homogeneity of the kernel in ensuring convergence to a global minimum.

5.3 Experiments 2 & 3: Influence of Parameters (Number of Particles m, Regularisation Parameter λ, and Type of Regularisation)

In these experiments, we explore the impact of various parameters on the performance of BKerNN. Detailed descriptions can be found in Appendix B.2, the results are presented in Figure 4.2 and the R^2 score is described in Equation (4.15).

Experiment 2. The first two subplots of Figure 4.2 illustrate the effects of the number of particles m and the regularisation parameter λ while keeping the data generation process

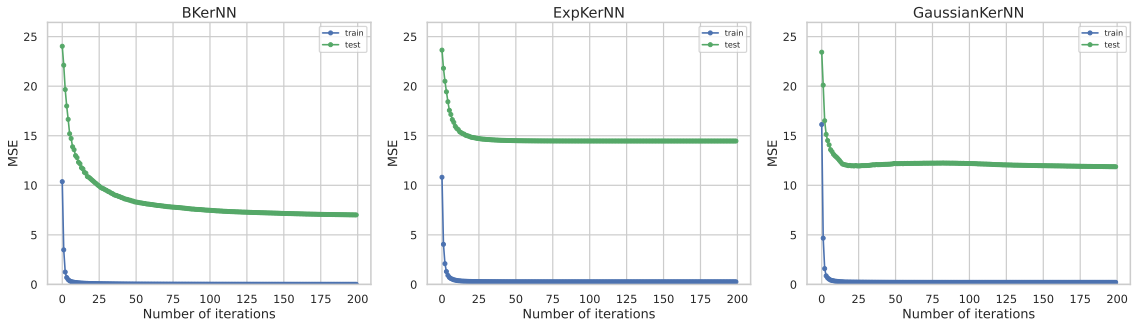


Figure 4.1: MSE across optimisation procedure for different kernels.

consistent. The data set is the same for the two subplots. We used 412 training samples and 1024 test samples, with a data dimensiona of $d = 20$ and $k = 5$ relevant features. The standard deviation of additive Gaussian noise was set to 0.1. The covariates were sampled uniformly from $[-1, 1]^d$. The target variable y was computed as $y = \sum_{a=1}^{k} |2\pi x_a| + \text{noise}$.

Number of Particles (m): the first subplot shows that with too few particles, the estimator struggles to fit the training data, leading to poor performance on the test set. However, beyond a certain threshold, increasing the number of particles does not yield significant improvements in performance.

Regularisation Parameter (λ): the second subplot demonstrates the typical behaviour of a regularised estimator. When λ is too small, the model overfits the training data, resulting in poor test performance. Conversely, when λ is too large, the model underfits, performing poorly on both the training and test sets. Optimal performance on both sets is achieved with an intermediate value of λ.

Experiment 3. The third subplot in Figure 4.2 examines the influence of the type of regularisation across three distinct data-generating mechanisms: (1) without underlying features, i.e., where all of the data is needed, (2) with few relevant variables, (3) with few relevant features. We used 214 training samples and 1024 test samples, with a data dimensionality of $d = 20$ and $k = 5$ relevant features. The standard deviation of additive Gaussian noise was set to 0.5, and the data set was generated 20 times with different seeds. The covariates were always sampled uniformly on $[-1, 1]^d$ but the response was generated in three different ways. In the "no underlying structure" data set, we had $y = \sum_{a=1}^{d} \sin(X_a) + \text{noise}$. In the "few relevant variables" data set, we had $y = \sum_{a=1}^{k} \sin(x_a) + \text{noise}$. In the "few relevant features data set", we sampled P a $d \times d$ matrix from the orthogonal group uniformly, truncated it to size $d \times k$ and the response was generated as $y = \sum_{a=1}^{k} \sin((P^\top x)_a) + \text{noise}$. The mean and standard deviation of the R^2 score on the test set are reported.

When there is no underlying structure, all regularisers perform somewhat similarly. However, for data sets featuring relevant variables, the Ω_{variable} and $\Omega_{\text{concave variable}}$ regularisations shine, delivering superior performance. Similarly for the Ω_{feature} and $\Omega_{\text{concave feature}}$ regularisations on data sets with few relevant features. Remarkably, for data with underlying structure, the concave versions of both Ω_{variable} and Ω_{feature} regularisations outperform their non-concave counterparts. This demonstrates their superior ability to effectively select relevant information in the data while maintaining strong predictive power.

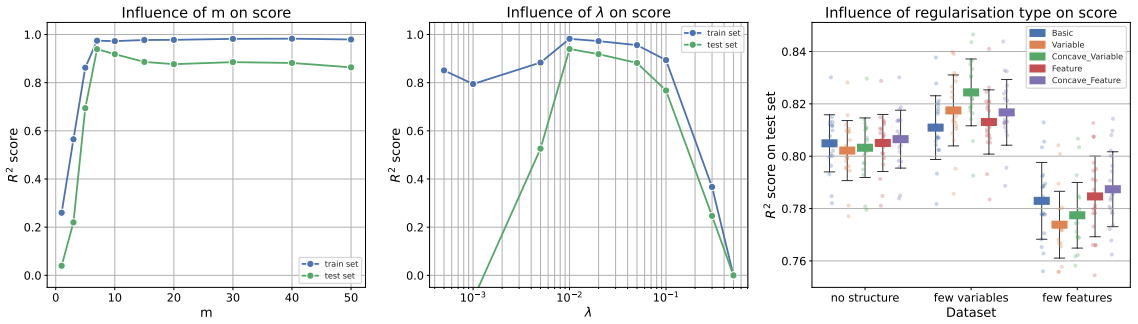


Figure 4.2: Influence of parameters: left: m, middle: λ, right: type of penalty.

5.4 Experiment 4: Comparison to Neural Network on 1D Examples, Influence of Number of Particles/Width of Hidden Layer m

In Experiment 4, we compare the learning capabilities of BKerNN against a simple neural network, ReLUNN. We study three distinct functions, corresponding to each row in Figure 4.3. In all rows, the training set is represented by small black crosses, while the target function is shown in blue. The first two columns depict BKerNN using two different numbers of particles: $m = 1$ and $m = 5$. The last three columns show results for ReLUNN with varying numbers of neurons in the hidden layer: 1, 5, and 32. See Appendix B.3 for more experimental details.

Notably, BKerNN demonstrates great learning capabilities, successfully capturing the functions even with just one particle. Increasing the number of particles (second column) offers minimal additional benefit, underscoring BKerNN's efficiency. In stark contrast, ReLUNN struggles significantly when limited to the same number of hidden neurons as BKerNN's particles. However, once the hidden layer is expanded to 32 neurons, ReLUNN begins to show satisfactory learning capabilities. These results highlight BKerNN's superior efficiency in learning functions with a minimal number of particles, outperforming ReLUNN, which requires a more complex architecture to achieve comparable performance.

5.5 Experiment 5: Prediction Score and Feature Learning Score Against Growing Dimension and Sample Size, a Comparison of BKerNN with Brownian Kernel Ridge Regression and a ReLU Neural Network

In Experiment 5, we evaluate the performance of BKerNN, BKRR and ReLUNN across varying sample sizes and dimensions on simulated data sets. The estimators are presented in Section 5.1. The R^2 and feature learning score used to assess performance are described in Equations (4.15) and (4.16) respectively. The results are presented in Figure 4.4. For more details about the experiment, see Appendix B.4.

The two subplots on the top row of Figure 4.4 show the effect of increasing the sample size while keeping the dimension fixed. In the two subplots of the bottom row, the sample size is fixed, and the dimension is increased. For each combination of sample size and dimension, ten data sets were generated. We display the two scores of each method on each data set, as well as the average score across data sets. The feature learning score for BKRR is not defined and, therefore, not displayed. The number of particles (for BKerNN) and hidden neurons (for ReLUNN) is fixed at 50 across all experiments.

For all the data sets, the covariates were uniformly sampled in $[-1, 1]^d$, the underlying

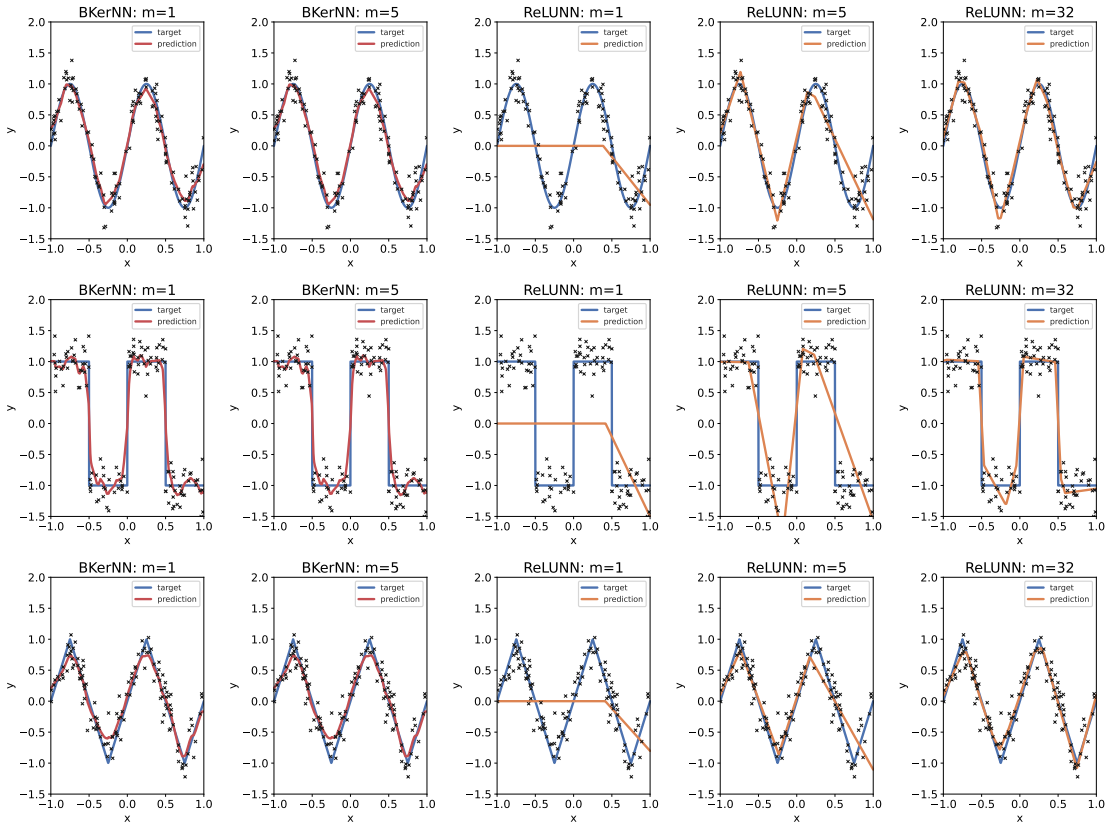


Figure 4.3: Comparison to neural network on 1D examples.

features matrix P was uniformly sampled from the orthogonal group, then truncated to have $k = 3$ relevant features, and the response was set as $y = \left|\sum_{a=1}^{k} \sin\left((P^\top x)_a\right)\right|$.

In the first two subplots, the dimension is fixed at 15. As the sample size increases, we observe improvements in the prediction scores of all three methods. However, the prediction score of BKRR improves at a much slower pace. Both BKerNN and ReLUNN achieve high prediction scores more rapidly, with BKerNN requiring fewer samples to do so. Notably, BKerNN excels in feature learning, effectively capturing the underlying feature space, while ReLUNN fails regardless of the number of samples.

In the last two subplots, where the sample size is fixed at 212, we notice a general decline in performance as the dimension increases. BKRR shows the most rapid deterioration because it cannot learn features, struggling significantly with higher dimensions. In contrast, BKerNN demonstrates resilience to increasing dimensionality, maintaining better performance compared to the other methods. ReLUNN falls somewhere in between, neither as robust as BKerNN nor as weak as BKRR. Similarly, for the feature learning score, both BKerNN and ReLUNN show decreased performance, but BKerNN is slightly less affected, underscoring its ability to handle high-dimensional data.

5.6 Experiment 6: Comparison on Real Data Sets Between BKerNN, Brownian Kernel Ridge Regression and a ReLU Neural Network

In Experiment 6, we evaluate the R^2 score (Equation 4.15), of four methods: BKRR, BKerNN with concave variable regularisation, BKerNN with concave feature regularisation, and ReLUNN, across 17 real-world data sets. These were obtained from the

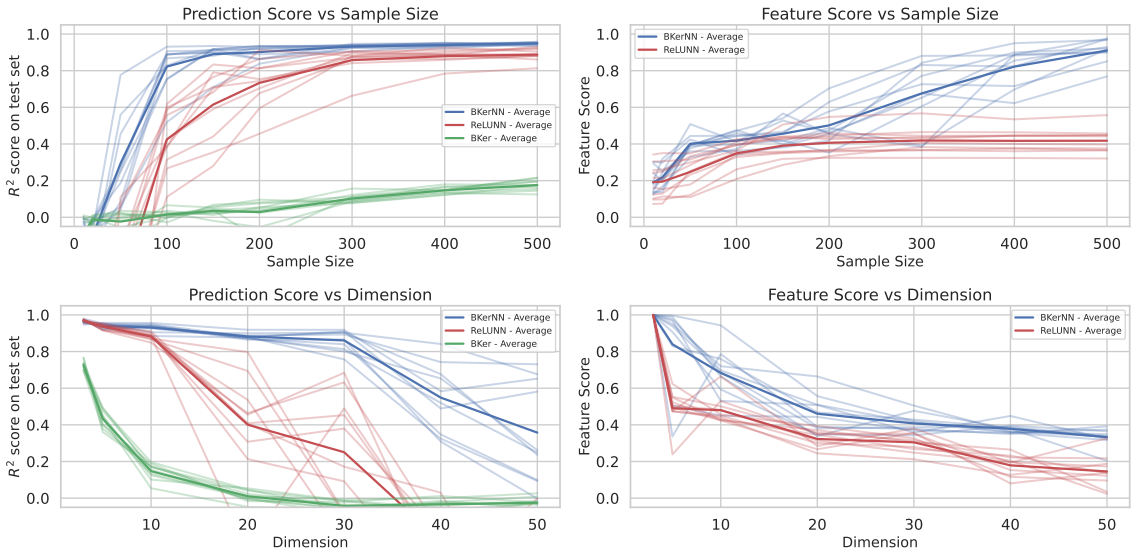


Figure 4.4: Performance comparison across varying sample sizes and dimensions.

tabular benchmark numerical regression suite via the OpenML platform, as described by Grinsztajn et al. [2022]. Each data set was processed to only include numerical variables, cropped to 400 training samples and 100 testing samples, rescaled to have centred covariates with standard deviation equal to one, with dimensionality varying across data sets as shown in Figure 4.5. For both BKerNN and ReLUNN, the number of particles or hidden neurons was set to twice the dimension of each data set, while the training parameters were fixed. Details are available in Appendix B.5.

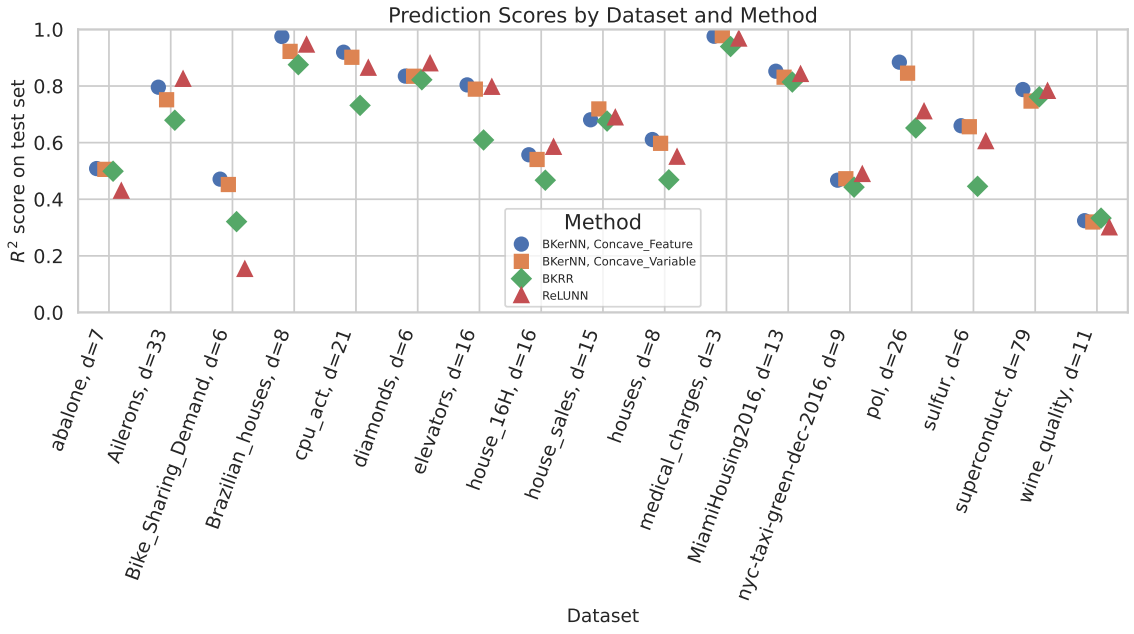


Figure 4.5: Comparison of R^2 scores on real data sets.

The results indicate that BKRR often performs the worst among all methods. In contrast, BKerNN with concave feature regularisation and ReLUNN frequently emerge as the best estimators, performing similarly well on average across the various data sets.

6 Conclusion

To conclude, we have introduced a novel framework for feature learning and function estimation in supervised learning, termed Brownian kernel neural network (BKerNN). By leveraging regularised empirical risk minimisation over averages of Sobolev spaces on one-dimensional projections of the data, we established connections to kernel ridge regression and infinite-width one-hidden layer neural networks. We provide an efficient computational method for BKerNN, emphasising the importance of the positive homogeneity of the Brownian kernel. Through rigorous theoretical analysis, we demonstrated that, in the well-specified setting for subgaussian data, BKerNN achieves convergence of its expected risk to the minimal risk with explicit rates, potentially independent of the data dimension, underscoring the efficacy of our approach. We have extensively discussed the relationship between the space of functions we propose and other classical functions spaces. Numerical experiments across simulated scenarios and real data sets confirm BKerNN's superiority over traditional kernel ridge regression and competitive performance with neural networks employing ReLU activations, achieved with fewer particles or hidden neurons. Future research directions include the development of more efficient algorithms for the computation of the estimator, improved analysis of the Gaussian complexity, and theoretical investigation of other penalties.

Appendix

A Extra Lemmas and Proofs

In this appendix, we present and/or prove some of the results needed in the main text.

A.1 Well-Definition of $\mathcal{F}_\infty$

Recall the definition of $\mathcal{F}_\infty$ given in Definition 4. Let

$$\mathcal{F}_\infty := \left\{ f \mid f(\cdot) = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top \cdot)\, \mathrm{d}\mu(w), \quad \Omega_0(f) < \infty \right\},$$

where c is a constant in $\mathbb{R}$, $\mathcal{S}^{d-1}$ is the unit sphere for some norm, $\mu \in \mathcal{P}(\mathcal{S}^{d-1})$ is a probability measure on $\mathcal{S}^{d-1}$, and $\forall w \in \mathcal{S}^{d-1}$, $g_w : \mathbb{R} \to \mathbb{R} \in \mathcal{H}$, with

$$\Omega_0(f) := \inf_{c \in \mathbb{R}, (g_w)_w \in \mathcal{H}^{\mathcal{S}^{d-1}}, \mu \in \mathcal{P}(\mathcal{S}^{d-1})} \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}}\, \mathrm{d}\mu(w),$$

such that $f = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top \cdot)\, \mathrm{d}\mu(w)$.

Now consider a different and more formal definition, which we show to be equivalent. We define $\mathcal{F}_\infty$ to be the space of functions $f : \mathbb{R}^d \to \mathbb{R}$ such that there exists a Borel measure $\tilde{\mu}$ such that $f = c + \int_{\mathcal{H} \times \mathcal{S}^{d-1}} g(w^\top \cdot) \mathrm{d}\tilde{\mu}(g, w)$ with $\Omega_0(f) < \infty$,where Ω_0 on f is defined as the infimum of $\int_{\mathcal{H} \times \mathcal{S}^{d-1}} \|g\|_{\mathcal{H}} \mathrm{d}\tilde{\mu}(g, w)$ over all measures defining f. This is the variation norm associated to the map $(g, w) \in \mathcal{H} \times \mathcal{S}^{d-1} \to g(w^\top \cdot)$ (which is a function from $\mathbb{R}^d$ to $\mathbb{R}$), see Kurkova and Sanguineti [2001] and Bach [2024, Section 9.3.2].

Since $\int_{\mathcal{H} \times \mathcal{S}^{d-1}} \|g\|_{\mathcal{H}} \mathrm{d}\tilde{\mu}(g, w)$ does not depend on w, we can write

$$f = c + \int_{\mathcal{S}^{d-1}} \left(\int_{\mathcal{H}} g(w^\top \cdot) \mathrm{d}\tilde{\mu}(g|w) \right) \mathrm{d}\tilde{\mu}(w),$$

where we then see that if we define for $w \in \mathcal{S}^{d-1}$ the function $g_w := \int_{\mathcal{H}} g \mathrm{d}\tilde{\mu}(g|w)$, it indeed belongs to $\mathcal{H}$ and $\|g_w\|_{\mathcal{H}} \leq \int_{\mathcal{H}} \|g\|_{\mathcal{H}} \mathrm{d}\tilde{\mu}(g|w)$. For any optimal measure $\tilde{\mu}^*$ in the definition of $\Omega_0(f)$, we must have the equality $\|g_w\|_{\mathcal{H}} = \int_{\mathcal{H}} \|g\|_{\mathcal{H}} \mathrm{d}\tilde{\mu}^*(g|w)$, as otherwise we could take a Dirac at g_w for $\tilde{\mu}^*(g, w)$ and improve the infimum defining Ω_0. We then have

$$\Omega_0(f) = \int_{\mathcal{H} \times \mathcal{S}^{d-1}} \|g\|_{\mathcal{H}} \mathrm{d}\tilde{\mu}(g, w) = \int_{\mathcal{S}^{d-1}} \left(\int_{\mathcal{H}} g(w^\top \cdot) \mathrm{d}\tilde{\mu}(g|w) \right) \mathrm{d}\tilde{\mu}(w) = \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}} \mathrm{d}\tilde{\mu}(w),$$

with the measure μ of the first definition being equal to the marginal of $\tilde{\mu}$, which ends the presentation of the equivalence between the two definitions.

A.2 Proofs of Section 2.3 Lemmas

Here we give the proofs of the lemmas describing characteristics of the function space $\mathcal{F}_\infty$.

A.2.1 Proof of Lemma 24

Proof of Lemma 24. We first check that $\mathcal{F}_\infty$ is a vector space.

Let $f \in \mathcal{F}_\infty$ with $f(\cdot) = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top \cdot) \mathrm{d}\mu(w)$ and $\tau \in \mathbb{R}$ then $\tau f(\cdot) = \tau c + \int_{\mathcal{S}^{d-1}} \tau g_w(w^\top \cdot) \mathrm{d}\mu(w)$ and $\tau g_w \in \mathcal{H}$. We also see that $\Omega(\tau f) = |\tau| \Omega(f)$, hence $\tau f \in \mathcal{F}_\infty$.

Now let $f, \tilde{f} \in \mathcal{F}_\infty$, then $(f+\tilde{f})(\cdot) = c+\tilde{c}+\int_{\mathcal{S}^{d-1}} g_w(w^\top \cdot)\mathrm{d}\mu(w)+\int_{\mathcal{S}^{d-1}} \tilde{g}_w(w^\top \cdot)\mathrm{d}\tilde{\mu}(w) = c+\tilde{c}+\int_{\mathcal{S}^{d-1}}(\mathrm{d}\frac{2\mu(w)}{\mu(w)+\tilde{\mu}(w)}g_w + \mathrm{d}\frac{2\tilde{\mu}(w)}{\mu(w)+\tilde{\mu}(w)}\tilde{g}_w)(w^\top \cdot)\mathrm{d}(\frac{\mu+\tilde{\mu}}{2})(w)$. We also have that

$$\begin{aligned}
\Omega_0(f+\tilde{f}) &= \int_{\mathcal{S}^{d-1}} \left\| \mathrm{d}\frac{2\mu(w)}{\mu(w)+\tilde{\mu}(w)} g_w + \mathrm{d}\frac{2\tilde{\mu}(w)}{\mu(w)+\tilde{\mu}(w)}\tilde{g}_w \right\|_{\mathcal{H}} \mathrm{d}\left(\frac{\mu+\tilde{\mu}}{2}\right)(w) \\
&\leq \int_{\mathcal{S}^{d-1}} \mathrm{d}\frac{2\mu(w)}{\mu(w)+\tilde{\mu}(w)}\|g_w\|_{\mathcal{H}} + \mathrm{d}\frac{2\tilde{\mu}(w)}{\mu(w)+\tilde{\mu}(w)}\|\tilde{g}_w\|_{\mathcal{H}}\mathrm{d}\left(\frac{\mu+\tilde{\mu}}{2}\right)(w) \\
&\leq \Omega_0(f) + \Omega_0(\tilde{f}) < \infty,
\end{aligned}$$

hence $f+\tilde{f}$ belongs to $\mathcal{F}_\infty$. This yields that $\mathcal{F}_\infty$ is a vector space. We also see that $\Omega(f+\tilde{f}) = \max(f(0)+\tilde{f}(0), \Omega_0(f+\tilde{f})) \leq \Omega(f) + \Omega(\tilde{f})$. Since $\Omega(f) = 0 \iff f = 0$, we have that Ω is a norm on $\mathcal{F}_\infty$.

We now check the Hölder continuity property

$$\begin{aligned}
f(x) - f(x') &= c + \int_{\mathcal{S}^{d-1}} g_w(w^\top x)\mathrm{d}\mu(w) - c - \int_{\mathcal{S}^{d-1}} g_w(w^\top x')\mathrm{d}\mu(w) \\
&= \int_{\mathcal{S}^{d-1}} \langle g_w, k_{w^\top x} - k_{w^\top x'}\rangle \mathrm{d}\mu(w) \\
|f(x) - f(x')| &\leq \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}}\|k_{w^\top x} - k_{w^\top x'}\|_{\mathcal{H}}\mathrm{d}\mu(w) \leq \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}}\sqrt{|w^\top(x-x')|}\mathrm{d}\mu(w) \\
&\leq \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}}\sqrt{\|x-x'\|^*}\mathrm{d}\mu(w) \leq \Omega_0(f)\sqrt{\|x-x'\|^*}.
\end{aligned}$$

□

A.2.2 Proof of Lemma 25

Proof of Lemma 25. Let us assume now that we only consider functions f with support on the ball with centre 0, radius R and norm $\|\cdot\|^*$, which we denote $B(0,R)$. Then we can actually consider the functions g_w which define $\mathcal{F}_\infty$ to belong to $\mathcal{H}' := \{g : \mathbb{R} \to \mathbb{R} \mid g(0) = 0, \int_{-R}^{R}(g'(t))^2\mathrm{d}t\}$, and it is still a RKHS with the same reproducing kernel. Let $f \in \mathcal{F}_\infty, f = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top \cdot)\mathrm{d}\mu(w)$. We have assumed it has a Fourier decomposition, such that

$$f(x) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} \hat{f}(\omega)e^{i\omega^\top x}\mathrm{d}\omega,$$

and then we have

$$\Omega_0(f) \leq \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} |\hat{f}(\omega)|\Omega_0(e^{i\omega^\top x})\mathrm{d}\omega$$

and we can then study $\Omega_0(e^{i\omega^\top x})$.

We have $e^{i\omega^\top x} = g_\omega(\frac{\omega}{\|\omega\|}^\top x)$ with $g_\omega : t \in [-R, R] \to e^{it\|\omega\|}$ which belongs to (the complex version of) $\mathcal{H}$, with $\|g_\omega\|_{\mathcal{H}} = \sqrt{\int_{-R}^{R}\|\omega\|^2|e^{it\|\omega\|}|^2\mathrm{d}t} \leq \sqrt{2R}\|\omega\|$.

This yields

$$\Omega_0(f) \leq \frac{\sqrt{2R}}{(2\pi)^d}\int_{\mathbb{R}^d} |\hat{f}(\omega)| \cdot \|\omega\|\mathrm{d}\omega.$$

□

A.3 Proofs of Section 2.4 Lemmas

In this section we present the proof of lemmas used to transform the optimisation problem defining BKerNN.

A.3.1 Proof of Lemma 26

Proof of Lemma 26. Our goal is to transform Equation (4.5). We begin with the following trick for the m particles setting

$$\frac{1}{m}\sum_{j=1}^{m}\|g_j\|_{\mathcal{H}} = \inf_{\beta\in\mathbb{R}_+^m}\frac{1}{2m}\sum_{j=1}^{m}\left(\frac{\|g_j\|_{\mathcal{H}}^2}{\beta_j}+\beta_j\right).$$

Fix $(w_j)_{j\in[m]}$ and $(\beta_j)_{j\in[m]}$ in Equation (4.5), yielding the following minimisation problem on the functions $(g_j)_{j\in[m]}$

$$\min_{c\in\mathbb{R},g_1,\dots,g_m\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n}\ell(y_i,c+\frac{1}{m}\sum_{j=1}^{m}g_j(w_j^\top x_i))+\frac{\lambda}{2}\frac{1}{m}\sum_{j=1}^{m}\frac{\|g_j\|_{\mathcal{H}}^2}{\beta_j}. \tag{4.17}$$

Using the representer theorem [Schölkopf et al., 2001], we express each $x\to g_j(w_j^\top x)$ as

$$x\to\sum_{i=1}^{n}\alpha_i^{(j)}k^{(B)}(w_j^\top x_i,w_j^\top x),$$

which leads to

$$\|g_j\|_{\mathcal{H}}^2=\sum_{i,i'=1}^{n}\alpha_i^{(j)}\alpha_{i'}^{(j)}k^{(B)}(w_j^\top x_i,w_j^\top x_{i'}).$$

Rewriting the norm and evaluation in kernel form with $K_{i,i'}^{(w_j)}=k^{(B)}(w_j^\top x_i,w_j^\top x_{i'})$, we obtain

$$\|g_j\|_{\mathcal{H}}^2=(\alpha^{(j)})^\top K^{(w_j)}\alpha^{(j)},$$

and

$$g_j(w_j^\top x_i)=(K^{(w_j)}\alpha^{(j)})_i.$$

Thus, we transform Equation (4.17) into

$$\min_{c\in\mathbb{R},\alpha^{(1)},\dots,\alpha^{(m)}\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\frac{1}{m}\sum_{j=1}^{m}(K^{(w_j)}\alpha^{(j)})_i+c)+\frac{\lambda}{2}\frac{1}{m}\sum_{j=1}^{m}\frac{(\alpha^{(j)})^\top K^{(w_j)}\alpha^{(j)}}{\beta_j}.$$

We show that minimisation is attained for vectors $\alpha^{(j)}$ equal to $\beta_j\alpha$ for a single vector α. Consider the convex problem

$$\min_{\alpha^{(1)},\dots,\alpha^{(m)}\in\mathbb{R}^d}\frac{1}{2}\frac{1}{m}\sum_{j=1}^{m}\frac{(\alpha^{(j)})^\top K^{(w_j)}\alpha^{(j)}}{\beta_j},$$

subject to $\frac{1}{m}\sum_{j=1}^{m}K^{(w_j)}\alpha^{(j)}=z$ where $z\in\mathbb{R}^d$. We define the Lagrangian

$$\mathcal{L}(\alpha^{(1)},\cdots,\alpha^{(m)},\alpha)=\frac{1}{2}\frac{1}{m}\sum_{j=1}^{m}\frac{(\alpha^{(j)})^\top K^{(w_j)}\alpha^{(j)}}{\beta_j}+\alpha^\top\left(z-\frac{1}{m}\sum_j K^{(w_j)}\alpha^{(j)}\right).$$

By taking the differential of $\mathcal{L}$ with respect to $\alpha^{(j)}$ at the optimum, we get

$$\frac{\partial \mathcal{L}}{\partial \alpha^{(j)}} = \frac{1}{m} K^{(w_j)} \left(\frac{\alpha^{(j)}}{\beta_j} - \alpha \right) = 0.$$

The differential with respect to α yields that at the optimum, the constraint is verified, i.e., $z = \frac{1}{m} \sum_j K^{(w_j)} \alpha^{(j)}$. We note that for $\alpha^{(j)} = \beta_j \alpha$, all equations are satisfied, yielding the desired result.

We can then write Equation (4.5) as

$$\min_{w_1,\dots,w_m \in \mathbb{R}^d, c \in \mathbb{R}, \beta \in \mathbb{R}_+^m, \alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2} \alpha^\top K \alpha + \frac{\lambda}{2} \frac{1}{m} \sum_{j=1}^m \beta_j,$$

with the constraints $\forall j \in [m], w_j \in \mathcal{S}^{d-1}$, and $K = \frac{1}{m} \sum_{j=1}^m \beta_j K^{(w_j)}$.

We notice that $\beta_j K^{(w_j)} = K^{(\beta_j w_j)}$ due to the positive homogeneity of the Brownian kernel. We therefore introduce the change of variable $\beta_j w_j = \tilde{w}_j$

$$\min_{\tilde{w},\dots,\tilde{w}_m \in \mathbb{R}^d, c \in \mathbb{R}, \beta \in \mathbb{R}_+^m, \alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2} \alpha^\top K \alpha + \frac{\lambda}{2} \frac{1}{m} \sum_{j=1}^m \|\tilde{w}_j\|,$$

with $K = \frac{1}{m} \sum_{j=1}^m K^{(\tilde{w}_j)}$ and no constraint on the norm of $\tilde{w}_j$. For ease of exposition in the main text, we replace $\tilde{w}$ by w. □

A.3.2 Proof of Lemma 27

Proof of Lemma 27. The proof follows the same steps as the proof of Lemma 26, systematically replacing any $\frac{1}{m} \sum_{j=1}^m$ with the appropriate integral over $\mathcal{S}^{d-1}$ with respect to measure μ. Before the change of variables, the problem is

$$\min_{\mu \in \mathcal{P}(\mathcal{S}^{d-1}), c \in \mathbb{R}, (\beta_w)_w \in \mathbb{R}_+^{\mathcal{S}^{d-1}}, \alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2} \alpha^\top K \alpha + \frac{\lambda}{2} \int_{\mathcal{S}^{d-1}} \beta_w \mathrm{d}\mu(w),$$

where $K = \int_{\mathcal{S}^{d-1}} \beta_w K^{(w)} \, d\mu(w) = \int_{\mathcal{S}^{d-1}} K^{(\beta_w w)} \, d\mu(w)$. The change of variables $\beta_w w = \tilde{w}$ transforms the problem into

$$\min_{(\beta_w)_w \in \mathbb{R}_+^{\mathcal{S}^{d-1}}, \nu \in \mathcal{P}(\{\beta_w w, w \in \mathcal{S}^{d-1}\}), c \in \mathbb{R}, \alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2} \alpha^\top K \alpha + \frac{\lambda}{2} \int_{\mathbb{R}^d} \|\tilde{w}\| \mathrm{d}\nu(\tilde{w}),$$

with $K = \int_{\mathbb{R}^d} K^{(\tilde{w})} \mathrm{d}\nu(\tilde{w})$. We can consider the integral over $\mathbb{R}^d$ instead of $\{\beta_w w, w \in \mathcal{S}^{d-1}\}$ by extending ν with $\nu(\mathbb{R}^d \setminus \{\beta_w w, w \in \mathcal{S}^{d-1}\}) = 0$. This is equivalent to considering the minimum over $\nu \in \mathcal{P}(\mathbb{R}^d)$ instead of the minimum over $(\beta_w)_{w \in \mathcal{S}^{d-1}} \in \mathbb{R}_+$ and $\nu \in \mathcal{P}(\{\beta_w w, w \in \mathcal{S}^{d-1}\})$.

The first minimum is smaller as it is considered over a larger space, but they are equal because both the norm $\|\cdot\|$ and the kernel K are positively homogeneous. Hence, the problem finally becomes

$$\min_{\nu \in \mathcal{P}(\mathbb{R}^d), c \in \mathbb{R}, \alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n \ell(y_i, (K\alpha)_i + c) + \frac{\lambda}{2} \alpha^\top K \alpha + \frac{\lambda}{2} \int_{\mathbb{R}^d} \|\tilde{w}\| \mathrm{d}\nu(\tilde{w}),$$

with $K = \int_{\mathbb{R}^d} K^{(\tilde{w})} \mathrm{d}\nu(\tilde{w})$. Learning an optimal ν yields an optimal μ by taking $\mathrm{d}\mu(w) =$

$\mathrm{d}\nu(\{\tilde{w} \in \mathbb{R}^d \mid \tilde{w}/\|\tilde{w}\| = w\})$. For ease of exposition in the main text, we replace $\tilde{w}$ by w. □

A.4 Proofs of Section 3.1 Lemmas

In this section, we provide the proofs of the lemmas used to compute the estimator.

A.4.1 Proof of Lemma 28

Proof of Lemma 28. For a fixed α, the optimal c is given by $c = \frac{\mathbb{1}^\top Y}{n} - \frac{\mathbb{1}^\top K\alpha}{n}$. Substituting this back into the objective function, we obtain

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{2n}\|\Pi Y - \Pi K\alpha\|_2^2 + \frac{\lambda}{2}\alpha^\top K\alpha,$$

which is minimised for α satisfying $(K\Pi K + n\lambda K)\alpha = K\Pi Y$. We can further simplify this by observing that if $(\Pi K + n\lambda I)\alpha = \Pi Y$, then the previous condition is satisfied.

From the equation $n\lambda\alpha = \Pi Y - \Pi K\alpha$, we can deduce that $\Pi\alpha = \alpha$ because $\Pi^2 = \Pi$. Therefore, we can express α as $\alpha = \Pi\tilde{\alpha}$. Substituting this change of variable into the original problem, we define $\tilde{K} := \Pi K\Pi$ and $\tilde{Y} := \Pi Y$, transforming the problem into

$$\min_{\tilde{\alpha}\in\mathbb{R}^n} \frac{1}{2n}\|\tilde{Y} - \tilde{K}\tilde{\alpha}\|_2^2 + \frac{\lambda}{2}\tilde{\alpha}^\top\tilde{K}\tilde{\alpha}.$$

This is a standard kernel ridge regression problem (noting that $\tilde{K}$ is still a valid kernel matrix), for which the solution is known to be $\tilde{\alpha} = (\tilde{K} + n\lambda I)^{-1}\tilde{Y}$. We also have $\Pi\tilde{\alpha} = \tilde{\alpha}$, implying $\alpha = \tilde{\alpha}$ because one can show that $\mathbb{1}^\top\tilde{\alpha} = 0$. To see why, note that $\mathbb{1}^\top\tilde{\alpha} = \langle\mathbb{1}, \tilde{\alpha}\rangle = \langle(\tilde{K} + n\lambda I)^{-1}\mathbb{1}, \tilde{Y}\rangle$. Since $(\tilde{K} + n\lambda I)^{-1}\mathbb{1}$ is proportional to $\mathbb{1}$ (as $\mathbb{1}$ is an eigenvector of $\tilde{K} + n\lambda I$ and its inverse), and $\langle\tilde{Y}, \mathbb{1}\rangle = 0$, we obtain the desired result.

Finally, we verify the optimal condition $(K\Pi K + n\lambda K)\alpha = K\Pi Y$. Given $(\Pi K\Pi + n\lambda I)\tilde{\alpha} = \Pi Y$ by definition, multiplying by K yields $(K\Pi K\Pi + n\lambda K)\tilde{\alpha} = K\Pi Y$. Since $\tilde{\alpha} = \alpha = \Pi\alpha$, the desired result follows. □

A.4.2 Proof of Lemma 29

Proof of Lemma 29. First, we compute the derivative of $G = \frac{\lambda}{2}\tilde{Y}^\top(\tilde{K} + \lambda nI)^{-1}\tilde{Y}$ with respect to w_j

$$\begin{aligned}\frac{\partial G}{\partial w_j} &= \sum_{i,i'=1}^{n} \frac{\partial G}{\partial K_{i,i'}}\frac{\partial K_{i,i'}}{\partial w_j} \\ &= \frac{1}{m}\sum_{i,i'=1}^{n} \frac{\partial G}{\partial K_{i,i'}} \frac{\Big(\operatorname{sign}(w_j^\top x_i)x_i + \operatorname{sign}(w_j^\top x_{i'})x_{i'} - \operatorname{sign}(w_j^\top(x_i - x_{i'}))(x_i - x_{i'})\Big)}{2}.\end{aligned} \tag{4.18}$$

We know that

$$\frac{\partial G}{\partial(\tilde{K} + \lambda nI)} = -\frac{\lambda}{2}(\tilde{K} + \lambda nI)^{-1}\tilde{Y}\tilde{Y}^\top(\tilde{K} + \lambda nI)^{-1},$$

thus

$$\begin{aligned}\frac{\partial G}{\partial K_{i,i'}} &= \sum_{l,k} \frac{\partial G}{\partial (\tilde{K} + \lambda n I)_{l,k}} \frac{\partial (\Pi K \Pi + \lambda n I)_{l,k}}{\partial K_{i,i'}} \\ &= \sum_{l,k} -\frac{\lambda}{2} ((\tilde{K} + \lambda n I)^{-1} \tilde{Y} \tilde{Y}^\top (\tilde{K} + \lambda n I)^{-1})_{l,k} \Pi_{l,i} \Pi_{i',k} \\ &= -\frac{\lambda}{2} \left(\Pi (\tilde{K} + \lambda n I)^{-1} \tilde{Y} \tilde{Y}^\top (\tilde{K} + \lambda n I)^{-1} \Pi \right)_{i,i'} \\ &= -\frac{\lambda}{2} (\Pi (\tilde{K} + \lambda n I)^{-1} \tilde{Y})_i (\Pi (\tilde{K} + \lambda n I)^{-1} \tilde{Y})_{i'}.\end{aligned}$$

Substituting this back into Equation (4.18) and introducing $S_j \in \mathbb{R}^{n \times n}$ with $(S_j)_{i,i'} = (\operatorname{sign}(w_j^\top x_i) x_i + \operatorname{sign}(w_j^\top x_{i'}) x_{i'} - \operatorname{sign}(w_j^\top (x_i - x_{i'}))(x_i - x_{i'}))/2$, we get

$$\begin{aligned}\frac{\partial G}{\partial w_j} &= -\frac{\lambda}{2m} \sum_{i,i'=1}^{n} (\Pi (\tilde{K} + \lambda n I)^{-1} \tilde{Y})_i (\Pi (\tilde{K} + \lambda n I)^{-1} \tilde{Y})_{i'} (S_j)_{i,i'} \\ &= -\frac{\lambda}{2m} \operatorname{tr} \left((\Pi (\tilde{K} + \lambda n I)^{-1} \tilde{Y})^\top S_j (\Pi (\tilde{K} + \lambda n I)^{-1} \tilde{Y}) \right) \\ &= -\frac{\lambda}{2m} \operatorname{tr} \left(((\tilde{K} + \lambda n I)^{-1} \tilde{Y})^\top \Pi S_j \Pi ((\tilde{K} + \lambda n I)^{-1} \tilde{Y}) \right).\end{aligned}$$

This implies that we can replace $(S_j)_{i,i'}$ with the i-th, i'-th component of any matrix with the same centred version, such as $\tilde{S}_j$ where $(\tilde{S}_j)_{i,i'} = -\operatorname{sign}(w_j^\top (x_i - x_i'))(x_i - x_i')$, yielding the desired result. □

A.4.3 Proof of Lemma 30

Proof of Lemma 30. We consider each penalty separately.

1. For $\Omega_{\text{basic}}(W) = \frac{1}{2m} \sum_{j=1}^{m} \|w_j\|$, the penalty corresponds to a group Lasso penalty on $W \in \mathbb{R}^{d \times m}$, where the groups are the columns. The proximal operator is given by:
$$(\operatorname{prox}_{\lambda\gamma\Omega}(W))_j = \left(1 - \frac{\lambda\gamma}{2m} \frac{1}{\|w_j\|}\right)_+ w_j,$$
as detailed in [Bach et al., 2012, Section 3.3].

2. For $\Omega_{\text{variable}}(W) = \frac{1}{2} \sum_{a=1}^{d} \left(\frac{1}{m} \sum_{j=1}^{m} |(w_j)_a|^2\right)^{1/2}$, this is a group Lasso setting where the groups are the rows of W. The proximal operator is:
$$(\operatorname{prox}_{\lambda\gamma\Omega}(w))^{(a)} = \left(1 - \frac{\lambda\gamma}{2\sqrt{m}} \frac{1}{\|W^{(a)}\|_2}\right)_+ W^{(a)},$$
also found in Bach et al. [2012, Section 3.3].

3. For $\Omega_{\text{feature}}(W) = \frac{1}{2} \operatorname{tr}\left(\left(\frac{1}{m} \sum_{j=1}^{m} w_j w_j^\top\right)^{1/2}\right)$, this penalty corresponds to a Lasso penalty on the singular values. Given $W = USV^\top$ (SVD), we have:
$$\operatorname{prox}_{\lambda\gamma\Omega}(W) = U\tilde{S}V^\top \quad \text{with} \quad \tilde{S} = \left(1 - \frac{\lambda\gamma}{2\sqrt{m}|S|}\right)_+ S,$$
using results from Bach et al. [2012, Section 3.3].

4. For $\Omega_{\text{concave variable}}(W) = \frac{1}{2s}\sum_{a=1}^{d}\log\left(1+s\left(\frac{1}{m}\sum_{j=1}^{m}|(w_j)_a|^2\right)^{1/2}\right)$, the loss is separable along the d dimensions. Considering each $W^{(a)}$ separately, we compute the proximal operator:

$$\text{prox}_{\frac{\lambda\gamma}{2s}\log(1+\frac{s}{\sqrt{m}}\|\cdot\|_2)}(W^{(a)}) = \min_{u^{(a)}\in\mathbb{R}^m}\frac{1}{2}\|W^{(a)}-u^{(a)}\|_2^2+\frac{\lambda\gamma}{2s}\log(1+\frac{s}{\sqrt{m}}\|u^{(a)}\|_2).$$

The subgradients of $\mathcal{L}(u^{(a)}) := \frac{1}{2}\|W^{(a)}-u^{(a)}\|_2^2+\frac{\lambda\gamma}{2s}\log(1+\frac{s}{\sqrt{m}}\|u^{(a)}\|_2)$ are:

$$\frac{\partial\mathcal{L}}{\partial u^{(a)}} = -(W^{(a)}-u^{(a)})+\frac{\lambda\gamma}{2s}\frac{s}{\sqrt{m}}\frac{1}{1+\frac{s}{\sqrt{m}}\|u^{(a)}\|_2}v^{(a)},$$

where $\|v^{(a)}\|_2\le 1$ if $u^{(a)}=0$, and otherwise $v^{(a)}=u^{(a)}/\|u^{(a)}\|_2$.
For $u^{(a)}\neq 0$, there is a scalar $c\in\mathbb{R}^+$ such that $u^{(a)}=cW^{(a)}$, yielding:

$$c\left(1+\frac{\lambda\gamma}{2\sqrt{m}}\frac{1}{c\|W^{(a)}\|_2}\frac{1}{1+\frac{sc}{\sqrt{m}}\|W^{(a)}\|_2}\right)=1.$$

This is a second-order polynomial in c that can be solved explicitly. The determinant Δ is

$$\Delta = (1-\frac{s}{\sqrt{m}}\|W^{(a)}\|_2)^2-4\left(\frac{\lambda\gamma}{2\sqrt{m}}\frac{1}{\|W^{(a)}\|_2}-1\right)\frac{s}{\sqrt{m}}\|W^{(a)}\|_2.$$

When $\Delta\le 0$, the proximal operator is $u^{(a)}=0$. Otherwise, it suffices to compare the two possible values of c and choose the one for which $\mathcal{L}$ is the smallest.

5. For $\Omega_{\text{concave feature}}(W) = \frac{1}{2s}\sum_{a=1}^{d}\log\left(1+\frac{s}{\sqrt{m}}\sigma_a(w_1,\dots,w_n)\right)$, we combine the results of the third and fourth items above. The proximal operator is

$$\text{prox}_{\lambda\gamma\Omega}(W) = U\tilde{S}V^\top,$$

where $\tilde{S}$ is obtained by replacing all $\|W^{(a)}\|_2$ by σ_a in the computations of the proximal of $\Omega_{\text{concave variable}}$.

□

A.5 Extra Lemma and Proofs Related to Section 4 Except Section 4.2

Here we provided the proofs of the lemmas used to bound the Gaussian complexity.

A.5.1 Proof of Lemma 31

Proof of Lemma 31. Recall that

$$G_n(\{f\in\mathcal{F}_\infty,\Omega(f)\le D\}) = \mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega_0(f)\le D,c\le D}\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i f(x_i)\right).$$

We start by considering the expectation over ε only. Using the definitions, we obtain

$$\begin{aligned}
&\mathbb{E}_\varepsilon\Big(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D}\frac{1}{n}\sum_{i=1}^n\varepsilon_i f(x_i)\Big)\\
&\quad=\mathbb{E}_\varepsilon\Big(\sup_{|c|\leq D,\int_{\mathcal{S}^{d-1}}\|g_w\|_{\mathcal{H}}\mathrm{d}\mu(w)\leq D}\frac{1}{n}\sum_{i=1}^n\varepsilon_i\left(c+\int_{\mathcal{S}^{d-1}}g_w(w^\top x_i)\mathrm{d}\mu(w)\right)\Big)\\
&\quad=\mathbb{E}_\varepsilon\Big(D\frac{1}{n}\Big|\sum_{i=1}^n\varepsilon_i\Big|+\sup_{\int_{\mathcal{S}^{d-1}}\|g_w\|_{\mathcal{H}}\mathrm{d}\mu(w)\leq D}\frac{1}{n}\sum_{i=1}^n\varepsilon_i\int_{\mathcal{S}^{d-1}}\langle g_w,k^{(B)}_{w^\top x_i}\rangle\mathrm{d}\mu(w)\Big)\\
&\quad\leq D\frac{1}{\sqrt{n}}+\mathbb{E}_\varepsilon\Big(\sup_{\int_{\mathcal{S}^{d-1}}\|g_w\|_{\mathcal{H}}\mathrm{d}\mu(w)\leq D}\int_{\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^n\varepsilon_i\langle g_w,k^{(B)}_{w^\top x_i}\rangle\mathrm{d}\mu(w)\Big)
\end{aligned}$$

For the second term of the equation right above, we then have equality to

$$\begin{aligned}
&=\mathbb{E}_\varepsilon\Big(\sup_{\int_{\mathcal{S}^{d-1}}\|g_w\|_{\mathcal{H}}\mathrm{d}\mu(w)\leq D}\int_{\mathcal{S}^{d-1}}|\frac{1}{n}\sum_{i=1}^n\varepsilon_i\langle g_w,k^{(B)}_{w^\top x_i}\rangle|\mathrm{d}\mu(w)\Big)\\
&=\mathbb{E}_\varepsilon\Big(\sup_{\int_{\mathcal{S}^{d-1}}\|g_w\|_{\mathcal{H}}\mathrm{d}\mu(w)\leq D}\sup_{w\in\mathcal{S}^{d-1}}|\frac{1}{n}\sum_{i=1}^n\varepsilon_i\langle g_w,k^{(B)}_{w^\top x_i}\rangle|\Big)\\
&=\mathbb{E}_\varepsilon\Big(\sup_{w\in\mathcal{S}^{d-1},\|g\|_{\mathcal{H}}\leq D}|\frac{1}{n}\sum_{i=1}^n\varepsilon_i\langle g,k^{(B)}_{w^\top x_i}\rangle|\Big)\\
&=\mathbb{E}_\varepsilon\Big(\sup_{w\in\mathcal{S}^{d-1},\|g\|_{\mathcal{H}}\leq D}|\langle g,\frac{1}{n}\sum_{i=1}^n\varepsilon_i k^{(B)}_{w^\top x_i}\rangle|\Big)=\mathbb{E}_\varepsilon\Big(\sup_{w\in\mathcal{S}^{d-1},\|g\|_{\mathcal{H}}\leq D}\langle g,\frac{1}{n}\sum_{i=1}^n\varepsilon_i k^{(B)}_{w^\top x_i}\rangle\Big)\\
&=D\mathbb{E}_\varepsilon\Big(\sup_{w\in\mathcal{S}^{d-1},\|g\|_{\mathcal{H}}\leq 1}\frac{1}{n}\sum_{i=1}^n\varepsilon_i\langle g,k^{(B)}_{w^\top x_i}\rangle\Big)=D\mathbb{E}_\varepsilon\Big(\sup_{w\in\mathcal{S}^{d-1},\|g\|_{\mathcal{H}}\leq 1}\frac{1}{n}\sum_{i=1}^n\varepsilon_i g(w^\top x_i)\Big).
\end{aligned}$$

Taking the expectation over the data set on both sides yields the desired result. □

A.5.2 Lemma 34 and its Proof

This lemma provides an explicit formula for computing the supremum over functions within the unit ball of $\mathcal{H}$, which we can then use for the calculation of Gaussian complexity.

Lemma 34 (Optimal g in Gaussian Complexity)**.** *For any data set $(x_1,\ldots,x_n)$, $w\in\mathbb{R}^d$, with $K^{(w)}\in\mathbb{R}^{n\times n}$ the kernel matrix of $k^{(B)}$ with the data projected on w and $\varepsilon\in\mathbb{R}^n$,*

$$\sup_{\|g\|_{\mathcal{H}}\leq 1}\frac{1}{n}\sum_{i=1}^n\varepsilon_i g(w^\top x_i)=\frac{1}{n}\sqrt{\varepsilon^\top K^{(w)}\varepsilon}$$

Proof of Lemma 34. By applying the definitions, we obtain

$$\begin{aligned}
\sup_{\|g\|_{\mathcal{H}}\leq 1}\frac{1}{n}\sum_{i=1}^n\varepsilon_i g(w^\top x_i)&=\sup_{\|g\|_{\mathcal{H}}\leq 1}\frac{1}{n}\sum_{i=1}^n\varepsilon_i\langle g,k^{(B)}_{w^\top x_i}\rangle=\sup_{\|g\|_{\mathcal{H}}\leq 1}\langle g,\frac{1}{n}\sum_{i=1}^n\varepsilon_i k^{(B)}_{w^\top x_i}\rangle\\
&=\frac{1}{n}\left\langle\frac{\sum_{i=1}^n\varepsilon_i k^{(B)}_{w^\top x_i}}{\|\sum_{i=1}^n\varepsilon_i k^{(B)}_{w^\top x_i}\|_{\mathcal{H}}},\sum_{j=1}^n\varepsilon_j k^{(B)}_{w^\top x_j}\right\rangle\\
&=\frac{1}{n}\|\sum_{i=1}^n\varepsilon_i k^{(B)}_{w^\top x_i}\|_{\mathcal{H}}=\frac{1}{n}\sqrt{\varepsilon^\top K^{(w)}\varepsilon},
\end{aligned}$$

which is the desired result. □

A.5.3 Proof of Lemma 32

Proof of Lemma 32. Define g_ζ such that $g_\zeta(0) = 0$ and $g'_\zeta(x) = \min(|g'(x)|, 1/\zeta)\operatorname{sign}(g'(x))$. Note that $\|g'_\zeta\|_\infty \leq \frac{1}{\zeta}$, thus g_ζ is $\frac{1}{\zeta}$-Lipschitz. Additionally, for any $a \in \mathbb{R}$,

$$\begin{aligned}
|g_\zeta(a) - g(a)| &= \left| \int_0^a (g'_\zeta(t) - g'(t))\,\mathrm{d}t \right| \\
&\leq \int_0^a \left| g'_\zeta(t) - g'(t) \right| \mathrm{d}t \leq \int_{-\infty}^{+\infty} \mathbb{1}_{|g'(t)|\geq 1/\zeta} (|g'(t)| - 1/\zeta)\,\mathrm{d}t \\
&\leq \int_{-\infty}^{+\infty} \mathbb{1}_{|g'(t)|\geq 1/\zeta} |g'(t)|\,\mathrm{d}t \leq \int_{-\infty}^{+\infty} \zeta |g'(t)|^2\,\mathrm{d}t \\
&\leq \zeta \quad \text{since} \quad \int_{-\infty}^{+\infty} (g'(t))^2\,\mathrm{d}t \leq 1,
\end{aligned}$$

yielding the desired result. □

A.5.4 Lemma 35 and its Proof

Lemma 35 (Gaussian Complexity of Finite Set of Lipschitz Functions)**.** *Let $h_1, \dots h_M$ be 1-Lipschitz functions from $\mathbb{R}$ to $\mathbb{R}$ and let ε be a random centred Gaussian vector with identity covariance matrix. Then*

$$\begin{aligned}
&\mathbb{E}_\varepsilon \left(\sup_{h\in\{h_1,\dots,h_M\}, w\in\mathcal{S}^{d-1}} \frac{1}{n} \sum_{i=1}^n \varepsilon_i h(w^\top x_i) \right) \\
&\qquad \leq \mathbb{E}_\varepsilon \left(\left\| \frac{\sqrt{2}}{n} \sum_{i=1}^n \varepsilon_i x_i \right\|^* + \sqrt{8 \frac{\sum_{i=1}^n (\|x_i\|^*)^2}{n^2}} \sqrt{2 \log M} \right).
\end{aligned}$$

This lemma is inspired by Bartlett and Mendelson [2002].

Proof of Lemma 35. We use Slepian's lemma [Ledoux and Talagrand, 1991, Corollary 3.14]. For $h \in \{h_1, \dots h_M\}, w \in \mathcal{S}^{d-1}$, let

$$X_{h,w} := \frac{1}{n} \sum_{i=1}^n \varepsilon_i h(w^\top x_i) \text{ and } Y_{h,w} = \frac{\sqrt{2}}{n} \sum_{i=1}^n \varepsilon_i w^\top x_i + \sum_{j=1}^M \mathbb{1}_{h=h_j} \tilde{\varepsilon}_j \sqrt{8 \frac{\sum_{i=1}^n (\|x_i\|^*)^2}{n^2}},$$

where $\tilde{\varepsilon}$ is a centred Gaussian vector with identity covariance matrix independent of ε. Notice that for $h, \tilde{h} \in \{h_1, \dots h_M\}, w, \tilde{w} \in \mathcal{S}^{d-1}$, we have

$$\begin{aligned}
\mathbb{E}_\varepsilon((X_{h,w} - X_{\tilde{h},\tilde{w}})^2) &= \frac{1}{n^2} \sum_{i=1}^n (h(w^\top x_i) - \tilde{h}(\tilde{w}^\top x_i))^2 \\
&\leq \frac{1}{n^2} \sum_{i=1}^n (h(w^\top x_i) - h(\tilde{w}^\top x_i) + h(\tilde{w}^\top x_i) - \tilde{h}(\tilde{w}^\top x_i))^2 \\
&\leq \frac{2}{n^2} \sum_{i=1}^n (h(w^\top x_i) - h(\tilde{w}^\top x_i))^2 + (h(\tilde{w}^\top x_i) - \tilde{h}(\tilde{w}^\top x_i))^2.
\end{aligned}$$

We can then deal with the two terms separately. For the left term, the fact that h is 1-Lipschitz yields that

$$\frac{2}{n^2}\sum_{i=1}^{n}(h(w^\top x_i) - h(\tilde{w}^\top x_i))^2 \leq \frac{2}{n^2}\sum_{i=1}^{n}(w^\top x_i - \tilde{w}^\top x_i)^2.$$

Then, using the fact that $h - \tilde{h}$ is 2-Lipschitz and $h(0) = \tilde{h}(0) = 0$, we have

$$\begin{aligned}\frac{2}{n^2}\sum_{i=1}^{n}(h(\tilde{w}^\top x_i) - \tilde{h}(\tilde{w}^\top x_i))^2 &= \frac{2}{n^2}\sum_{i=1}^{n}(h(\tilde{w}^\top x_i) - \tilde{h}(\tilde{w}^\top x_i) - (h(0) - \tilde{h}(0))^2 \\ &\leq \frac{2}{n^2}\sum_{i=1}^{n}\mathbb{1}_{h\neq h'}4(w^\top x_i)^2 \leq \mathbb{1}_{h\neq\tilde{h}}\frac{8}{n^2}\sum_{i=1}^{n}(\|x_i\|^*)^2.\end{aligned}$$

All in all $\mathbb{E}_\varepsilon((X_{h,w} - X_{\tilde{h},\tilde{w}})^2) \leq \mathbb{E}_\varepsilon((Y_{h,w} - Y_{\tilde{h},\tilde{w}})^2)$ therefore we can apply Slepian's lemma and obtain

$$\begin{aligned}&\mathbb{E}_\varepsilon\left(\sup_{h\in\{h_1,\dots,h_M\},w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i h(w^\top x_i)\right) \\ &\leq \mathbb{E}_\varepsilon\left(\sup_{h\in\{h_1,\dots,h_M\},w\in\mathcal{S}^{d-1}}\frac{\sqrt{2}}{n}\sum_{i=1}^{n}\varepsilon_i w^\top x_i + \sum_{j=1}^{M}\tilde{\varepsilon}_j\mathbb{1}_{h=h_j}\sqrt{8\frac{\sum_{i=1}^{n}(\|x_i\|^*)^2}{n^2}}\right).\end{aligned}$$

We then remark that the first term of the expectation does not depend on h and that we can take the supremum over the sphere explicitly, while the second term does not depend on w and we can also take the supremum over $\{h_1,\dots,h_M\}$ explicitly

$$\begin{aligned}&\mathbb{E}_\varepsilon\left(\sup_{h\in\{h_1,\dots,h_M\},w\in\mathcal{S}^{d-1}}\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i h(w^\top x_i)\right) \\ &\leq \mathbb{E}_\varepsilon\left(\sup_{w\in\mathcal{S}^{d-1}}\frac{\sqrt{2}}{n}\sum_{i=1}^{n}\varepsilon_i w^\top x_i + \sup_{h\in\{h_1,\dots,h_M\}}\sum_{j=1}^{m}\tilde{\varepsilon}_j\mathbb{1}_{h=h_j}\sqrt{8\frac{\sum_{i=1}^{n}(\|x_i\|^*)^2}{n^2}}\right) \\ &\leq \mathbb{E}_\varepsilon\left(\sup_{w\in\mathcal{S}^{d-1}}\frac{\sqrt{2}}{n}\sum_{i=1}^{n}\varepsilon_i w^\top x_i + \sup_{j\in[M]}\tilde{\varepsilon}_j\sqrt{8\frac{\sum_{i=1}^{n}(\|x_i\|^*)^2}{n^2}}\right) \\ &\leq \mathbb{E}_\varepsilon\left(\left\|\frac{\sqrt{2}}{n}\sum_{i=1}^{n}\varepsilon_i x_i\right\|^* + \sqrt{8\frac{\sum_{i=1}^{n}(\|x_i\|^*)^2}{n^2}}\sqrt{2\log M}\right).\end{aligned}$$

□

A.5.5 Proof of Lemma 33

Proof of Lemma 33. We begin with the bounded case. The bounds on the expectations are clearly valid. Then, since $1+\sqrt{\|X\|^*}$ is a bounded variable, it is necessarily subgaussian with a variance proxy bounded by $\frac{(1+\sqrt{R})^2}{2\log(2)} \leq (1+\sqrt{R})^2$ [Vershynin, 2018, Proposition 2.5.2 (iv)].

Next, we consider the subgaussian case. Using the Cauchy-Schwarz inequality, we

handle the case where $\|\cdot\|^* = \|\cdot\|_2$ using Vershynin [2018, Proposition 2.5.2]

$$\sqrt{\mathbb{E}_X(\|X\|_2)} \leq \left(\mathbb{E}_X(\|X\|_2^2)\right)^{1/4} \leq \sqrt{6}\left(\sum_{a=1}^{d} \sigma_a^2\right)^{1/4}.$$

For the $\|\cdot\|_\infty$ case, applying Vershynin [2018, Exercise 2.5.10] with the constant made explicit yields the desired result.

For the second expectation with $\|\cdot\|^* = \|\cdot\|_2$, we have

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]} \|X_i\|_2^2\right) &= \mathbb{E}\max_{i\in[n]} \sum_{a=1}^{d} ((X_i)_a)^2 \leq \sum_{a=1}^{d} \mathbb{E}\max_{i\in[n]} \left((X_i)_a\right)^2 \\ &\leq \sum_{a=1}^{d} \frac{1}{t} \log\left(\mathbb{E}\left(e^{t\max_{i\in[n]}((X_i)_a)^2}\right)\right) \leq \sum_{a=1}^{d} \frac{1}{t} \log\left(n\mathbb{E}\left(e^{t((X_i)_a)^2}\right)\right),\end{aligned}$$

for all $t > 0$. We can then bound this by $\sum_{a=1}^{d} \frac{1}{t}\log(ne^{t(6\sqrt{2e}\sigma_a)^2})$ for $t < 1/(6\sqrt{2e}\sigma_a)^2$, yielding:

$$\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]} \|X_i\|_2^2\right) \leq 72e(1+\log(n)) \sum_{a=1}^{d} \sigma_a^2.$$

The same proof technique applies to $\mathbb{E}_{\mathcal{D}_n}\left(\max_{i\in[n]} \|X_i\|_\infty^2\right)$, yielding the desired result.

Finally, we consider the subgaussianity of $1 + \sqrt{\|X\|^*}$. Note that the sum of two subgaussian variables is subgaussian. Using Vershynin [2018, Proposition 2.5.2 (ii)], for two real random variables Z and $\tilde{Z}$ with variance proxies σ^2 and $\tilde{\sigma}^2$ respectively, we have that $Z + \tilde{Z}$ is subgaussian with variance proxy $(\sigma + \tilde{\sigma})^2$. Additionally, the absolute value of a subgaussian variable is also subgaussian with the same variance proxy [Vershynin, 2018, Proposition 2.5.2].

For $\|\cdot\| = \|\cdot\|_2$, we have $1 + \sqrt{\|X\|_2} \leq 1 + \sum_{a=1}^{d} |X_a|$. Since 1 and X_a are subgaussian variables, this yields the desired result.

For $\|\cdot\|_\infty$, for all $t > 0$,

$$\begin{aligned}\mathbb{P}(\|X\|_\infty \geq \sqrt{2\sigma^2\log(2d)} + t) &\leq 2de^{-\frac{(\sqrt{2\sigma^2\log(2d)}+t)^2}{2\sigma^2}} \\ &\leq 2e^{-\frac{t^2}{2\sigma^2} - \frac{t\sqrt{\log(2d)}}{\sqrt{2\sigma^2}}} \leq 2e^{-\frac{t^2}{2\sigma^2}}.\end{aligned}$$

Thus, $\|X\|_\infty - \sqrt{2\sigma^2\log(2d)}$ is subgaussian with variance proxy σ^2. Therefore, $\|X\|_\infty$ is subgaussian with variance proxy bounded by $\sigma^2(1 + \sqrt{\log(2d)})^2$. Then, $1 + \sqrt{\|X\|_\infty}$ is subgaussian because it is less than $2 + \|X\|_\infty$, which is subgaussian [Vershynin, 2018, Proposition 2.5.2] with a variance proxy bounded by that of $2 + \|X\|_\infty$, yielding the desired result. □

A.6 Lemmas Needed for Section 4.2 and their Proofs

Here we provide lemmas necessary for the proof of Theorem 9 and the analysis of its distribution-dependent terms.

A.6.1 Lemma 36 and its Proof

Lemma 36 relates the Gaussian complexity to useful quantities to bound the expected risk.

Lemma 36. *(Use of Gaussian Complexity) Let $D > 0$ and the data set $\mathcal{D}_n = (x_i, y_i)_{i\in[n]}$ consists of i.i.d. samples of the random variable $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. Assume that the loss ℓ is L-Lipschitz in its second (bounded) argument, i.e., $\forall y \in \mathcal{Y}, a \in \{f(x) \mid x \in \mathcal{X},\ f \in \mathcal{F}_\infty,\ \Omega(f) \leq D\}, a \to \ell(y, a)$ is L-Lipschitz. Then, we have*

$$\mathbb{E}_{\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) + \sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right) \leq 6DL\left(\frac{1}{\sqrt{n}} + G_n\right).$$

Proof of Lemma 36. By Bach [2024, Proposition 4.2], we have

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) + \sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right)\\ &\leq 4\mathbb{E}_{\tilde{\varepsilon},\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i \ell(y_i, f(x_i))\right),\end{aligned}$$

where $\tilde{\varepsilon}$ consists of i.i.d. Rademacher variables.

Next, applying the contraction principle from Bach [2024, Proposition 4.3], we get

$$\mathbb{E}_{\tilde{\varepsilon},\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i \ell(y_i, f(x_i))\right) \leq \mathbb{E}_{\tilde{\varepsilon},\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \frac{1}{n}\sum_{i=1}^n \tilde{\varepsilon}_i f(x_i)\right).$$

Then, using Wainwright [2019, Exercise 5.5], we have

$$\mathbb{E}_{\tilde{\varepsilon},\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \frac{1}{n}\sum_{i=1}^n \tilde{\varepsilon}_i f(x_i)\right) \leq \sqrt{\frac{\pi}{2}}\mathbb{E}_{\varepsilon,\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(x_i)\right),$$

where $\varepsilon \sim \mathcal{N}(0, I_d)$.

Finally, by applying Lemma 31 and combining all these results, we obtain the desired inequality

$$\mathbb{E}_{\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) + \sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right) \leq 6DL\left(\frac{1}{\sqrt{n}} + G_n\right).$$

□

A.6.2 Lemma 37 and its Proof

Lemma 37 describes a useful property on the expectation of the hyperbolic cosine of a subgaussian random variable.

Lemma 37. *(Technical Lemma on Subgaussian Random Variables) Let Z be a real-valued random variable (not necessarily centred) that is subgaussian (see Definition 7.) Then, for all $\lambda \in \mathbb{R}$,*

$$\mathbb{E}\left(\cosh(\lambda Z)\right) \leq e^{(6\sqrt{2e})^2\sigma^2\lambda^2}.$$

Proof of Lemma 37. An equivalent definition of subgaussianity is that for all $\lambda \in \mathbb{R}$, if $6\sqrt{2e}\sigma|\lambda| \leq 1$, then $\mathbb{E}(e^{\lambda^2 Z^2}) \leq e^{(6\sqrt{2e})^2\sigma^2\lambda^2}$, see Vershynin [2018, Proposition 2.5.2].

First, in the case $|\lambda| \leq \frac{1}{6\sqrt{2e}\sigma}$. Using the inequality $e^x \leq x + e^{x^2}$ for all $x \in \mathbb{R}$, we get

$$\mathbb{E}\left(\cosh(\lambda Z)\right) \leq \mathbb{E}\left(\frac{\lambda Z + e^{\lambda^2 Z^2} - \lambda Z + e^{\lambda^2 Z^2}}{2}\right) = \mathbb{E}\left(e^{\lambda^2 Z^2}\right) \leq e^{(6\sqrt{2e})^2\sigma^2\lambda^2}.$$

Next, consider the case $|\lambda| \geq \frac{1}{6\sqrt{2e}\sigma}$. We can bound the expectation as follows

$$\mathbb{E}\left(\cosh(\lambda Z)\right) \leq \mathbb{E}\left(e^{|\lambda Z|}\right) = \mathbb{E}\left(e^{6\sqrt{2e}\sigma|\lambda|\frac{|Z|}{6\sqrt{2e}\sigma}}\right) \leq \mathbb{E}\left(e^{(6\sqrt{2e})^2\sigma^2\lambda^2/2+\frac{Z^2}{2(6\sqrt{2e})^2\sigma^2}}\right)$$
$$\leq e^{(6\sqrt{2e})^2\sigma^2\lambda^2/2}e^{1/2} \leq e^{(6\sqrt{2e})^2\sigma^2\lambda^2},$$

where we use the fact that $(6\sqrt{2e})^2\sigma^2\lambda^2 \geq 1$ to justify the final inequality.

Thus, in both cases, we have shown that $\mathbb{E}\left(\cosh(\lambda Z)\right) \leq e^{(6\sqrt{2e})^2\sigma^2\lambda^2}$, proving the lemma. □

A.6.3 Lemma 38 and its Proof

Lemma 38 is an application of McDiarmid's inequality (a specific version by Meir and Zhang [2003] for subgaussian random variables) to our learning problem.

Lemma 38. *(Use of McDiarmid's Inequality) Let $D > 0$ and $\delta \in (0,1)$. Assume that $1 + \sqrt{\|X\|^*}$ is subgaussian with variance proxy σ^2 and that the loss ℓ is L-Lipschitz in its second (bounded) argument, i.e., $\forall y \in \mathcal{Y}, a \in \{f(x) \mid x \in \mathcal{X},\ f \in \mathcal{F}_\infty,\ \Omega(f) \leq D\}, a \to \ell(y,a)$ is L-Lipschitz. Then, with probability greater than $1-\delta$,*

$$\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) + \sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)$$
$$\leq \mathbb{E}_{\mathcal{D}_n}\left(\sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \widehat{\mathcal{R}}(f) - \mathcal{R}(f) + \sup_{f\in\mathcal{F}_\infty,\Omega(f)\leq D} \mathcal{R}(f) - \widehat{\mathcal{R}}(f)\right)$$
$$+ \frac{48\sqrt{2e}LD\sigma}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}.$$

Proof of Lemma 38. We use a specific version of McDiarmid's inequality [Meir and Zhang, 2003, Theorem 3]. First, we show that the conditions for applying the theorem are met. Let $\tilde{H} := \{h : (x,y) \in \mathcal{X}\times\mathcal{Y} \to \ell(y,f(x)) - \ell(y,\tilde{f}(x)) \mid \Omega(f) \leq D, \Omega(\tilde{f}) \leq D\}$. For any $\lambda > 0$, we have

$$\mathbb{E}_{X,Y}\left(\sup_{h,\tilde{h}\in\tilde{H}} \cosh(2\lambda(h(X,Y) - \tilde{h}(X,Y)))\right)$$
$$= \mathbb{E}_{X,Y}\left(\sup_{f,\Omega(f)\leq D,\tilde{f},\Omega(\tilde{f})\leq D} \cosh(2\lambda(\ell(Y,f(X)) - \ell(Y,\tilde{f}(X))))\right)$$
$$\leq \mathbb{E}_{X,Y}\left(\sup_{f,\Omega(f)\leq D,\tilde{f},\Omega(\tilde{f})\leq D} \cosh(2\lambda L|f(X) - \tilde{f}(X)|)\right)$$
$$\leq \mathbb{E}_{X,Y}\left(\sup_{f,\Omega(f)\leq D,\tilde{f},\Omega(\tilde{f})\leq D} \cosh(4\lambda LD(1+\sqrt{\|X\|^*}))\right)$$
$$= \mathbb{E}_{X,Y}\left(\cosh(4\lambda LD(1+\sqrt{\|X\|^*}))\right) \leq e^{(48\sqrt{e})^2L^2D^2\sigma^2\lambda^2/2},$$

where the last inequality follows from Lemma 37. Hence, the condition is verified with $M = 48\sqrt{e}LD\sigma$ and applying Meir and Zhang [2003, Theorem 3] yields the desired result. □

B Numerical Experiments

In this section, we detail the parameters and methodology used in the different experiments. The code needed to run the experiments can be found at https://github.com/BertilleFollain/BKerNN.

B.1 Experiment 1: Optimisation procedure, Importance of Positive Homogeneous Kernel

Each method was tuned using 5-fold cross-validation with grid search, using negative mean squared error as the scoring metric. The training was set for 20 iterations and the step-size parameter (γ) was set to 500, with backtracking enabled. Regularisation parameter candidates were $\lambda = \{0.05, 0.1, 0.5, 1, 1.5\} \times 2 \max_{i\in[n]} \|x_i\|_2/n$. Once the regularisation parameters had been selected, we trained from scratch for 200 iterations, with the other parameters kept as before.

B.2 Experiments 2 & 3: Influence of Parameters (Number of Particles m, Regularisation Parameter λ, and Type of Regularisation)

For Experiment 2, in the first subplot, we set the step-size parameter γ to 500 and the number of iterations to 50. The regularisation type was set to Ω_{basic} and the regularisation parameter to $\lambda = 0.02$. The tested values of m were 1, 3, 5, 7, 10, 15, 20, 30, 40, and 50.

In the second subplot, we varied the regularisation parameter λ in 0.0005, 0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.3, and 0.5, while keeping the number of particles fixed at $m = 10$.

In Experiment 3, the BKerNN model was instantiated with a fixed number of particles $m = 20$, step-size parameter $\gamma = 500$, and number of iterations 25. The regularisation parameter λ was set as $2 \max_{i\in n} \|x_i\|_2/n$.

B.3 Experiment 4: Comparison to Neural Network on 1D Examples, Influence of Number of Particles/Width of Hidden Layer m

In Experiment 4, we investigated the performance of two learning methods, BKerNN and ReLUNN, on three different 1D functions. The training set always consists of 128 samples, with x sampled uniformly between -1 and 1, while the target function/test set without noise consists of 1024 equally spread out points. The response was then generated as follows. For the first function, $y = \sin(2\pi x) + \text{noise}$, for the second $y = \text{sign}(\sin(2\pi x)) + \text{noise}$, for the third $y = 4|x + 1 - 0.25 - \lfloor x + 1 - 0.25 \rfloor - 0.5| - 1 + \text{noise}$, where the noise is always normal, centred and with standard deviation equal to 0.2. For BKerNN, the regularisation parameter λ was selected from [0.005, 0.01, 0.02, 0.05] using 5-fold cross-validation and the negative mean squared error score. ReLUNN was trained using a batch size of 16, a number of iterations equal to 400,000 and a step-size of 0.005.

B.4 Experiment 5: Prediction Score and Feature Learning Score Against Growing Dimension and Sample Size, a Comparison of BKerNN with Kernel Ridge Regression and a ReLU Neural Network

In Experiment 5, data sets were generated with input data uniformly sampled within the hypercube $[-1, 1]^d$. The feature matrix P was generated from the orthogonal group. For each configuration, training and test sets of sizes n and $n_{\text{test}} = 201$, respectively, were created. Output labels y were computed as $y_i = |\sum_{a=1}^{k}(\sin((P^\top x_i)_a))|$, with $k = 3$ relevant features.

The first two plots fixed the dimension at 15 and varied sample sizes across [10, 20, 50, 100, 150, 200, 300, 400, 500]. The last two plots fixed the sample size at 212 and varied dimensions across [3, 5, 10, 20, 30, 40, 50]. Each configuration was repeated 10 times with different random seeds.

For BKerNN: λ was set to $2\max_{i\in[n]}(\|x_i\|_2)/n$, the number of particles was $m = 50$, the regularisation type Ω_{feature}, the number of iterations 20, and step-size $\gamma = 500$ with backtracking line search. For BKRR, λ was chosen similarly to BKerNN. For ReLUNN, the number of neurons was set to 50, learning rate to 0.05, batch size to 16, and number of iterations to 1500.

B.5 Experiment 6: Comparison on Real Data Sets Between BKerNN, Kernel Ridge Regression and a ReLU Neural Network

In Experiment 6, BKRR and both versions of BKerNN had regularisation parameter fixed equal to $\max_{i\in[n]}(\|x_i\|_2)/n$, where n is the number of training samples (i.e. 400). Backtracking line search was used for BKerNN and the starting step-size was 500, while the number of iterations was 40. For ReLUNN, the batch size was 16, while the number of iterations was 2500 which corresponds to 100 epochs, and the step-size was set to 0.01.

Conclusion

In this thesis, we addressed the challenge of supervised learning, where it is assumed that the prediction function is composed of a non-parametric function and hidden linear features, as described by the multi-index model. Our research was dedicated to developing algorithmic methods that enhance learning in this context, with the added benefit that these methods can be adapted to the variable selection setting. We exclusively employed the regularised empirical risk minimisation framework, given its versatility across various learning problems where a risk can be defined, thus possibly extending its applicability beyond standard supervised learning. We focused on making minimal assumptions about the data-generating process and ensuring our algorithms were computationally feasible. Moreover, we sought to provide explicit convergence rates that do not suffer from exponential dependency on the original data dimension and to limit the assumptions about the true prediction function.

Overall, our work highlights the potential of leveraging hidden linear structures within data to significantly enhance the learning process within a regularised empirical risk minimisation framework. By exploring these structures, we lay the groundwork for developing more computationally and statistically efficient methods that can be applied to a broader range of problems. Below, we briefly summarise the contributions of each chapter.

In Chapter 2, we introduced KTNGrad. This method employs regularised empirical risk minimisation within a reproducing kernel Hilbert space (RKHS) that includes the partial derivatives of the kernel, with regularisation applied through a trace norm penalty on the sample matrix of gradients. The optimisation procedure for KTNGrad is based on a convex problem with an explicit convergence rate, though it is computationally intensive. Our theoretical analysis of well-specified settings demonstrated that KTNGrad achieves convergence rates for the expected risk that do not exponentially depend on the data dimension. While the method reliably recovers the underlying features in a safe filter manner, it does so without explicit rates and without recovering the dimensionality of the hidden linear features. Numerical experiments confirmed that KTNGrad is competitive with the state-of-the-art method MAVE in estimating the features when the dimension is known. However, further enquiry showed that the assumption for the true regression function to belong to usual RKHS is not compatible with the multi-index model, which led us to study a broader space of functions that is better adapted to both the multi-index model and the variable selection framework in the next chapter.

Indeed, in Chapter 3, we introduced RegFeaL, a method that leverages the decomposition of functions in an orthonormal Hermite polynomials basis of the Hilbert space

of square-integrable function w.r.t. the multidimensional normal measure. This approach provides a clear interpretation of dependency on select variables or linear projections through the basis coefficients, which allows the introduction of a derivatives-based overlapping group LASSO penalty. By exploiting the orthogonality and rotational invariance of Hermite polynomials, RegFeaL iteratively aligns the data with the leading directions through an alternative minimisation process. However, the method requires a complex sampling technique to approximate the kernel at each optimisation step, which adds computational difficulty. Despite this, and without making strong assumptions about the true regression function, we demonstrated that the expected risk converges to the minimal risk, albeit with an exponential dependency on the data dimension. Numerical experiments showcased the ability of RegFeaL to effectively learn features in a similar manner to MAVE in some settings. Nonetheless, using an infinite-dimensional basis introduces computational burdens due to the sampling process and results in exponential terms in the theoretical analysis. Consequently, in the next chapter, we explored an alternative infinite-dimensional function space, which, while also computationally intractable due to its definition using an integral, is easily approximated by particles with theoretical justifications.

Consequently, in Chapter 4, we introduced BKerNN, a novel approach combining the strengths of kernel methods and neural networks. Our method represents functions as expectations over Sobolev spaces across all possible one-dimensional projections of the data, bridging the gap between kernel ridge regression and infinite-width one-hidden-layer neural networks. By leveraging the positive homogeneity of the Brownian kernel, we developed a principled optimisation procedure using particles to approximate the expectation. Theoretical analysis shows that BKerNN achieves convergence of the expected risk to the minimal risk in well-specified settings, with rates that are independent of the data dimension up to logarithmic factors, though this comes at the cost of increased sensitivity to sample size. We briefly discussed the adaptivity of the method to misspecified settings. Extensive experiments on both simulated and real datasets demonstrated BKerNN's superior performance compared to traditional kernel ridge regression and its competitive advantage over neural networks with ReLU activations.

Throughout this thesis, a consistent theme has been the careful design of an appropriate function space for non-parametric supervised learning with hidden linear features. This journey led us through various infinite-dimensional spaces, starting with reproducing kernel Hilbert spaces, progressing to Hilbert spaces with a Hermite polynomials basis, and ultimately culminating in a non-Hilbertian space.

We now present some perspectives of interest related to the presented work.

Computational savings for BKerNN from Chapter 4. Although already applicable to relatively high-dimensional settings, as seen in the numerical experiments of Chapter 4, the complexity of BKerNN remains high: $O(n^3 + mn^2d + md\min(m, d))$ (with m the number of particles). Computational savings could be obtained through techniques such as the Nyström approximation [Drineas and Mahoney, 2005, Rudi et al., 2015] or random features, which have been routinely used in the context of kernel methods, for example by Rahimi and Recht [2007]. This would extend the scope of applications of BKerNN to higher-dimensional contexts where it is most relevant.

Explicit adaptivity results for BKerNN from Chapter 4. We have briefly discussed the adaptivity of BKerNN to hidden linear features in misspecified settings. This aspect is crucial because, in practical scenarios, we cannot be certain whether we op-

erate within a well-specified framework. The adaptivity of BKerNN arises from the fact that, under the standard multi-index model ($f^* = g^*(P^\top \cdot)$), the norm inequality $\Omega_0(f^*) \leq \Omega_0(g^*)$ holds, which implies that the estimation of f^* benefits from the reduced dimensionality of g^*. However, this relationship is not entirely explicit. Future research could aim to develop more explicit theoretical results on the adaptivity of BKerNN to latent linear variables, building on the work done for neural networks by Bach [2024, Chapter 9].

Extensions to other function estimation problems with latent linear features. In this work, we focused on supervised learning with covariate/response pairs, where the goal is to infer the relationship between them. However, by employing the regularised empirical risk minimisation framework, we have developed methods that remain relevant whenever a risk function can be defined. This includes areas like control theory [see Dorf and Bishop, 2000] and reinforcement learning [see Sutton and Barto, 2018], where problems such as estimating the value function could benefit from leveraging latent linear features.

Exploration of non-linear feature learning. The empirical success of deep neural networks is largely attributed to their ability to learn complex non-linear features [Goodfellow et al., 2016, Chapter 15]. A logical next step in relation to the present thesis is extending our framework to encompass non-linear feature learning by, for instance, investigating neural networks with two hidden layers. This is a promising area of research, with introductory works such as Moniri et al. [2024]. This extension would allow to retain the benefits of the smaller-dimensional features, as in the multi-index model and methods presented in this thesis, while also enhancing the model's ability to capture more complex patterns.

Résumé des Contributions

Nous résumons ici en français les contributions présentées dans la thèse. Afin de comprendre les enjeux auxquels répondent ces contributions, nous conseillons fortement la lecture de l'introduction donnée dans le Chapitre 1 (en anglais), qui contient également les informations présentées ici. Nous proposons trois méthodes distinctes pour l'apprentissage non-paramétrique avec des caractéristiques linéaires cachées, en utilisant la minimisation du risque empirique régularisé. Avant d'aborder les contributions de chaque chapitre, nous fournissons un aperçu de la structure de la thèse. Chaque chapitre introduit une méthode différente, avec ses propres notations (dont la majorité est commune) et résultats, permettant une lecture indépendante. Les chapitres sont présentés dans l'ordre de leur développement au cours de cette thèse.

Dans le Chapitre 1, l'objectif est d'établir le cadre théorique et les motivations de cette thèse, en situant notre travail dans le contexte plus large de la recherche actuelle. Nous commençons par un examen des concepts clés de l'apprentissage supervisé et de la théorie de l'apprentissage, en mettant particulièrement l'accent sur les garanties de généralisation pour des échantillons finis, qui sont essentielles pour évaluer les performances des algorithmes proposés. Le chapitre explore ensuite les défis posés par les données de grande dimension, en se concentrant particulièrement sur l'utilisation des hypothèses de parcimonie pour gérer et réduire la complexité des problèmes d'apprentissage. Ensuite, nous nous intéressons au modèle multi-index, en fournissant une revue des méthodologies existantes dans ce domaine. Le chapitre se termine par un résumé et une analyse des trois principales contributions de cette thèse. Cette partie résumant les contributions est ici retranscrite en français.

Dans le Chapitre 2, nous explorons une approche novatrice en tant qu'extension du travail de Rosasco et al. [2013] sur la sélection de variables. La méthode incorpore une pénalité de norme nucléaire sur la matrice empirique des gradients dans des espaces de Hilbert à noyau reproduisant (RKHS) qui incluent les dérivées partielles de leurs noyaux. L'idée clé est d'exploiter les gradients de la fonction pour capturer la structure linéaire sous-jacente des données.

Dans le Chapitre 3, nous étendons le cadre de la minimisation du risque empirique en introduisant une pénalité de type LASSO groupé basée sur les dérivées et appliquée aux fonctions représentées dans une base de polynômes de Hermite orthonormaux multidimensionnels. En utilisant les propriétés d'orthogonalité et d'invariance par rotation des polynômes de Hermite, nous faisons pivoter les données de manière itérative, les alignant avec les directions les plus informatives.

Dans le Chapitre 4, nous introduisons une méthode novatrice utilisant des moyennes d'espaces de Sobolev sur des projections unidimensionnelles des données. La méthode combine des méthodes à noyaux et des réseaux de neurones avec une couche cachée de largeur infinie. Notre approche, centrée sur le noyau Brownien, remplace la non-linéarité des activations ReLU dans les réseaux de neurones par une fonction issue d'un RKHS. L'homogénéité positive du noyau Brownien est essentielle pour guider le processus d'optimisation.

Enfin, la conclusion aborde plusieurs questions de recherche importantes restées ouvertes. Nous passons maintenant à un examen plus détaillé de chaque contribution, en commençant par une discussion des méthodes, suivie des principaux résultats, et en terminant par une analyse des forces et faiblesses de chaque approche. Les références pertinentes pour les contributions complètes sont fournies dans les chapitres correspondants.

Pénalité de Norme Nucléaire sur la Matrice Empirique des Gradients

Ici, nous présentons le travail non encore publié du Chapitre 2, tandis que le code correspondant est disponible à `https://github.com/BertilleFollain/KTNGrad`.

Méthode. Dans le Chapitre 2, en nous appuyant sur les travaux de Rosasco et al. [2013] sur la sélection de variables, nous introduisons une méthode novatrice appelée KTNGrad. L'idée clé derrière KTNGrad est d'exploiter les informations sur l'espace des caractéristiques sous-jacent contenues dans les gradients tout en opérant dans des espaces de Hilbert à noyau reproduisant suffisamment réguliers. Cela nous permet d'éviter le calcul des gradients par différences finies et, à la place, de les calculer directement grâce aux propriétés intrinsèques des RKHS.

Nous commençons par noter que si le minimiseur du risque attendu (ici considéré pour la perte quadratique), f^*, satisfait le modèle multi-index, alors il existe une fonction g^* et une matrice $P \in \mathbb{R}^{d\times s}$ telles que $f^* = g^*(P^\top \cdot)$. En supposant que toutes les quantités pertinentes soient bien définies, cela implique que pour tout $x \in \mathcal{X}$, le gradient satisfait $\nabla f^*(x) = P\nabla g^*(P^\top x)$. Par conséquent, le gradient en un point donné contient des informations sur l'espace des caractéristiques sous-jacent. Plus formellement, pour toute fonction f appartenant à l'espace de Sobolev $H^1(\rho_X)$ (avec ρ_X la distribution des covariables), défini comme $H^1(\rho_X) := \{f \in L^2(\rho_X) \mid \forall a \in [d],\ \partial f(x)/\partial x^{(a)} \in L^2(\rho_X)\}$, nous pouvons exprimer la matrice de covariance des gradients de f comme suit

$$\mathrm{cov}(\nabla f) := \mathrm{cov}(\nabla f(X)) = \mathbb{E}_{\rho_X}\left(\nabla f(X)\nabla f(X)^T\right) \in \mathbb{R}^{d\times d}.$$

Dans ce cadre, la matrice de covariance des gradients de la véritable fonction f^* satisfait $\mathrm{cov}\left(\nabla f^*\right) = P\,\mathrm{cov}\left(\nabla g^*(P^T X)\right)P^T$. En supposant que le rang de $\mathrm{cov}\left(\nabla g^*(P^T X)\right)$ est égal à s, le nombre de caractéristiques linéaires, il en résulte que le rang de $\mathrm{cov}(\nabla f^*)$ est également s, qui est typiquement bien plus petit que d.

Cependant, comme le rang est à la fois non continu et non convexe, il présente des défis significatifs en tant que pénalité d'optimisation. En outre, le calcul direct de la matrice de covariance est irréalisable puisque ρ_X est inconnu. Pour surmonter ces problèmes et en suivant des extensions classiques de la régularisation par norme ℓ_1, nous utilisons une relaxation convexe en employant la norme nucléaire ($\|\cdot\|_*$) de la matrice d'échantillons des gradients

$$\nabla_n f := (\nabla f(x_1)^T, \nabla f(x_2)^T, \ldots, \nabla f(x_n)^T)^T/\sqrt{n} \in \mathbb{R}^{n\times d},$$

qui estime $\mathrm{tr}\left(\sqrt{\mathrm{cov}(\nabla f)}\right)$, fournissant une alternative convexe au rang de $\mathrm{cov}(\nabla f)$.

Il reste deux défis à relever : comment calculer les gradients aux points de données, étant donné que les différences finies sont souvent peu fiables et instables dans le contexte des covariables aléatoires, et comment calculer efficacement le minimiseur du problème de minimisation du risque empirique régularisé. C'est ici que les espaces de Hilbert à noyau reproduisant (RKHS) deviennent avantageux. Soit $\mathcal{H}$ un RKHS associé à un noyau reproduisant k. Si nous supposons que le noyau est deux fois différentiable, comme c'est le cas pour le noyau Gaussien, alors pour tout $a \in [d]$, $(\partial_a k)_x := t \to \partial k(x,t)/\partial x_a$ (la dérivée par rapport à la a-ième composante de x) appartient également à $\mathcal{H}$ pour tout $x \in \mathcal{X}$. De plus, pour toute fonction $f \in \mathcal{H}$, la dérivée partielle de f en x par rapport à x_a peut être calculée comme $\frac{\partial f(x)}{\partial x_a} = \langle f, (\partial_a k)_x \rangle_{\mathcal{H}}$. Cette propriété nous permet de calculer les gradients aux points de données directement en utilisant la structure du RKHS, sans avoir besoin de différences finies. En outre, la minimisation du risque empirique dans un RKHS est généralement abordable grâce au théorème de représentation, qui garantit que le minimiseur peut être exprimé comme une combinaison linéaire des fonctions noyau aux points de données.

Formellement, l'estimateur KTNGrad $\hat{f}$ est défini par le problème d'optimisation suivant

$$\hat{f}_\tau = \arg\min_{f \in \mathcal{H}} \widehat{\mathcal{R}}(f) + 2\tau \|\nabla_n f\|_* + \tau\nu \|f\|_{\mathcal{H}}^2,$$

où la perte définissant le risque est la perte quadratique, et $\tau > 0$ est un paramètre de régularisation à choisir. Une régularisation supplémentaire consistant en la norme du RKHS au carré est utilisée pour garantir la stabilité computationnelle et statistique, avec un paramètre fixe et très petit ν.

Résultat principal. Les principales propriétés statistiques de KTNGrad sont résumées dans le théorème informel suivant, qui décrit les hypothèses clés ainsi que les capacités de prédiction et d'apprentissage des caractéristiques linéaires.

Théorème 1 (Informel)**.** *Supposons que la véritable fonction de régression f^* appartient à $\mathcal{H}$, avec un noyau reproduisant deux fois différentiable.*

- ***Convergence du risque attendu :*** *Le risque attendu de KTNGrad converge vers le risque minimal $\mathcal{R}(f^*)$ sans dépendance exponentielle à la dimension des données d. Plus précisément, il existe une constante universelle $C > 0$ telle que pour tout $\delta \in (0,1]$, avec probabilité au moins $1-\delta$,*

$$\mathcal{R}(\hat{f}_\tau) - \mathcal{R}(f^*) \leq C\left(\frac{1}{\sqrt{n}}\left(\frac{1}{\sqrt{\tau\nu}} + 1\right)^2 + \sqrt{\frac{\tau}{\nu}}\frac{d^{5/4}}{n^{1/4}}\right)\log\frac{6+2d}{\delta} + \tau\left(2\|\nabla f^*\|_* + \nu\|f^*\|_{\mathcal{H}}^2\right).$$

- ***Récupération des caractéristiques linéaires cachées :*** *Lorsque la taille de l'échantillon augmente, la méthode est capable de récupérer l'espace des caractéristiques sous-jacent en norme de Frobenius lorsque la dimension du sous-espace des caractéristiques est connue et, sinon, en tant que filtre sûr. Pour toute suite positive $(\tau_n)_{n\in\mathbb{N}}$ telle que $\tau_n \to 0$ et $(\sqrt{n}\tau_n^2)^{-1} \to 0$ lorsque $n \to \infty$, avec Π_Q la matrice de projection associée à une matrice Q, nous avons*

$$\|\Pi_P - \Pi_{\hat{P}_s}\|_F^2 \xrightarrow{P} 0 \quad et \quad \|\Pi_P(I_d - \Pi_{\hat{P}})\|_F^2 \xrightarrow{P} 0.$$

Analyse. Nous résumons les principales contributions et les enseignements tirés du Chapitre 2.

- **Taux de convergence :** KTNGrad atteint des taux de convergence pour le risque attendu qui ne dépendent pas de manière exponentielle de la dimension des données, répondant à un défi majeur dans l'apprentissage non-paramétrique en grande dimension. Cependant, ce résultat repose sur l'hypothèse forte que le modèle est bien spécifié ($f^* \in \mathcal{H}$), et il existe encore une dépendance polynomiale à la dimension des données.

- **Récupération des caractéristiques :** KTNGrad démontre une forte capacité à récupérer l'espace des caractéristiques sous-jacent, comme le confirment les analyses statistiques et les résultats expérimentaux. Cependant, bien que la méthode identifie systématiquement les caractéristiques linéaires, elle ne parvient pas à récupérer sa dimension. Cela est visible à la fois dans les expériences et dans la théorie, car nous prouvons seulement que l'estimation de la dimension est asymptotiquement supérieure à la dimension réelle de l'espace des caractéristiques en raison de la semi-continuité inférieure du rang, conduisant à un filtre sûr. Toutefois, une méthode adaptative pourrait pallier ce problème.

- **Coût computationnel :** La méthode nécessite la résolution d'un problème d'optimisation convexe avec une pénalité de norme nucléaire sur la matrice empirique des gradients. Bien que cela assure de bonnes propriétés de convergence, cela entraîne un coût computationnel substantiel de $O(n^3 d^4)$. Même si l'emploi d'une approximation de Nyström pourrait aider à réduire une partie de la charge computationnelle, la dépendance aux dérivées de toutes les variables et le contexte de travail dans un espace de Hilbert à noyau reproduisant rendent la méthode intrinsèquement gourmande en ressources.

- **Espace fonctionnel inadéquat :** Une limitation conceptuelle importante de KTNGrad réside dans le fait que les RKHS associés aux noyaux usuels ne sont pas bien adaptés au modèle multi-index. Le problème central provient de l'incompatibilité entre les hypothèses $f^* \in \mathcal{H}$ et $f^* = g^*(P^\top \cdot)$, une critique qui s'applique également au cadre de sélection de variables discuté par Rosasco et al. [2013]. Par exemple, appartenir au RKHS correspondant au noyau gaussien impose que toutes les dérivées premières de f^* soient de carré intégrables par rapport à la mesure de Lebesgue sur $\mathbb{R}^d$. Cependant, dans le cas simple d'une seule variable pertinente $f^*(x) = g^*(x_1)$, cette condition devient $\int_{\mathbb{R}^d} \left((g^*)'(x_1)\right)^2 \mathrm{d}x_1 \dots \mathrm{d}x_d < \infty$, ce qui n'est possible que dans des cas très particuliers. Ce raisonnement nous conduit à explorer dans le chapitre suivant un espace de Hilbert de fonctions ayant une base orthonormée de polynômes de Hermite. La décomposition des fonctions dans cette base révèle que l'espace fonctionnel s'aligne bien avec le modèle multi-index et le cadre de sélection de variables, offrant une interprétation claire de la dépendance à quelques variables ou projections linéaires à travers les coefficients de la base.

Pénalité de LASSO Groupé sur une Décomposition en Polynômes de Hermite

Cette contribution correspond au contenu du Chapitre 3, qui a été accepté par le journal *Electronic Journal of Statistics*: Follain and Bach [2024b], tandis que le code est disponible à `https://github.com/BertilleFollain/RegFeaL`.

Méthode. La méthode proposée, RegFeaL, exploite l'orthogonalité et l'invariance par rotation des polynômes de Hermite normalisés pour effectuer une sélection de variables ou un apprentissage des caractéristiques linéaires. Tout d'abord, nous mettons en évidence les propriétés pertinentes des polynômes de Hermite. Les polynômes de Hermite normalisés unidimensionnels $(h_k(x))_{k\geq 0}$ forment une base orthonormale pour la mesure gaussienne standard sur $\mathbb{R}$. Les premiers polynômes sont donnés par $h_0(x) = 1, \quad h_1(x) = x$, $h_2(x) = \frac{1}{\sqrt{2}}(x^2-1), \quad h_3(x) = \frac{1}{\sqrt{6}}(x^3-3x)$. Ces polynômes sont étendus au cas multivarié en définissant, pour $\alpha \in \mathbb{N}^d$,

$$H_\alpha(x) = \prod_{a=1}^{d} h_{\alpha_a}(x_a).$$

Cette famille forme une base orthonormale pour l'espace de Hilbert des fonctions au carré intégrable avec la distribution q, $L^2(q)$, où $q(x) = e^{-\|x\|^2/2}/(2\pi)^{d/2}$ désigne la loi normale standard sur $\mathbb{R}^d$.

Dans ce contexte, si une fonction $f \in L^2(q)$ est exprimée comme $f = \sum_{\alpha\in\mathbb{N}^d} \hat{f}(\alpha)H_\alpha$, alors la fonction f ne dépend pas d'une variable x_a si et seulement si tous les coefficients $\hat{f}(\alpha)$ pour $\alpha \in \mathbb{N}^d$ tels que $\alpha_a > 0$ sont nuls.[5] Ce motif particulier de parcimonie dans les coefficients motive l'utilisation d'une pénalité de type LASSO groupé superposé (overlapping group LASSO). Par conséquent, la base des polynômes de Hermite est bien adaptée à la sélection de variables.

Pour ce faire, nous introduisons une pénalité induisant de la parcimonie, dépendant des hyperparamètres $r \in (0, +\infty)$ et $(c_k)_{k\in\mathbb{N}^*}$ (soit $c_k = \mathbb{1}_{k\leq M}$, soit $c_k = \rho^k$ avec $M \in \mathbb{N}^*$ et $\rho \in (0, 1)$)

$$\Omega_{\text{var}}(f) = \left(\sum_{a=1}^{d} \left(\sum_{\alpha\in(\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} \right)^{1/r},$$

où $|\alpha| = \sum_{a=1}^{d} \alpha_a$. Cette pénalité encourage la parcimonie dans la dépendance de f aux variables individuelles. La condition

$$\left(\sum_{\alpha\in(\mathbb{N}^d)^*} \alpha_a \frac{1}{c_{|\alpha|}} \hat{f}(\alpha)^2 \right)^{r/2} = 0 \iff \int_{\mathbb{R}^d} \left(\frac{\partial f}{\partial x_a} \right)^2 q = 0$$

met en évidence que la pénalité impose la nullité de la dérivée de f par rapport à x_a. Nous estimons f^* dans le cadre de la sélection de variables en résolvant le problème d'optimisation suivant

$$f_{\text{var}}^{\lambda,\mu} := \operatorname*{arg\,min}_{f\in\mathcal{F}} \ \widehat{\mathcal{R}}(f) + \lambda\Omega_0^2(f) + \mu\Omega_{\text{var}}^r(f),$$

où λ est un paramètre fixé pour la pénalité induisant la régularité Ω_0, μ est un hyperparamètre à choisir, et la perte définissant le risque est une perte convexe quelconque. Lorsque $r \geq 1$ et que la fonction de perte est convexe, la fonction objectif est fortement convexe, garantissant un minimiseur global unique. Pour $r < 1$, typiquement utilisé en pratique pour éviter les biais trop importants des méthodes de type LASSO, seul un minimiseur local peut être trouvé.

La propriété d'invariance par rotation des polynômes de Hermite est centrale pour étendre cette méthode à l'apprentissage des caractéristiques linéaires. Plus précisément,

[5] Ici, $\hat{f}(\alpha)$ correspond aux coefficients dans la décomposition en polynômes de Hermite de f, et non à l'estimateur de f^*.

pour tout $x, x' \in \mathbb{R}^d$, tout $k \in \mathbb{N}$, et toute matrice orthogonale R de taille $d \times d$, nous avons

$$\sum_{|\alpha|=k} H_\alpha(x) H_\alpha(x') = \sum_{|\alpha|=k} H_\alpha(Rx) H_\alpha(Rx').$$

Cette propriété permet le développement d'une pénalité adaptée à l'apprentissage des caractéristiques, définie comme suit

$$\Omega_{\text{feat}}(f) = \left(\text{tr}\left(M_f^{r/2}\right)\right)^{1/r},$$

où la matrice M_f est donnée par

$$(M_f)_{a,b} = \sum_{\alpha \in \mathbb{N}^d} \frac{1}{c_{|\alpha|+1}} \sqrt{\alpha_a + 1}\sqrt{\alpha_b + 1}\, \hat{f}(\alpha + e_a)\hat{f}(\alpha + e_b), \quad a, b \in [d].$$

Encore une fois, il existe un lien entre la nullité des dérivées et la définition de la pénalité, décrit dans le texte principal. La matrice M_f est semi-définie positive, et la pénalité $\Omega_{\text{feat}}(f)$ encourage la parcimonie en poussant les valeurs propres de M_f vers zéro, favorisant ainsi une solution de faible rang. Il est important de noter que $c_{|\alpha|}$ dépend uniquement de $|\alpha|$, ce qui garantit que la pénalité reste invariante par rotation, un aspect crucial pour éviter que la pénalité ne favorise des directions spécifiques.

La décomposition spectrale $M_f = UDU^\top$ révèle que si le rang de D est s, alors la fonction f dépend uniquement de s combinaisons linéaires des variables originelles, correspondant aux directions dans U avec des valeurs propres non nulles. En outre, on peut construire une fonction pivotée $g = f(U\cdot)$ telle que la pénalité des caractéristiques sur f soit équivalente à la pénalité de sélection de variables sur g. Cela montre que l'apprentissage des caractéristiques peut être vu comme une extension de la sélection de variables permettant des rotations dans l'espace des caractéristiques.

L'estimateur de f^* dans le cadre de l'apprentissage des caractéristiques est donc défini de manière similaire à celui de la sélection de variables, en remplaçant Ω_{var} par Ω_{feat}. Pour calculer l'estimateur, nous employons une formulation variationnelle pour $\Omega_{\text{feat}}(f)$ qui reformule le problème comme la minimisation sur deux variables : la fonction f et une variable auxiliaire Λ. Plus précisément, nous résolvons

$$f_{\text{feat}}^{\lambda,\mu},\ \Lambda_{\text{feat}}^{\lambda,\mu} = \underset{f \in \mathcal{F},\ \Lambda \in \mathbb{R}^{d\times d}}{\arg\min}\ \widehat{\mathcal{R}}(f) + \lambda \Omega_0^2(f) + \mu \text{trace}(\Lambda^{-1} M_f),$$

sous les contraintes que $\Lambda = R\text{Diag}(\eta)R^\top$ avec R une matrice orthogonale $d \times d$ et $\sum_{a=1}^d \eta_a^{r/(2-r)} = 1$, avec η un vecteur positif. Le processus d'optimisation alterne entre la mise à jour de Λ à f fixée par calcul de la décomposition spectrale de M_f, et la la mise à jour de f à Λ fixée en résolvant un problème de régression ridge à noyau, avec le noyau reproduisant k_Λ défini grâce aux polynômes de Hermite

$$k_\Lambda(x, x') = \sum_{\alpha \in (\mathbb{N}^d)^*} \frac{c_{|\alpha|} H_\alpha(R^\top x) H_\alpha(R^\top x')}{\lambda + \mu \alpha^\top \eta^{-1}}.$$

Cette procédure itérative peut être interprétée comme une rotation progressive des données pour découvrir et s'aligner avec les caractéristiques sous-jacentes tout en apprenant simultanément la fonction de prédiction.

Étant donné que le noyau est défini comme une somme infinie sur les polynômes de Hermite, un calcul direct est impossible. Nous approchons donc le noyau par échantillon-

nage. Nous utilisons une technique d'échantillonnage adaptative où les coefficients α sont tirés d'une distribution guidée par la variable auxiliaire η. Le processus d'optimisation converge rapidement en pratique, typiquement en quelques itérations. La complexité d'une itération est donnée par

$$O\Big(\underbrace{nm'd + nd^2}_{\text{Polynômes de Hermite}} + \underbrace{d^2(m')^2 + d^3}_{M_f \text{ et sa décomposition}} + \underbrace{md}_{\text{Échantillonnage}} + \underbrace{nm' \max(n, m')}_{\text{Ridge Kernel}}\Big),$$

où m est le nombre d'échantillons tirés pour α (et m' le nombre d'échantillons uniques obtenus). Cette complexité peut être substantielle, car m' doit être suffisamment grand pour garantir que la représentation du noyau soit précise.

Résultat principal. Le résultat statistique principal de ce chapitre est que l'estimateur RegFeaL, sous des hypothèses minimales, atteint la convergence du risque attendu vers le risque minimal avec une haute probabilité, malgré une sensibilité à la dimensionnalité des données. Nous illustrons cela par un théorème informel dans le cas où les covariables sont bornées et où la séquence de régularisation est choisie comme $c_k = \rho^k$.

Théorème 2 (Informel)**.** *Supposons que les covariables X soient bornées, c'est-à-dire que $\|X\|_2 \leq R$ presque sûrement, et que la fonction de perte ℓ soit lipschitzienne avec une constante G. Supposons que la véritable fonction de régression f^* existe et appartienne à $L^2(q)$. Le paramètre de régularisation λ est fixé à zéro, et μ est choisi en fonction de paramètres connus du problème. Nous définissons la norme $\Omega(f)$ comme $\Omega_{\text{feat}}(f) + |\hat{f}(0)|$ ou $\Omega_{\text{var}}(f) + |\hat{f}(0)|$. Alors, pour tout $\delta \in (0, 1)$, avec une probabilité d'au moins $1 - \delta$, le risque attendu $\mathcal{R}(f^\mu)$ de l'estimateur RegFeaL f^μ satisfait*

$$\mathcal{R}(f^\mu) \leq \mathcal{R}(f^*) + \Omega(f^*) \cdot \frac{G}{\sqrt{n}} \sqrt{1 + \frac{e^{R^2/2}}{(1-\rho)^d}} \left(16\sqrt{\frac{\pi}{2}} + 4\sqrt{2}\sqrt{\log\frac{2}{\delta}}\right),$$

et la norme de l'estimateur est bornée par $\Omega(f^\mu) \leq 2\Omega(f^*)$.

Le Chapitre 3 présente ce résultat dans un cadre plus général, où les données ne sont pas nécessairement bornées et où tout choix de la séquence d'hyperparamètres $(c_k)_{k>0}$ peut être considéré. Ce résultat informel met en évidence la sensibilité de la méthode à la dimensionnalité d, en particulier en raison de la dépendance exponentielle introduite par la base infinie des polynômes de Hermite, tout en maintenant une dépendance favorable à la taille de l'échantillon. L'hypothèse sur la fonction vraie f^* est modérée, étant donné la grande généralité de l'espace fonctionnel considéré.

Analyse. Nous résumons les principales contributions et les enseignements tirés du Chapitre 3.

- **Utilisation des polynômes de Hermite :** La méthode exploite l'orthogonalité et l'invariance par rotation des polynômes de Hermite pour aligner efficacement les données avec leurs directions des caractéristiques. La structure des polynômes de Hermite est particulièrement bien adaptée à la sélection de variables et à l'apprentissage des caractéristiques, car elle permet la définition d'une pénalité de type LASSO groupé superposé (overlapping group LASSO) qui décrit la dépendance à quelques variables ou projections linéaires. Par rapport à l'espace fonctionnel précédent du Chapitre 2, la décomposition en polynômes de Hermite est avantageuse car elle s'adapte naturellement aux fonctions qui dépendent d'un petit sous-ensemble de variables ou de caractéristiques linéaires.

- **Résultats statistiques :** La méthode offre des garanties statistiques sous des hypothèses minimales, notamment par rapport à l'espace fonctionnel auquel appartient la véritable fonction de prédiction. Cette large applicabilité est une force majeure. Cependant, la dépendance à une base infinie introduit une dépendance exponentielle à la dimensionnalité des données, ce qui pose des défis significatifs dans les contextes de grande dimension où les modèles multi-index sont les plus pertinents. Bien que des techniques de preuve alternatives puissent réduire cette dépendance, une telle approche n'est pas encore évidente.

- **Coût computationnel :** Le recours à une base infinie de polynômes de Hermite nécessite une méthode d'échantillonnage sophistiquée pour approximer le noyau à chaque itération du processus d'optimisation. Bien que cette approche soit théoriquement fondée, elle est coûteuse sur le plan computationnel et limite l'utilisation pratique de la méthode, en particulier dans les problèmes de grande dimension.

- **Limitations de l'espace fonctionnel :** L'espace de fonctions proposé, basé sur les polynômes de Hermite, est bien adapté pour capturer les motifs de parcimonie nécessaires à la sélection de variables et à l'apprentissage de caractéristiques linéaires. Comparé aux RKHS envisagés dans le chapitre précédent, cet espace est réellement compatible avec le modèle multi-index et la dépendance en un nombre limité de variables, tout en étant plus large, ce qui permet des hypothèses plus modérées sur la véritable fonction de régression f^*. Cependant, cette expansivité introduit également des défis significatifs. La dépendance exponentielle à la dimensionnalité dans les résultats statistiques, ainsi que la nécessité d'une méthode d'échantillonnage complexe dans la procédure d'optimisation, suggèrent que d'autres espaces fonctionnels pourraient être plus appropriés. Dans le chapitre suivant, nous considérons un autre espace fonctionnel basé sur la fusion de réseaux de neurones à une couche cachée de largeur infinie et des méthodes à noyaux. Bien que cet espace soit également intraitable sur le plan computationnel en raison de sa définition utilisant une intégrale, il peut être approximé plus efficacement à l'aide de particules, offrant une alternative plus simple à échantillonnage utilisé dans RegFeaL.

Intégration des Réseaux de Neurones et des Méthodes à Noyaux

Cette contribution correspond au contenu du Chapitre 4, disponible dans le preprint (en cours de révision pour le Journal of Machine Learning Research) : Follain and Bach [2024a], tandis que le code est disponible à `https://github.com/BertilleFollain/BKerNN`.

Méthode. Dans ce chapitre, nous introduisons une approche novatrice appelée Brownian kernel neural network (BKerNN), qui fusionne les réseaux de neurones et les méthodes à noyaux. L'idée clé derrière BKerNN est la construction d'un espace fonctionnel sur mesure inspiré par les réseaux de neurones à une seule couche cachée de largeur infinie, où la non-linéarité est remplacée par une fonction issue d'un espace de Hilbert à noyau reproduisant (RKHS). Cet espace fonctionnel, noté $\mathcal{F}_\infty$, permet à chaque fonction d'être représentée comme une intégrale sur des combinaisons linéaires des caractéristiques d'entrée, pondérées par une mesure de probabilité. Plus précisément, les fonctions de cet espace prennent la forme

$$f(x) = c + \int_{\mathcal{S}^{d-1}} g_w(w^\top x)\,\mathrm{d}\mu(w),$$

où c est une constante, w est un vecteur directionnel situé sur la sphère unité $\mathcal{S}^{d-1}$ pour une norme $\|\cdot\|$ donnée (soit ℓ_1, soit ℓ_2), g_w est une fonction qui varie avec w et appartient à un espace de Sobolev qui est également un RKHS $\mathcal{H}$, et μ est une mesure de probabilité sur la sphère. L'espace $\mathcal{H}$ contient des fonctions avec des dérivées faibles au carré intégrables, garantissant un certain degré de régularité, et telles que $g(0) = 0$. Le produit scalaire de $\mathcal{H}$ est défini par $\langle g, \tilde{g}\rangle = \int_{\mathbb{R}} g'\tilde{g}'$, et son noyau reproduisant est donné par $k^{(B)}(a,b) = (|a|+|b|-|a-b|)/2 = \min(|a|,|b|)\mathbb{1}_{ab>0}$, également connu sous le nom de noyau Brownien.

En pratique, nous approchons cet espace de dimension infinie par une version de largeur finie $\mathcal{F}_m$, où l'intégrale est remplacée par une somme finie sur m particules (similaires aux neurones dans un réseau de neurones). Ainsi, les fonctions dans $\mathcal{F}_m$ sont exprimées comme suit

$$f(x) = c + \frac{1}{m}\sum_{j=1}^{m} g_j(w_j^\top x),$$

où $(w_j)_{j\in[m]}$ sont les vecteurs directionnels, et $(g_j)_{j\in[m]}$ sont les fonctions correspondantes issues de l'espace $\mathcal{H}$. Le processus d'apprentissage dans BKerNN est guidé par un terme de régularisation qui contrôle la complexité de la fonction apprise. La régularisation de base est définie comme suit

$$\Omega_0(f) = \int_{\mathcal{S}^{d-1}} \|g_w\|_{\mathcal{H}} \,\mathrm{d}\mu(w),$$

où $\|g_w\|_{\mathcal{H}}$ mesure la rugosité de la fonction g_w. Cette régularisation induit de la parcimonie dans les représentations apprises en limitant le nombre de fonctions non nulles g_w, ce qui favorise indirectement la sélection de caractéristiques. Pour renforcer davantage la capacité d'apprentissage des caractéristiques de BKerNN, d'autres termes de régularisation peuvent être introduits. Par exemple, une pénalité sur les variables encourage le modèle à dépendre de seulement quelques variables en pénalisant les normes des lignes de la matrice de poids contenant les $(w_j)_{j\in[m]}$. Une pénalité pour les caractéristiques, quant à elle, favorise l'apprentissage d'une représentation de faible rang en appliquant une pénalité de norme nucléaire à la matrice des particules, encourageant la dépendance à seulement quelques transformations linéaires des données.

L'objectif d'optimisation est de minimiser le risque empirique régularisé

$$\hat{f}_\lambda = \operatorname*{arg\,min}_{f\in\mathcal{F}} \ \widehat{\mathcal{R}}(f) + \lambda\Omega_{\text{weights}}(f),$$

où λ est le paramètre de régularisation et Ω_{weights} est l'une des pénalités considérées.

BKerNN peut être vue sous deux perspectives différentes : comme une méthode à noyau et comme un réseau de neurones. Avec le point de vue des méthodes à noyaux, le processus d'apprentissage consiste en une régression ridge à noyau avec un noyau appris pendant l'entraînement. La matrice de noyau est définie comme suit

$$K = \frac{1}{m}\sum_{j=1}^{m} K^{(w_j)},$$

où $K^{(w_j)}$ est la matrice de noyau associée au noyau Brownien pour les données projetées dans la direction w_j.

Avec le point de vue des réseaux de neurones, BKerNN ressemble à un réseau de neurones à une seule couche cachée où les poids de la couche d'entrée à la couche cachée sont les vecteurs directionnels $(w_j)_{j\in[m]}$, et les fonctions d'activation sont les fonctions apprises $(g_j)_{j\in[m]}$. Contrairement aux réseaux de neurones traditionnels où les fonctions

d'activation sont prédéfinies, BKerNN apprend directement les fonctions d'activation, ajoutant ainsi de la flexibilité au modèle.

L'espace fonctionnel $\mathcal{F}_\infty$ est plus large que l'espace des fonctions représentables par les réseaux de neurones traditionnels à une seule couche cachée avec des activations ReLU. Cela est visible grâce à une analyse de transformée de Fourier, indiquant que BKerNN peut capturer une plus grande variété de fonctions. Cette puissance de représentation élargie ne se traduit pas par une complexité d'optimisation accrue.

Le calcul de BKerNN repose sur une version adaptée du théorème de représentation, qui fournit une formulation paramétrique pour la minimisation. Nous nous concentrons ici sur la perte quadratique pour simplifier l'exposé et permettre des solutions en forme close. Cependant, la méthode est généralisable à d'autres fonctions de perte en utilisant des techniques basées sur le gradient. Le problème d'optimisation peut être formulé comme suit

$$\min_{w_1,\dots,w_m\in\mathbb{R}^d, c\in\mathbb{R}, \alpha\in\mathbb{R}^n} \frac{1}{2n}\|Y - K\alpha - c\mathbb{1}_n\|_2^2 + \frac{\lambda}{2}\alpha^\top K\alpha + \lambda\Omega_{\text{weights}}(w_1,\dots,w_m),$$

où K est la matrice noyau définie par les poids $(w_j)_{j\in[m]}$ (qui ne sont plus contraints après reformulation) et α apparaît grâce au théorème de représentation. Le processus d'optimisation alterne entre deux étapes : optimiser les coefficients α et l'intercept c en gardant les poids $(w_j)_{j\in[m]}$ fixes, puis optimiser les poids $(w_j)_{j\in[m]}$ en utilisant une descente de gradient proximal.

Lorsque les poids $(w_j)_{j\in[m]}$ sont fixes, la matrice noyau K devient constante, permettant de résoudre explicitement l'optimisation de α et c, comme dans un problème classique de régression ridge à noyau. La complexité de cette étape est de $O(n^3 + n^2 d)$, ce qui peut être coûteux pour des ensembles de données volumineux. Pour réduire ce coût, des techniques telles que la méthode de Nyström peuvent être employées pour approximer la matrice de noyau.

La deuxième étape consiste à optimiser les poids $(w_j)_{j\in[m]}$ tout en gardant α et c fixes. Cela est plus difficile, car la fonction objectif résultante G n'est pas convexe par rapport aux poids et elle est seulement différentiable presque partout. Les poids sont donc mis à jour en utilisant une descente de gradient proximal.

L'opérateur proximal dépend de la pénalité. Par exemple, la mise à jour pour la pénalité de base Ω_0 est donnée par

$$w_j \leftarrow \text{prox}_{\lambda\gamma\Omega}\left(w_j - \gamma\frac{\partial G}{\partial w_j}\right) \quad \text{où } \text{prox}_{\lambda\gamma\Omega}(u) = \left(1 - \frac{\lambda\gamma}{2m}\frac{1}{\|u\|}\right)_+ u,$$

où γ est le pas, ajusté via une recherche linéaire avec retour en arrière pour garantir une optimisation efficace. Chaque étape proximale est facile à calculer à l'aide des formules explicites, avec des complexités allant de $O(md)$ pour les pénalités de base et de variable, à $O(md\min(m,d))$ pour les pénalités sur les caractéristiques.

Le processus d'optimisation tire parti de l'homogénéité positive du noyau Brownien, qui garantit le bon comportement de la dynamique d'optimisation . Les perspectives théoriques issues des réseaux de neurones dans la limite de champ moyen (mean-field) soutiennent la convergence, bien qu'une preuve formelle fasse défaut en raison de la non-différentiabilité du noyau Brownien. En pratique, la procédure s'avère robuste, avec des expériences confirmant son efficacité.

Résultat principal. Le théorème informel suivant fournit un aperçu des capacités de généralisation de BKerNN en proposant une borne de haute probabilité sur le risque attendu de l'estimateur.

Théorème 3 (Informel)**.** *Considérons l'estimateur BKerNN $\hat{f}_\lambda$ avec la pénalité de base Ω_0. Supposons que la perte soit convexe et lipschitzienne avec une constante L, que la véritable fonction de régression f^* appartienne à $\mathcal{F}_\infty$ et que $1 + \sqrt{\|X\|^*}$ soit sous-gaussienne avec un proxy de variance σ^2, où $\|\cdot\|^*$ est la norme duale de celle utilisée pour définir la sphère $\mathcal{S}^{d-1}$. Alors, avec λ choisi en fonction de paramètres connus du problème (indépendant de $\Omega_0(f^*)$), avec une probabilité d'au moins $1-\delta$, le risque attendu de $\hat{f}_\lambda$ est borné par*

$$\mathcal{R}(\hat{f}_\lambda) \leq \mathcal{R}(f^*) + \Omega_0(f^*) CL \left(\frac{1}{\sqrt{n}} + G_n + \frac{\sigma}{\sqrt{n}} \sqrt{\log \frac{1}{\delta}} \right),$$

où C est une constante universelle, et G_n désigne la complexité gaussienne de la classe de fonctions avec Ω_0 contrainte sous un certain seuil. La quantité G_n est en outre bornée, en utilisant une autre constante universelle C', comme suit

$$G_n \leq C' \min \left(\sqrt{\frac{d}{n}} \sqrt{\log(n)} \sqrt{\mathbb{E}_X \|X\|^*},\ \frac{1}{n^{1/6}} (\log d)^{1/4} \left(\mathbb{E}_{X_1 \dots X_n} \big(\max_{i \in [n]} \|X_i\|^* \big)^2 \right)^{1/4} \right).$$

La borne sur G_n est présentée sous deux formes : une borne dépendant de la dimension et une borne indépendante de la dimension. La borne dépendant de la dimension évolue bien avec la taille de l'échantillon et montre seulement une dépendance quadratique à la racine de la dimension. En revanche, la borne indépendante de la dimension, bien que moins favorable en termes de taille d'échantillon ($n^{-1/6}$), dépend uniquement de la dimension de manière logarithmique. Les termes dépendant de la distribution des données ont également une dépendance raisonnable à la dimension.

Analyse. Nous résumons les contributions clés et les enseignements tirés du Chapitre 4.

- **Fusion des forces :** BKerNN fusionne efficacement certains des avantages des méthodes à noyaux et des réseaux de neurones à une couche cachée de largeur infinie. Pour ce faire, la fonction d'activation non linéaire traditionnelle est remplacée par une fonction tirée d'un espace de Hilbert à noyau reproduisant, ce qui améliore le pouvoir d'expression du modèle.
- **Optimisation efficace :** Le processus d'optimisation est simple et robuste par rapport aux réseaux de neurones classiques, bénéficiant de l'homogénéité positive du noyau Brownien. Cette propriété permet d'utiliser les idées tirées de l'analyse des réseaux de neurones dans la limite de champ moyen, ce qui rend le processus théoriquement fondé. L'approximation de l'espace de largeur infinie à l'aide de particules est facile comparée au processus d'échantillonnage utilisé pour RegFeaL.
- **Garanties de généralisation :** L'analyse statistique fournit des bornes de haute probabilité sur le risque attendu, montrant que BKerNN atteint des taux de convergence compétitifs. Deux types de bornes sont fournis : une borne dépendant de la dimension qui s'adapte bien à la taille de l'échantillon et une borne indépendante de la dimension qui s'adapte moins favorablement à la taille de l'échantillon mais qui dépend seulement de manière logarithmique de la dimension des données. Les hypothèses légères sur la distribution des données et la spécification du modèle rendent ces résultats largement applicables.

- **Performance pratique :** Les expériences numériques confirment les résultats théoriques, BKerNN surpassant souvent les méthodes classiques à noyaux et rivalisant favorablement avec les réseaux de neurones sur des ensembles de données réels.

- **Adaptabilité dans les modèles mal spécifiés :** Si le modèle est mal spécifié mais que le prédicteur de Bayes f^* est lipschitzien, les réseaux de neurones avec des activations ReLU et une norme de Banach bornée atteignent un taux de convergence de $O(n^{-1/(d+5)})$, tandis que les méthodes à noyaux atteignent $O(n^{-1/(d+1)})$, toutes deux limitées par la malédiction de la dimension. Dans les modèles mal spécifiés sous le modèle multi-index où $f^* = g^*(P^\top \cdot)$, les méthodes basées sur les RKHS ne parviennent pas à exploiter la réduction de dimensionnalité, ce qui entraîne des taux inchangés. Cependant, les réseaux de neurones peuvent s'adapter à cette structure, produisant des taux qui dépendent de la dimension inférieure de P plutôt que de d. BKerNN partage cette adaptabilité, comme l'indique le fait que $\Omega_0(f^*) \leq \Omega_0(g^*)$, ce qui, avec ses garanties théoriques solides dans les modèles bien spécifiés et ses excellentes performances pratiques, souligne son importance dans le domaine de l'apprentissage supervisé non-paramétrique avec des caractéristiques linéaires cachées.

Cela conclut le résumé des contributions proposé en français. Le Chapitre 1 (correspondant à l'introduction) décrit le contexte et les enjeux dans lesquels s'inscrivent cette thèse. Chaque contribution est présentée en détail dans les Chapitres 2, 3 et 4. La Conclusion présente des perspectives liées à ce travail.

Bibliography

R. Adams and J. Fournier. *Sobolev Spaces*. Elsevier Science, 2003.

A. Antoniadis, S. Lambert-Lacroix, and F. Leblanc. Effective dimension reduction methods for tumor classification using gene expression data. *Bioinformatics*, 19(5):563–570, 2003.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3): 243–272, 2008.

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4: 40–79, 2010.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3): 337–404, 1950.

D. Babichev and F. Bach. Slice inverse regression with score functions. *Electronic Journal of Statistics*, 12(1):1507–1543, 2018.

F. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9(35):1019–1048, 2008.

F. Bach. *Learning Theory from First Principles*. MIT Press, 2024. URL https://www.di.ens.fr/~fbach/ltfp_book.pdf. to appear.

F. Bach, G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *International Conference on Machine Learning (ICML)*, 2004.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1): 185–209, 2015.

A. Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2017.

R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, 2011.

A. Bietti, J. Bruna, C. Sanford, and M. J. Song. Learning single-index models with shallow neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 9768–9783, 2022.

A. Bietti, J. Bruna, and L. Pillaud-Vivien. On learning Gaussian multi-index models with gradient flow, 2023. URL https://arxiv.org/abs/2310.19793.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: PS*, 9:323–375, 2005.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

C. Boyer, A. Chambolle, Y. De Castro, V. Duval, F. de Gournay, and P. Weiss. On representer theorems and convex regularization. *SIAM Journal on Optimization*, 29(2):1260–1281, 2019.

D. Brillinger. A generalized linear model with "Gaussian" regressor variables. In *Selected Works of David Brillinger*, pages 589–606. Springer, 2012.

V. Cabannes, L. Pillaud-Vivien, F. Bach, and A. Rudi. Overcoming the curse of dimensionality with Laplacian regularization in semi-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 30439–30451, 2021.

E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2007.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.

L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

L. Chizat and F. Bach. Gradient descent on infinitely wide neural networks: global convergence and generalization. In *Proceedings of the International Congress of Mathematicians 2022*, pages 5398–5419, 2022.

R. D. Cook and S. Weisberg. Sliced inverse regression for dimension reduction: comment. *Journal of the American Statistical Association*, 86(414):328–332, 1991.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

A. Dalalyan, A. Juditsky, and V. Spokoiny. A new algorithm for estimating the effective dimension-reduction subspace. *Journal of Machine Learning Research*, 9(53):1647–1678, 2008.

I. Daubechies, R. DeVore, M. Fornasier, and C. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63:1–38, 01 2010.

A. Dontchev and T. Zolezzi. *Well-Posed Optimization Problems*. Lecture Notes in Mathematics. Springer, 2006.

R. Dorf and R. Bishop. *Modern Control Systems Global Edition*. Prentice-Hall, Inc., 2000.

P. Drineas and M. Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(72):2153–2175, 2005.

I. Ekeland and R. Témam. *Convex Analysis and Variational Problems*. Society for Industrial and Applied Mathematics, 1999.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

B. Follain and F. Bach. Enhanced feature learning via regularisation: integrating neural networks and kernel methods, 2024a. URL https://arxiv.org/abs/2407.17280.

B. Follain and F. Bach. Nonparametric linear feature learning in regression through regularisation. *Electronic Journal of Statistics*, 18(2):4075–4118, 2024b.

B. Follain, T. Wang, and R. J. Samworth. High-dimensional changepoint estimation with heterogeneous missingness. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 84(3):1023–1055, 2022.

J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

K. Fukumizu, F. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99, 2004.

K. Fukumizu, F. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

C. Geoffrey, L. Guillaume, and L. Matthieu. Robust high dimensional learning for Lipschitz and convex losses. *Journal of Machine Learning Research*, 21(233):1–47, 2020.

C. Giraud. *Introduction to High-Dimensional Statistics*. CRC Press, 2014.

M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(64):2211–2268, 2011.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 507–520, 2022.

M. Gruber. *Improving Efficiency by Shrinkage: The James-Stein and Ridge Regression Estimators*. Routledge, 1998.

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. Springer, 2002.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

C. Hermite. Sur un nouveau développement en série des fonctions. In *Œuvres de Charles Hermite*, volume 2, page 293–308. Cambridge University Press, 2009.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

M. Hristache, A. Juditsky, and V. Spokoiny. Direct estimation of the index coefficient in a single-index model. *The Annals of Statistics*, 29(2):593–623, 2001.

W. Härdle and T. M. Stoker. Investigating smooth multiple regression by the method of average derivatives. *Journal of the American Statistical Association*, 84(408):986–995, 1989.

R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. In *International Conference on Artificial Intelligence and Statistics (ICML)*, pages 366–373, 2010.

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

Q. M. Jing Zeng and X. Zhang. Subspace estimation with automatic dimension and variable selection in sufficient dimension reduction. *Journal of the American Statistical Association*, 119(545):343–355, 2024.

V. Koltchinskii, K. Lounici, and A. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

A. Krogh and J. Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 4, 1991.

V. Kurkova and M. Sanguineti. Bounds on rates of variable-basis and neural-network approximation. *IEEE Transactions on Information Theory*, 47(6):2659–2665, 2001.

G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5(Jan):27–72, 2004.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. A Series of Modern Surveys in Mathematics Series. Springer, 1991.

K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

K.-C. Li. On principal Hessian directions for data visualization and dimension reduction: another application of Stein's lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992.

F. Liu, L. Dadi, and V. Cevher. Learning with norm constrained, over-parameterized, two-layer neural networks. *Journal of Machine Learning Research*, 25(138):1–42, 2024.

U. Luxburg and O. Bousquet. Distance-based classification with Lipschitz functions. *Journal of Machine Learning Research*, 5(Jun):669–695, 2004.

P. Marion and R. Berthier. Leveraging the two-timescale regime to demonstrate convergence of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 64996–65029, 2023.

S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layer neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory (COLT)*, volume 99, pages 2388–2464, 2019.

R. Meir and T. Zhang. Generalization error bounds for Bayesian mixture algorithms. *Journal of Machine Learning Research*, 4(Oct):839–860, 2003.

Y. Mishura and G. Shevchenko. *Theory and Statistical Applications of Stochastic Processes*. John Wiley & Sons, Ltd, 2017.

B. Moniri, D. Lee, H. Hassani, and E. Dobriban. A theory of non-linear feature learning with one gradient step in two-layer neural networks. In *International Conference on Machine Learning (ICML)*, volume 235, pages 36106–36159, 2024.

A. Mousavi-Hosseini, D. Wu, and M. A. Erdogdu. Learning multi-index models with neural networks via mean-field langevin dynamics, 2024. URL https://arxiv.org/abs/2408.07254.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2): 227–234, 1995.

A. Nitanda and T. Suzuki. Stochastic particle gradient descent for infinite ensembles, 2017. URL https://arxiv.org/abs/1712.05438.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12(85): 2825–2830, 2011.

I. Pinelis and A. Sakhanenko. Remarks on inequalities for large deviation probabilities. *Theory of Probability & Its Applications*, 30(1):143–148, 1986.

R. T. Powers and E. Størmer. Free states of the canonical anticommutation relations. *Communications in Mathematical Physics*, 16(1):1–33, 1970.

Z. Z. Qian Lin and J. S. Liu. Sparse sliced inverse regression via Lasso. *Journal of the American Statistical Association*, 114(528):1726–1739, 2019.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20, 2007.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

M. Reid. Generalization bounds. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, pages 447–454. Springer, 2010.

L. Rosasco, S. Villa, S. Mosci, M. Santoro, and A. Verri. Nonparametric sparsity and regularization. *Journal of Machine Learning Research*, 14(52):1665–1714, 2013.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Computational Learning Theory (COLT)*, pages 416–426, 2001.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

J. Sirignano and K. Spiliopoulos. Mean field analysis of neural networks: a law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

B. Sriperumbudur, K. Fukumizu, and G. Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(70):2389–2410, 2011.

N. C. Stenseth, Y. Tao, C. Zhang, B. Bramanti, U. Büntgen, X. Cong, Y. Cui, H. Zhou, L. A. Dawson, S. J. Mooney, D. Li, H. G. Fell, S. Cohn, F. Sebbane, P. Slavin, W. Liang, H. Tong, R. Yang, and L. Xu. No evidence for persistent natural plague reservoirs in historical and modern Europe. *Proceedings of the National Academy of Sciences of the United States of America*, 119(51), 2022.

G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

T. M. Stoker. Consistent estimation of scaled coefficient. *Econometrica*, 54(6):1461–1481, 1986.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.

G. Szegő. *Orthogonal Polynomials*. American Mathematical Society Colloquium Publications. American Mathematical Society, 1939.

E. Süli and D. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 58(1):267–288, 1996.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused Lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(1):91–108, 2005.

V. Vapnik. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 4, 1991.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2013.

R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

V. Vovk. Kernel ridge regression. In B. Schölkopf, Z. Luo, and V. Vovk, editors, *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 105–116. Springer, 2013.

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

M. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

C. Williams and C. E. Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

S. Wright. Correlation and causation. *Journal of Agricultural Research*, 20(3):557–585, 1921.

Y. Xia. A multiple-index model and dimension reduction. *Journal of the American Statistical Association*, 103(484):1631–1640, 2008.

Y. Xia, H. Tong, W. K. Li, and L.-X. Zhu. An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(3):363–410, 2002.

Z. Yang, K. Balasubramanian, Z. Wang, and H. Liu. Estimating high-dimensional non-Gaussian multiple index models via Stein's lemma. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Y. Ying and C. Campbell. Rademacher chaos complexities for learning the kernel problem. *Neural Computation*, 22(11):2858–2886, 11 2010.

M. Yuan. On the identifiability of additive index models. *Statistica Sinica*, 21(4):1901–1911, 2011.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 68(1):49–67, 2006.

C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

D.-X. Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of Computational and Applied Mathematics*, 220(1):456–463, 2008.

H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101 (476):1418–1429, 2006.

RÉSUMÉ

Nous considérons le problème de l'apprentissage supervisé lorsqu'il existe des structures de données cachées, en nous concentrant sur les cas où quelques caractéristiques linéaires pertinentes expliquent la relation entre la réponse et les covariables, comme dans le modèle "multi-index". Notre objectif est de développer des méthodes qui exploitent ces structures cachées pour améliorer l'apprentissage. De nombreuses approches existantes reposent sur des hypothèses fortes concernant la génération de données et se heurtent à la malédiction de la dimensionnalité, présentant souvent une dépendance exponentielle en la dimension des données. Nous explorons les modèles "multi-index" en utilisant la minimisation du risque empirique régularisé (ERM), car ce cadre flexible est applicable à tout problème pour lequel un risque peut être défini. Tout au long de cette thèse, nous explorons trois méthodes innovantes pour simultanément apprendre les caractéristiques et estimer la fonction de prédiction dans un contexte non-paramétrique. Chaque méthode intègre des éléments des espaces de Hilbert à noyau reproduisant (RKHS), contient des pénalités d'apprentissage des caractéristiques qui sont adaptables à la sélection de variables, utilise des procédures d'optimisation basées sur la repondération pour un calcul efficace et s'appuie sur des hypothèses limitées sur le mécanisme de génération des données. Nous avons veillé à la facilité d'utilisation du code développé et à la reproductibilité des expériences. La première méthode, KTNGrad, considère l'ERM dans un RKHS avec une pénalité de norme nucléaire sur la matrice empirique des gradients. L'analyse théorique montre que KTNGrad a des taux de convergence pour le risque attendu dans les contextes bien spécifiés qui ne dépendent pas exponentiellement de la dimension, tout en estimant l'espace des caractéristiques pertinentes d'une manière sûre. La deuxième méthode, RegFeaL, exploite les propriétés d'orthogonalité et d'invariance par rotation des polynômes de Hermite. Cette méthode fait pivoter les données de manière itérative pour les aligner avec les caractériques. Le risque attendu converge vers le risque minimal avec des taux explicites, sans hypothèses fortes sur la véritable fonction de régression. Enfin, la troisième méthode, BKerNN, introduit un nouveau modèle qui combine les méthodes à noyaux et les réseaux de neurones. Cette méthode optimise les poids de la première couche par descente de gradient tout en ajustant explicitement la non-linéarité et les poids de la deuxième couche. L'optimisation tire parti de l'homogénéité positive du noyau Brownien, et l'analyse de la complexité de Rademacher montre que le risque attendu de BKerNN atteint des taux de convergence favorables qui sont indépendants de la dimension, sans hypothèses fortes sur la véritable fonction de régression ou sur les données.

MOTS CLÉS

apprentissage de caractéristiques, noyau reproduisant, minimisation du risque empirique régularisé, parcimonie, apprentissage supervisé, réseaux de neurones

ABSTRACT

We tackle the challenge of supervised learning with hidden data structures, focusing on cases where a few relevant linear features explain the relationship between response and covariates, as in the multi-index model. We aim to develop methods that leverage these hidden structures to improve learning. Many existing approaches rely on strong assumptions about data generation and struggle with the curse of dimensionality, often exhibiting exponential dependency on data dimension. We explore multi-index models through regularised empirical risk minimisation (ERM), as this flexible framework is applicable to any problem where a risk can be defined. Throughout this thesis, we explore three innovative methods for joint feature learning and function estimation in nonparametric learning. Each method integrates elements from reproducing kernel Hilbert spaces (RKHS), contains sparsity-inducing penalties for feature learning which are adaptable to the variable selection setting, uses optimisation procedures based on reweighting for efficient computation and relies on limited assumptions on the data-generating mechanism. We ensured the usability of the developed code and the reproducibility of the experiments. The first method, KTNGrad, considers ERM within an RKHS, augmented by a trace norm penalty on the sample matrix of gradients. Theoretical analysis shows that KTNGrad achieves convergence rates that do not depend exponentially on the dimension for the expected risk in well-specified settings while recovering the underlying feature space in a safe-filter manner. The second method, RegFeaL, leverages Hermite polynomials' orthogonality and rotation invariance properties. This method iteratively rotates the data to align with leading directions. The expected risk converges to the minimal risk with explicit rates without strong assumptions on the true regression function. Finally, the third method, BKerNN, introduces a novel framework that combines kernel methods and neural networks. This method optimises the first layer's weights via gradient descent while explicitly adjusting the non-linearity and weights of the second layer. The optimisation leverages the positive homogeneity of the Brownian kernel, and Rademacher complexity analysis shows that BKerNN achieves favourable convergence rates that are dimension-independent without strong assumptions on the true regression function or the data.

KEYWORDS

feature learning, reproducing kernel, regularised empirical risk minimisation, sparsity, supervised learning, neural networks