Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Resource Efficient Large-Scale Machine Learning

by

Luigi Carratino

Theses Series **DIBRIS-TH-2020-XXXII**

DIBRIS, Università di Genova
Via Opera Pia, 13 16145 Genova, Italy `http://www.dibris.unige.it/`

Università degli Studi di Genova

Dipartimento di Informatica, Bioingegneria,

Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in Computer Science and Systems Engineering
Computer Science Curriculum

Resource Efficient Large-Scale Machine Learning

by

Luigi Carratino

March, 2020

Dottorato di Ricerca in Informatica ed Ingegneria dei Sistemi
Indirizzo Informatica
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università degli Studi di Genova

DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy
http://www.dibris.unige.it/

Ph.D. Thesis in Computer Science and Systems Engineering
Computer Science Curriculum
(S.S.D. INF/01)

Submitted by Luigi Carratino
DIBRIS, Univ. di Genova

Date of submission: March 2020

Title: Resource Efficient Large-Scale Machine Learning

Advisors:
Lorenzo Rosasco
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova

Alessandro Rudi
INRIA - Département d'Informatique - École Normale Supérieure
PSL Research University

Supervisor:
Lorenzo Rosasco
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova

Ext. Reviewers:
Aymeric Dieuleveut
Center of Applied Mathematics - École Polytechnique
Institut Polytechnique de Paris

Zoltán Szabó
Center of Applied Mathematics - CNRS, École Polytechnique
Institut Polytechnique de Paris

Abstract

Non-parametric models provide a principled way to learn non-linear functions. In particular, kernel methods are accurate prediction tools that rely on solid theoretical foundations. Although they enjoy optimal statistical properties, they have limited applicability in real-world large-scale scenarios because of their stringent computational requirements in terms of time and memory. Indeed their computational costs scale at least quadratically with the number of points of the dataset and many of the modern machine learning challenges requires training on datasets of millions if not billions of points. In this thesis, we focus on scaling kernel methods, developing novel algorithmic solutions that incorporate budgeted computations. To derive these algorithms we mix ideas from statistics, optimization, and randomized linear algebra. We study the statistical and computational trade-offs for various non-parametric models, the key component to derive numerical solutions with resources tailored to the statistical accuracy allowed by the data. In particular, we study the estimator defined by stochastic gradients and random features, showing how all the free parameters provably govern both the statistical properties and the computational complexity of the algorithm. We then see how to blend the Nyström approximation and preconditioned conjugate gradient to derive a provably statistically optimal solver that can easily scale on datasets of millions of points on a single machine. We also derive a provably accurate leverage score sampling algorithm that can further improve the latter solver. Finally, we see how the Nyström approximation with leverage scores can be used to scale Gaussian processes in a bandit optimization setting deriving a provably accurate algorithm. The theoretical analysis and the new algorithms presented in this work represent a step towards building a new generation of efficient non-parametric algorithms with minimal time and memory footprints.

Table of Contents

Chapter 1

Introduction

In this thesis, we study how to derive new learning algorithms that require minimal memory and time footprints. In particular, we study kernel methods as a principled and statistically sound way to perform non-linear learning.

First, we see how in the statistical learning setting ideas from statistics, optimization and randomized linear algebra can be blended to derive numerical solutions with optimal statistical properties and low computational complexity. Secondly, we discuss in a bandit optimization setting how similar ideas can allow to scale non-parametric algorithms, deriving approximations with no loss of accuracy and preserving no-regret guarantees.

1.1 On the Need for Efficient Machine Learning

We live in a world awash with data. The speed at which we collect them is impressively high and the type of data is of the most various. Social networks daily collect millions of messages and pictures from people all around the world, while the particle accelerator at CERN generates 1 Gigabyte of raw data per second. Thanks to this abundance of data the potential applications of machine learning seem endless. But an underestimated aspect is the number of resources that are needed to run state of the art models on these immense datasets. An example is AlphaGo, the model developed by Google DeepMind that defeated the human world champion of Go, used a distributed system of resources for a total of 1,202 CPUs and 176 GPUs [SHM+16]. This scale of resources is accessible only too few and soon we may reach a point where adding more hardware and engineering effort will not be enough. The computational difficulty is often caused by the size of the dataset. While in practice people often use only a subset of the data they could work with, this comes at a cost in terms of accuracy of the learned model. In this thesis, we try to address the problem of designing efficient algorithms that can scale on large datasets and that

can be accurate both empirically and theoretically.

1.2 Can We Scale Non-parametric Methods ?

We discuss in the following the computational difficulties of non-parametric methods first in the statistical learning setting and then in the bandit optimization setting.

1.2.1 Statistical Learning Setting

The goal in supervised learning is to learn from examples a function that predicts well new data. In statistical machine learning, data are assumed to be samples from a fixed and unknown distribution [Vap99]. In most real-world scenarios, this distribution can be complex and so is often crucial to learn from the provided data a function that can potentially be highly non-linear.

Non-parametric learning methods provide a way to learn non-linear functions. In particular, we focus on Kernel methods [SS02, HSS08]. Kernel methods have proved to be a reliable approach to learn estimators with high predictive accuracy. Further, they enjoy excellent theoretical properties, making them one of the most popular learning techniques for almost thirty years.

Unfortunately, they have limited applications in settings where the number of training data is huge. Indeed time and memory requirements of these methods scale with the number of samples n that compose the training set. In particular, in its basic form, solving a kernel method takes a cost in time and memory respectively of $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$. When the order of magnitude of training points exceeds 10^5, this computational complexity makes the problem intractable.

Overcoming these limitations has motivated a variety of practical approaches. The learning problem can be reduced to the minimization of an empirical objective function that depends on the training data. This minimization problem is a starting point of some computational studies. In particular, optimization methods including gradient methods [YRC07], as well as accelerated [BPR07], stochastic [DFB17, RV15, Ora14, DXH$^+$14, LR17a, TRVR16] and preconditioned extensions [ACW16, GOSS16, MB17], have been used to improve time complexity, solving faster the minimization problem. Random projections instead provide an approach to reduce memory requirements through an approximation of the model. They allow deriving a different approximated minimization problem that can be solved with fewer memory constraints. Popular methods include Nyström [WS01, SS00], random features [RR08], and their numerous extensions [FSC$^+$16, HXGD14, KMT09]. But how much these techniques can be exploited without loosing accuracy is not obvious. Recent results show that there is a large class of problems for which, by combining the random features or the Nyström approach with ridge regression, it is possible to substantially reduce computations, while preserving the same optimal statistical accuracy of the non-approximated model [Bac13, AM15a, RCR15, RR17, Bac17]. While statistical

lower bounds exist for this setting, there are no corresponding computational lower bounds. So a first key question is to characterize statistical and computational trade-offs. Understanding if, or under which conditions, computational gains come at the expense of statistical accuracy would be fundamental to derive new and more efficient learning algorithms.

Another way to improve the scalability of kernel methods is to improve the approximation techniques themself. As we just discussed random projections are broadly used to reduce the computational burden. In particular, the Nyström method can be seen as a way to approximate a large matrix that defines the minimization problem. The Nyström method consists in replacing the large matrix with a smaller one made out of a subset of columns chosen uniformly at random from the original one. This approach is fast to compute, but the number of columns needed for a prescribed approximation accuracy does not take advantage of the possible low-rank structure of the matrix at hand. As discussed in [AM15a], leverage score sampling provides a way to tackle this shortcoming. Here columns are sampled proportionally to suitable weights, called leverage scores (LS) [DMIMW12, AM15a]. With this sampling strategy, the number of columns needed for a prescribed accuracy is governed by the so-called *effective dimension* which is a natural extension of the notion of rank. Despite these nice properties, performing leverage score sampling provides a challenge in its own right, since it has complexity in the same order of an eigendecomposition of the original matrix. In our specific case of learning with kernels, the original matrix would be of size $n \times n$ resulting in time complexity of roughly $\mathcal{O}(n^3)$. This computational burden dims any speedup that would follow the refined matrix approximation, making leverage score sampling useless. Much effort has been recently devoted to deriving fast and provably accurate algorithms for approximate leverage score sampling [Woo14, CLV17a, AM15a, MM17, CLV17c], but these results have poor approximation guarantees and/or do not match the complexity of state of the art approximated kernel methods without exploiting distributed resources. So a second question is how to lower the computational complexity of leverage score sampling resulting in an overall trim of the computational cost of the learning process.

1.2.2 Bandit Optimization Setting

Non-parametric methods are also popular in sequential decision making, in particular in the optimization under bandit feedback setting [LS19].

In this setting, a learning algorithm sequentially interacts with a reward function f. Over T interactions, the algorithm chooses a point x_t and it has only access to a noisy evaluation of f at x_t. The goal of the algorithm is to minimize the cumulative regret, which compares the reward accumulated at the points selected over time, $\sum_t f(x_t)$, to the reward obtained by repeatedly selecting the optimum of the function, i.e. $T \max_x f(x)$.

A popular and theoretically sound method is the GP-UCB algorithm first introduced by [SKKS10].

Starting from a Gaussian process (GP) prior over f [RW06], GP-UCB alternates between evaluating the function, and using the evaluations to build a posterior of f. This posterior is composed by a mean function μ that estimates the value of f, and a variance function σ that captures the uncertainty μ. These two quantities are combined in a single upper confidence bound (UCB) that drives the selection of the evaluation points and trades off between evaluating high-reward points (*exploitation*) and testing possibly sub-optimal points to reduce the uncertainty on the function (*exploration*).

The performance of GP-UCB has been studied by [SKKS10, VKM$^+$13, CG17] to show that GP-UCB provably achieves low regret both in a Bayesian and non-Bayesian setting. However, the main limiting factor to its applicability is its computational cost. Assuming the input space to have finite cardinality A, GP-UCB requires $\Omega(At^2)$ time/space to select each new point, resulting in a total time cost of $\mathcal{O}(AT^3)$ over T iterations. This computational complexity does not allow the algorithm to scale over complex optimization problems with many iterations.

Several approximations of GP-UCB have been suggested [QCRW07, LOSC18]. A first approach is to approximate the GP using the equivalent of the Nyström approximation in the Bayesian setting: *inducing points* [QCR05]. With this method, the GP can be restricted to lie in the range of a small subset of inducing points. The subset should cover the space well for accuracy, but also be as small as possible for efficiency. Methods referred to as *sparse GPs*, have been proposed to select the inducing points and an approximation based on the subset. Popular instances of this approach are the subset of regressors (SoR, [Wah90]) and the deterministic training conditional (DTC, [SWL03]). While these methods are simple to interpret and efficient, they do not come with regret guarantees. Moreover, when the subset does not cover the space well, they suffer from *variance starvation* [WGKJ18], as they underestimate the variance of points far away from the inducing points. A second approach is to use some variation of random features. Among these methods, [MK18] recently showed that discretizing the posterior on a fine grid of quadrature Fourier features (QFF) incurs a negligible approximation error. This is sufficient to prove that the maximum of the approximate posterior can be efficiently found and that it is accurate enough to guarantee that Thompson sampling with QFF provably achieves low regret. However this approach does not extend to non-stationary (or non-translation invariant) kernels and although its dependence on t is small, the approximation and posterior maximization procedure scale exponentially with the input dimension. Lastly, a more recent approach replaces the true GP likelihood with a variational approximation that can be optimized efficiently [HCKB19]. Although this method provides guarantees on the approximate posterior mean and variance, these guarantees only apply to GP regression and not to the harder optimization setting.

1.3 Contributions

We summarize in the following the main contributions of this thesis.

- **Stochastic Gradient Descent with Random Features.**
  We first consider an estimator defined by mini-batched stochastic gradients and random features within the least-squares framework. This estimator can be seen as shallow networks with random weights [CS09], or also as approximate kernel methods [RR17]. We use the theory of reproducing kernel Hilbert spaces [Aro50] as rigorous mathematical framework to study the properties of the learned estimator. The approach we consider is not based on penalizations or explicit constraints. Indeed the regularization is implicit and controlled by different parameters. In particular, we present an analysis that shows how the number of random features, iterations, step-size and mini-batch size control the stability and learning properties of the solution. By deriving finite sample bounds, we investigate how optimal learning rates can be achieved with different parameter choices and how these choices govern the interplay between statistical and computational performances.

- **FALKON.**
  We then propose a new algorithm that combines the Nyström approximation with preconditioned conjugate gradient [Saa03]. We use this technique to approximate the kernel ridge regression (KRR) problem, but also to efficiently compute a preconditioner to be used with the conjugate gradient optimization method. We prove that this new algorithm, that we named FALKON, can at the same time preserve optimal theoretical guarantees and run on millions of points utilizing only a fraction of the computational resources of previously proposed methods. More precisely, we take a substantial step in provably reducing the computational requirements, showing that, up to logarithmic factors, a time/memory complexity of $\widetilde{\mathcal{O}}(n\sqrt{n})$ and $\mathcal{O}(n)$ is sufficient for optimal statistical accuracy. The theoretical analysis that we derive provides optimal statistical rates both in a basic setting and under refined benign conditions for which faster rates are possible. Further, we broadly test on available large scale data-sets the empirical performances of FALKON, showing that even on a single machine FALKON can outperform state of the art methods both in terms of time efficiency and prediction accuracy.

- **BLESS.**
  We consider how to speed up leverage score sampling in the case of positive semi-definite matrices. We first propose and study BLESS, a novel algorithm for approximate leverage scores sampling. Our analysis shows that the new algorithm can achieve state of the art accuracy and computational complexity without requiring distributed resources. The key idea is to follow a coarse to fine strategy, alternating uniform and leverage scores sampling on sets of increasing size. Our second contribution is considering leverage score sampling in statistical learning with least squares. We extend the FALKON algorithm. In particular, we study the impact of replacing uniform sampling with leverage score sampling. We prove that the derived method still achieves optimal learning bounds but the time and memory is now $\widetilde{\mathcal{O}}(n\widehat{\mathcal{N}})$, and $\widetilde{\mathcal{O}}(\widehat{\mathcal{N}}^2)$ respectively, where $\widehat{\mathcal{N}}$ is the effective dimension which is never larger, and possibly much smaller, than $\sqrt{n}$.

- **BKB.**
  We present the BKB (budgeted kernel bandit) algorithm: a kernelized bandit optimization algorithm that achieves near-optimal regret with a computational complexity drastically smaller than GP-UCB. This is achieved without assumptions on the complexity of the input or on the kernel function. BKB leverages several well-known tools: a DTC approximation of the posterior variance, based on inducing points, and a confidence interval construction based on state-of-the-art self-normalized concentration inequalities [AYPS11]. It also introduces two novel tools: a selection strategy to select inducing points based on ridge leverage score (RLS) sampling that is provably accurate, and an approximate confidence interval that is not only nearly as accurate as the one of GP-UCB, but also efficient. Moreover denoting with $\widehat{\mathcal{N}}$ the effective dimension of the problem, in a problem with A arms, using a set of $\mathcal{O}(\widehat{\mathcal{N}})$ inducing points results in an algorithm with $\mathcal{O}(A\widehat{\mathcal{N}}^2)$ per-step runtime and $\mathcal{O}(A\widehat{\mathcal{N}})$ space, a significant improvement over the $\mathcal{O}(At^2)$ time and $\mathcal{O}(At)$ space cost of GP-UCB.

1.4 Structure of the Thesis

In the following, we briefly describe the structure of the thesis.

- **Chapter 2.**
  We introduce the supervised learning problem in the statistical learning setting and recall some algorithms to learn with kernels. In particular, we define the problem setting in Section 2.1 and Section 2.2, where we recall the theory of reproducing kernel Hilbert spaces. We report how the kernel ridge regression problem can be solved in a basic way and study its computational complexity in Section 2.3 and Section 2.4. We then state their theoretical guarantees in Section 2.5.

- **Chapter 3.**
  We present our first result studying the statistical and computational trade-offs of the estimator defined by stochastic gradients and random features. This chapter is based on results published in [CRR18]. We begin defining the estimator, studying its computational complexity and discussing related methods in Section 3.1. We then present our main theoretical results that capture statistical and computational trade-offs in Section 3.2 and provide the proofs in Section 3.3. We conclude the chapter with empirical evaluations in Section 3.4.

- **Chapter 4.**
  This chapter where we present the FALKON algorithm, is based on results published in [RCR17]. We start recalling the Nyström approximation and its effect on kernel ridge regression in Section 4.1. We present the learning algorithm in Section 4.2 and its theoretical analysis in Section 4.3. We compare FALKON with previous works in Section 4.4. We

then give a generalized version of FALKON in Section 4.5. We state the proofs of all the theoretical analysis in Section 4.6, 4.7, 4.8, and 4.9. In the end, we present the empirical performance of FALKON on a wide range of datasets in Section 4.10.

- **Chapter 5.**
  This chapter is based on results published in [RCCR18] We present how to speed up leverage score sampling and how the computational complexity of FALKON can be further reduced using the newly presented algorithms. Section 5.1 is dedicated to recalling leverage score sampling, present the new BLESS algorithm and compare it with previous leverage score sampling algorithms. Section 5.2 states the theoretical guarantees and proofs of BLESS. In Section 5.3 we describe how BLESS can be used in conjunction with FALKON to derive fast solvers in the statistical learning setting. In Section 5.4 we give the theoretical analysis of FALKON-BLESS. We give empirical evaluations of both BLESS and FALKON-BLESS in Section 5.5.

- **Chapter 6.**
  We introduce the bandit optimization problem and how it can be solved using the non-parametric GP-UCB algorithm. In particular, we define in Section 6.1 the bandit problem and recall the general learning framework. We also recall the two main ingredients with which the GP-UCB algorithm is built: the upper confidence bound principle as optimization strategy in Section 6.2; and the Gaussian processes as prior over the reward function of the bandit problem in Section 6.3. We then state the GP-UCB algorithm and its computational complexity in Section 6.4.

- **Chapter 7.**
  We present our last result: the BKB algorithm. This chapter is based on results published in [CCL$^+$19]. We state the algorithm, study its computational complexity and present its theoretical analysis in Section 7.1. We then compare it with other methods and discuss how some of the assumptions made can be relaxed in Section 7.2. We finally give the detailed proofs of the theoretical analysis in Section 7.3.

Chapter 2

Learning with Kernels in the Statistical Learning Setting

Solving a supervised learning problem consists in finding a function $\widehat{f}$ that describes well an input/output relation given a set $Z = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of input/output pairs. Given a new input point x_{new} not included in Z, a function $\widehat{f}$ is said to generalize if it can give a good estimate $\widehat{f}(x_{new})$ of the new output y_{new}.

The input/output pairs are sampled and often noisy. One of the challenges is to learn a function that is not strongly biased by the noise. When this happens the function is said to *overfit* the data. To contrast overfitting a procedure known as *regularization* is needed. Regularization is a technique that, adding information/constraints, prevents overfitting and ensures generalization.

A second challenge deals the computational aspect of learning a function. Given infinite time, trying all possible functions to find the optimal would be the ideal choice. This is of course not doable in practice and finding the procedure that requires the least amount of time to learn the best possible estimator is a question of major interest.

In this chapter, we introduce in detail these concepts and challenges. In particular, we see how the problem can be formalized in the statistical learning setting, we present two algorithms to solve the learning problem, and we see how the quality of the estimators can be measured and studied theoretically.

2.1 Statistical Learning Theory

In a supervised learning problem we assume there exists an input space $\mathcal{X}$ and output space $\mathcal{Y}$ from which pairs of points can be sampled. In particular, in statistical learning theory the data

space $\mathcal{X} \times \mathcal{Y}$ is modeled as a probability space with probability distribution ρ. The input/output pairs are assumed to be sampled independently and identically from the distribution, that is $\{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \rho^n$, and form the so called *training set*.

Given an estimator f, to measure the quality of an estimate $f(x)$ with respect to the corresponding y, a loss $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$, is defined. Considering $\mathcal{Y} \subseteq \mathbb{R}$, for $a, b \in \mathbb{R}$ one example is the squared loss

$$\mathcal{L}(a, b) = (a - b)^2 \tag{2.1}$$

which is the most common loss for solving regression tasks, measuring the deviation between the real output and the predicted value (i.e. $a = y, b = f(x)$). In classification tasks where $\mathcal{Y} \in \{-1, 1\}$ the squared loss can still be used, but other natural choices are for example the logistic loss

$$\mathcal{L}(a, b) = \log(1 + e^{-ab}), \tag{2.2}$$

and the hinge loss

$$\mathcal{L}(a, b) = |1 - ab|_+. \tag{2.3}$$

In this thesis we are going to mainly consider the squared loss.

As we have already stated, the ideal function f should predict correctly all possible inputs/outputs generated by ρ. Then, to measure the quality of a function f we define the so called *expected risk*

$$\mathcal{E}(f) = \int_{\mathcal{X} \times \mathcal{Y}} (f(x) - y)^2 d\rho(x, y). \tag{2.4}$$

This error measure takes into account all the possible points generated by ρ and weights the loss suffered by each point according to the probability to be sampled.

Defining the *target set* $\mathcal{T}$ as the space of functions for which the expected risk is well defined, the ideal solution of the learning problem is a function whose excess risk is close to

$$\inf_{f \in \mathcal{T}} \mathcal{E}(f) \tag{2.5}$$

For the squared loss it is possible to derive the above quantity. The following function, known as *target function*, achieves the infimum of the expected risk,

$$f_\rho(x) = \int_{\mathcal{Y}} y \, d\rho(y|x). \tag{2.6}$$

Unfortunately the target function cannot be computed directly since the distribution is unknown, and we do not have access to the set $\mathcal{T}$. So the question that arises is how to efficiently find a provably good solution on the basis of a given training set.

For these reasons, a set of candidate solutions $\mathcal{H} \subset \mathcal{T}$ is fixed and, given the set of training points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, an approximation of the expected risk is defined as

$$\widehat{\mathcal{E}}(f) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) \tag{2.7}$$

and an empirical counterpart of problem (2.5) is

$$\inf_{f \in \mathcal{H}} \widehat{\mathcal{E}}(f). \tag{2.8}$$

Note that solving the above problem can lead to solutions which are not close to the solution of the original problem (2.5). The reasons are multiple, one of which is the choice of the hypothesis space $\mathcal{H}$. Indeed, if on the one hand a large space $\mathcal{H}$ may imply that the infimum of the two problems are close

$$\inf_{f \in \mathcal{H}} \mathcal{E}(f) \approx \inf_{f \in \mathcal{T}} \mathcal{E}(f), \tag{2.9}$$

on the other hand the solution of the empirical approximation over the set $\mathcal{H}$ may be far away from the one of the expected problem

$$\inf_{f \in \mathcal{H}} \widehat{\mathcal{E}}(f) \not\approx \inf_{f \in \mathcal{H}} \mathcal{E}(f). \tag{2.10}$$

To avoid this issue, the minimization problem (2.8) is typically approached adding further constraints or penalizations to reduce the complexity of the learned function.

This procedure, called *regularization*, can be performed in different ways. One of them, known as *Tikhonov regularization*, consist in explicitly adding a penalization term to the empirical minimization problem. Defining a functional $\mathit{Pen} : \mathcal{H} \to [0, \infty)$ the regularized version of the empirical problem is defined as

$$\inf_{f \in \mathcal{H}} \widehat{\mathcal{E}}_\lambda(f), \qquad \widehat{\mathcal{E}}_\lambda(f) = \widehat{\mathcal{E}}(f) + \lambda \mathit{Pen}(f), \tag{2.11}$$

where $\lambda > 0$ is a parameter that balance the trade-off between the data fitting term and the penalization term. Another way is to explicitly add a constraint to the hypothesis set. This can be formalized as restrict the minimization problem (2.8) to a subset $\mathcal{H}_\lambda \subset \mathcal{H}$ of candidate solutions

$$\inf_{f \in \mathcal{H}_\lambda} \widehat{\mathcal{E}}(f), \tag{2.12}$$

with $\mathcal{H}_\lambda = \{f \in \mathcal{H} \mid \|f\|_{\mathcal{H}} \leq \lambda\}$, where $\lambda > 0$ controls the importance of the constraint.

These different regularization techniques prevent the learned estimator to overfit the training data if $\mathcal{H}$ is large, producing solutions with good generalization properties. We will also see in the next chapters how regularization can affect the computational costs of a learning algorithm and how constraining the computational resources can have an effect on regularization.

2.1.1 Measuring Generalization

We have seen so far how from the ideal learning problem (2.5) we can get empirically solvable problems of the form (2.11) or (2.12). We see now how to measure the quality of the solutions of these problems. Fixed a loss function and a hypothesis space, consider a function $\widehat{f}_\lambda$ learned through the minimization of a regularized empirical problem over a set of n with regularization parameter λ. Then a natural quantity to measure the quality of the estimator is the *excess risk*, defined as

$$\mathcal{R}(\widehat{f}_\lambda) = \mathcal{E}(\widehat{f}_\lambda) - \inf_{f \in \mathcal{H}} \mathcal{E}(f). \tag{2.13}$$

Notice that even if the algorithm that generates the estimator $\widehat{f}_\lambda$ can be deterministic, this quantity is still stochastic because of its dependence on the training set from which it is learned. Then, for a certain confidence $\delta \in [0, 1)$, the above quantity can then be studied through probabilistic inequalities of the form

$$\mathbb{P}\left(\mathcal{E}(\widehat{f}_\lambda) - \inf_{f \in \mathcal{H}} \mathcal{E}(f) \leq \mathcal{B}(n, \rho, \lambda, \delta)\right) \geq 1 - \delta, \tag{2.14}$$

where $\mathcal{B}(n, \rho, \lambda, \delta)$ is a quantity that depends on the number of points n, the distribution ρ from which the points are sampled, the imposed regularization λ and the required confidence δ. Ideally we would like an algorithm to be able to generate good solutions $\widehat{f}_\lambda$ such that, for a certain level of regularization and with a certain confidence, the quantity $\mathcal{B}(n, \rho, \lambda, \delta)$ goes to 0 as fast as possible as n goes to infinity. This would indicate that such a solution reaches the best possible solution as the number of points increases.

So formalizing this desirable property we say that an algorithm is *consistent* if generates solutions $\widehat{f}_\lambda$ such that (2.14) holds for a $\mathcal{B}(n, \rho, \lambda, \delta)$ such that

$$\lim_{n \to \infty} \mathcal{B}(n, \rho, \lambda, \delta) = 0 \tag{2.15}$$

for a proper λ. Further an algorithm is said to be *universally consistent* if it is consistent for all possible measure ρ.

2.2 Reproducing Kernel Hilbert Spaces

The hypothesis spaces that we are going to consider in the rest of this thesis are known as Reproducing Kernel Hilbert Spaces (RKHS)[Aro50]. They are some of the most useful spaces of functions in a wide range of applied sciences including machine learning. They allow to define potentially non-linear and nonparametric models. In this section, we are going to define a RKHS and state some of its properties which will be useful throughout the thesis.

Let $\mathcal{X}$ and $\mathcal{Y} \subseteq \mathbb{R}$ be sets. A RKHS is a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \to \mathcal{Y}$ for which is defined a continuous evaluation function $e_x : \mathcal{H} \to \mathcal{Y}$ such that

$$e_x(f) = f(x), \tag{2.16}$$

for any $f \in \mathcal{H}$ and $x \in \mathcal{X}$.

The evaluation function (2.16) is related to a *positive definite* (PD) function known as *reproducing kernel* that gives the name to this space. Formally given a Hilbert space $\mathcal{H}$, a reproducing kernel is a symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for all $x \in \mathcal{X}$

$$k(x, \cdot) \in \mathcal{H}, \tag{2.17}$$

and for all $f \in \mathcal{H}$, defining $k_x(\cdot) = k(x, \cdot)$

$$\langle f, k_x \rangle_{\mathcal{H}} = f(x). \tag{2.18}$$

It can be proved that every reproducing kernel k induces a unique RKHS and every RKHS has a unique reproducing kernel.

The above definition of RKHS is not particularly revealing of how to design an RKHS, then we state an equivalent characterization of RKHS: every positive definite function defines a unique RKHS, of which is the unique reproducing kernel. In particular we say that a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if it is symmetric and for all $N \in \mathbb{N}$, $a_1, \dots, a_N \in \mathbb{R}$ and $x_1, \dots, x_N \in \mathcal{X}$ the following holds

$$\sum_{i,j=1}^{N} a_i a_j k(x_i, x_j) \geq 0. \tag{2.19}$$

This last definition of kernel as PD function allows to easily prove some properties of reproducing kernels. For example it is easy to see that finite sums of PD functions are PD functions and then reproducing kernels. Another examples is that the product of kernels is still a kernel. In more details, given $(k_i)_{i=1}^m$ collection of m kernels with domain over as many sets $(\mathcal{X}_i)_{i=1}^m$, for any $x_j, x'_j \in \mathcal{X}_j$ with $j = 1, \dots, m$, the function

$$k((x_1, \dots, x_m), (x'_1, \dots, x'_m)) = k_1(x_1, x'_1) \dots k_m(x_m, x'_m) \tag{2.20}$$

is a kernel on $\widetilde{\mathcal{X}} \times \widetilde{\mathcal{X}}$, with $\widetilde{\mathcal{X}} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$.

Three popular examples of PD kernels are the *linear kernel*

$$k(x, x') = \langle x, x' \rangle ; \tag{2.21}$$

the *polinomial kernel* of degree $d \in \mathbb{N}$

$$k(x, x') = (\langle x, x' \rangle + 1)^d; \tag{2.22}$$

and the *Gaussian kernel* with bandwidth $\sigma > 0$

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}. \tag{2.23}$$

Lastly we state the connection between reproducing kernels and the so called *feature maps*. Let $\mathcal{F}$ be a Hilbert space called *feature space*, for each reproducing kernel k there exists a function $\Phi : \mathcal{X} \to \mathcal{F}$ (known as *feature map*), such that

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}}, \qquad \forall x \in \mathcal{X}. \tag{2.24}$$

This characterization suggests that every PD function and corresponding RKHS has at least one associated feature map satisfying the above equation. Moreover it unveils the property of kernels of implicitly mapping the input data $\mathcal{X}$ into a higher and potentially infinite dimensional space through the feature map Φ.

A trivial example of feature map is $\Phi(x) = k_x$, implying $\mathcal{F} = \mathcal{H}$ for the kernel $k(x, x') = \langle k_x, k_{x'} \rangle_{\mathcal{H}} = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}}$. Otherwise considering $\mathcal{F} = \ell^2$ space of squared summable sequences and defining $(\phi_j)_j$ any orthonormal basis in $\mathcal{H}$, then $\Phi(x) = (\phi_j(x))_j$ is a feature map that defines the kernel $k(x, x') = \sum_{j=1}^{\infty} \phi_j(x)\phi_j(x') = \langle k_x, k_{x'} \rangle_{\mathcal{H}} = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}}$.

2.3 Kernel Ridge Regression

Tikhonov regularization provides a way to approximate the solution of minimizing the expected risk given data. In this section, we investigate the computational aspects of the above problem choosing the hypothesis space $\mathcal{H}$ to be a RKHS, the regularizer *Pen* the corresponding squared norm and the loss function to be the squared loss.

We consider the regularized empirical minimization problem over n points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. Define with k the reproducing kernel associated to the RKHS $\mathcal{H}$, and let $\lambda > 0$ be the regularization parameter. We can then write the minimization problem as

$$\min_{f \in \mathcal{H}} \widehat{\mathcal{E}}_\lambda(f), \qquad \widehat{\mathcal{E}}_\lambda(f) = \frac{1}{n} \sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2 + \lambda \|f\|_{\mathcal{H}}^2. \tag{2.25}$$

Because the squared loss is a convex and continuous function, the objective function is strongly convex, continuous and coercive. Then the solution of the problem exists and is unique.

Being $\mathcal{H}$ a potentially infinite dimensional space, in the form stated in equation (2.25), the minimization problem does not explicitly suggest how to actually compute the solution. Thanks to the so called *representer theorem* it can be proved that the solution of the above problem can be

written as a linear combination of the kernel function evaluated at the training set points [SS02]. More precisely, the function $\widehat{f}_\lambda \in \mathcal{H}$ of the form

$$\widehat{f}_\lambda(x) = \sum_{i=1}^{n} k\left(x, x_i\right) c_i, \quad \forall x \in \mathcal{X} \tag{2.26}$$

is solution of (2.25) for a certain $\widehat{c}_\lambda = (c_1, \ldots, c_n)^\top \in \mathbb{R}^n$, where we denote with $A^\top$ the transpose of any matrix A. Then the quantities to be computed are the coefficients $\widehat{c}_\lambda$. As we see now, this result allows to write a finite dimensional minimization problem whose solution is related to (2.25) through (2.26). Let $\widehat{K}$ be the $n \times n$ matrix with entries $(\widehat{K})_{i,j} = k(x_i, x_j)$ called the *kernel matrix*, and note that for any function $\widehat{f}_\lambda$ as in (2.26) the corresponding norm in $\mathcal{H}$ can be written as

$$\left\|\widehat{f}_\lambda\right\|_{\mathcal{H}}^2 = \sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) = \left\langle c, \widehat{K}c \right\rangle_{\mathbb{R}^n}. \tag{2.27}$$

Then, plugging the representation (2.26) in problem (2.25), we can derive the following minimization problem

$$\min_{c \in \mathbb{R}^n} \frac{1}{n} \left\|\widehat{K}c - \widehat{y}\right\|_{\mathbb{R}^n}^2 + \lambda \left\langle c, \widehat{K}c \right\rangle_{\mathbb{R}^n}, \tag{2.28}$$

with $\widehat{y} = (y_1, \ldots, y_n)^\top$.

The squared loss allows to write the solution of the above problem in closed form. Taking the gradient of (2.28) and setting it equal to zero we recover the following linear system to be solved with respect to c

$$\widehat{K}(\widehat{K} + \lambda n I)c = \widehat{K}\widehat{y}. \tag{2.29}$$

Note that the simpler linear system

$$(\widehat{K} + \lambda n I)c = \widehat{y}. \tag{2.30}$$

shares the same solution and can be derived by first setting the gradient of (2.25) equal to zero and then plugging in the representation (2.26). This latter linear system requires less computational resources to be solved. The solution of (2.30) can then be written as

$$\widehat{c}_\lambda = (\widehat{K} + \lambda n I)^{-1}\widehat{y}, \tag{2.31}$$

defining what is known as the Kernel Ridge Regression (KRR) estimator

$$\widehat{f}_\lambda(x) = \sum_{i=1}^{n} k\left(x, x_i\right) (\widehat{c}_\lambda)_i. \tag{2.32}$$

The computational complexity of the overall problem is in computing $\widehat{c}_\lambda$. In details, a cost of $\mathcal{O}(n^2)$ in memory is needed to store the kernel matrix $\widehat{K}$, and, denoting with c_k the cost of evaluating the kernel function for one pair of points, $\mathcal{O}(n^3 + c_k n^2)$ is the cost in time for inverting an $n \times n$ matrix and constructing the kernel matrix. This analysis shows that computing (2.31) for large datasets is challenging.

2.4 Gradient Descent Learning

In the previous section we have seen how Tikhonov regularization can be used to approximatively solve the learning problem over a RKHS. In particular we have seen how to recover an empirical finite dimensional minimization problem and how this can be solved directly in closed form. We now focus on how to solve this empirical problem in an iterative way such that no explicit regularization is needed.

For the sake of simplicity, we are going to assume $\widehat{K}$ invertible, but all the reasoning can be extended considering simply the pseudoinverse.

An iterative solver defines a sequence of empirical solutions. The first elements of the sequence will be a rough approximation of the solution, and the latest will be the most refined and close to the exact minimizer of the empirical problem. Consider the empirical problem for squared loss for an RKHS $\mathcal{H}$

$$\min_{f\in\mathcal{H}} \widehat{\mathcal{E}}(f), \qquad \widehat{\mathcal{E}}(f) = \frac{1}{n}\sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2. \tag{2.33}$$

If we follow the same reasoning used in the previous section, by the representer theorem we recover the following finite dimensional minimization problem

$$\min_{c\in\mathbb{R}^n} \widehat{\mathcal{E}}(c), \qquad \widehat{\mathcal{E}}(c) = \frac{1}{n}\left\|\widehat{K}c - \widehat{y}\right\|_{\mathbb{R}^n}^2, \tag{2.34}$$

with minimizer $\widehat{c}_* = \widehat{K}^{-1}\widehat{y}$. Minimizing exactly this problem may lead to overfitting because of lack of explicit regularization. But we now see that with some care in the optimization procedure regularization can be implicitly induced. Using gradient descent (GD) to solve the above minimization problem gives the following iteration sequences for $t = 1, \dots, t_{max}$

$$c_t = c_{t-1} - \frac{2\gamma}{n}\widehat{K}\left(\widehat{K}c_{t-1} - \widehat{y}\right) \tag{2.35}$$

with initialization step $c_0 = 0$ and stepsize $\gamma > 0$. Being (2.34) a convex problem we know that the above sequence with enough iterations will eventually converge to the exact minimizer. The idea of iterative regularization is that early termination of the iteration has a regularizing effect, leading to an approximate solution of the ERM problem as Tikhonov regularization does. It can be proved by induction that we can also write (2.35) as

$$c_t = \frac{2\gamma}{n}\sum_{i=0}^{t}\left(I - \frac{2\gamma}{n}\widehat{K}^2\right)^i \widehat{K}\widehat{y}. \tag{2.36}$$

Denoting with $\|\cdot\|$ the operator norm for a bounded linear operator A, and recalling the Neumann series, we know that for each matrix A such that $\|A\| < 1$

$$\sum_{i=0}^{\infty}(I - A)^i = A^{-1}. \tag{2.37}$$

This suggests that if instead we consider a truncated series

$$\sum_{i=0}^{T}(I-A)^i \approx A^{-1}, \tag{2.38}$$

this produces an approximation of the inverse of the matrix A as accurate as T approaches infinity. If we now take $A = \frac{2\gamma}{n}\widehat{K}$ with γ such that $\|A\| < 1$, then we have

$$c_{t_{max}} = \frac{2\gamma}{n}\sum_{i=0}^{t_{max}}\left(I - \frac{2\gamma}{n}\widehat{K}\right)^i \widehat{y} \approx \widehat{K}^{-1}\widehat{y} = \widehat{c}_* \tag{2.39}$$

being an approximation of the exact minimizer, with the approximation level driven by the stopping rule t_{max}. We then refer to the function

$$\widehat{f}_{t_{max}}(x) = \sum_{i=1}^{n} k\left(x, x_i\right)\left(\widehat{c}_{t_{max}}\right)_i \tag{2.40}$$

as Gradient Descent estimator or as L^2-boosting estimator or Landweber estimator.

The computational complexity of this method depends on the stopping criterion. Each iteration (2.35) costs $\mathcal{O}(n^2)$ because of the matrix vector product. Keeping into account also the cost for computing the kernel matrix we have an overall cost in time of $\mathcal{O}(c_k n^2 + t_{max} n^2)$ and $\mathcal{O}(n^2)$ in space.

2.5 Learning Bounds

We have seen in the previous 2 sections how to compute two estimators based respectively on Tikhonov regularization and Gradient Descent. These estimators depends on some specific choices of regularization: the λ value for Tikhonov and the stepsize and number of iteration for Landweber. In this section, we see how these estimators can be studied theoretically in order to have certain statistical guarantees, and how the analysis suggests the right level of regularization to impose.

Recall from Section 2.1.1 that for a given estimator $\widehat{f}_\lambda$, and denoting with λ the regularization imposed, we would like to prove that

$$\mathbb{P}\left(\mathcal{E}(\widehat{f}_\lambda) - \inf_{f\in\mathcal{H}}\mathcal{E}(f) \leq \mathcal{B}(n,\rho,\lambda,\delta)\right) \geq 1-\delta, \tag{2.41}$$

for a certain $\mathcal{B}(n,\rho,\lambda,\delta)$ going to 0 as n goes to infinity. To theoretically prove the above bounds certain assumptions are required. In details we are going to identify two main class of assumptions which allows to recover two different regimes for the bound. The first one introduced in

Section 2.5.1 is a class of mild assumptions which covers a worst case regime. The second one Section 2.5.2 is a class of more refined assumptions which allows to get faster rates. For each one of this classes we present the bounds for the KRR and GD estimator.

2.5.1 Basic

Using as hypothesis space an RKHS, to derive statistical bounds of the form (2.41), we require the reproducing kernel to be bounded. This can be formalized by the following assumption.

Assumption 1. *There exists $\kappa \geq 1$ such that $k(x,x) \leq \kappa^2$ for any $x \in \mathcal{X}$*

Another required standard assumption in the context of non-parametric regression (see [CDV07]), consists in assuming a minimum for the expected risk, over the space of functions induced by the kernel.

Assumption 2. *If $\mathcal{H}$ is the RKHS with kernel k, there exists $f_{\mathcal{H}} \in \mathcal{H}$ such that*

$$\mathcal{E}(f_{\mathcal{H}}) = \inf_{f \in \mathcal{H}} \mathcal{E}(f).$$

This assumption state the existence of $f_{\mathcal{H}}$ we will see how it can be refined in the next section.

We also need some basic assumption on the data distribution. For all $x \in \mathcal{X}$, we denote by $\rho(y|x)$ the conditional probability of ρ and by $\rho_{\mathcal{X}}$ the corresponding marginal probability on $\mathcal{X}$. We need a standard moment assumption to derive probabilistic results. The assumption can be stated in slightly different ways, one of which is the following.

Assumption 3. *For any $x \in \mathcal{X}$, there exist σ, b satisfying $0 \leq \sigma \leq b$ such that*

$$\int_{\mathcal{Y}} |y - f_{\mathcal{H}}(x)|^p d\rho(y|x) \leq \frac{1}{2} p! \sigma^2 b^{p-2}, \qquad \forall p \geq 2 \in \mathbb{N}. \tag{2.42}$$

Note that the above assumption holds when y is bounded, sub-Gaussian or sub-exponential.

The two following propositions give a bound of the form (2.41) to the KRR estimator and the Landweber estimator, showing how regularization influence the bound.

Proposition 1 (from [CDV07])**.** *Let $\widehat{f}_\lambda$ be the KRR estimator as in* (2.32)*, and let $\delta \in (0,1)$. Under assumption Assumptions 1,2,3, the following holds with probability at least $1-\delta$*

$$\mathcal{R}(\widehat{f}_\lambda) \lesssim \ \lambda \ + \ \frac{1}{n\lambda} \ \log \frac{1}{\delta}, \tag{2.43}$$

where we ignored the constants which do not depend on n, λ, δ.

Proposition 2 (from [YRC07]). *Let $\widehat{f}_{t_{max}}$ be the GD estimator as in* (2.40)*, and let $\delta \in (0,1)$ and $\gamma \in (0, \kappa^{-2}]$. Under assumption Assumptions 1,2,3, the following holds with probability at least $1-\delta$*

$$\mathcal{R}(\widehat{f}_{t_{max}}) \lesssim \frac{1}{\gamma t_{max}} + \frac{\gamma t_{max}}{n} \log \frac{1}{\delta}, \tag{2.44}$$

where we ignored the constants which do not depend on n, t_{max}, δ.

From the above two results it is then possible to derive the optimal choice of regularization, which consist in an optimal choice for λ for KRR and the early stopping condition for Landweber. These choices are summed up in the following corollaries.

Corollary 1. *Let $\widehat{f}_{\lambda}$ be the KRR estimator as in* (2.32)*, and let $\delta \in (0,1)$ and $\lambda > 0$. Under assumption Assumptions 1,2,3, choosing λ_n such that*

$$\lambda_n = \frac{1}{\sqrt{n}} \tag{2.45}$$

the following holds with probability at least $1-\delta$

$$\mathcal{R}(\widehat{f}_{\lambda_n}) \lesssim \frac{1}{\sqrt{n}} \log \frac{1}{\delta}, \tag{2.46}$$

where we ignored the constants which do not depend on n, λ_n, δ.

Corollary 2. *Let $\widehat{f}^*_{t_{max}}$ be the GD estimator as in* (2.40)*, and let $\delta \in (0,1)$ and $\gamma = \kappa^{-2}$. Under assumption Assumptions 1,2,3, choosing t_{max} as*

$$t_{max} = \sqrt{n} \tag{2.47}$$

the following holds with probability at least $1-\delta$

$$\mathcal{R}(\widehat{f}_{t_{max}}) \lesssim \frac{1}{\sqrt{n}} \log \frac{1}{\delta}, \tag{2.48}$$

where we ignored the constants which do not depend on $n, \gamma, t_{max}, \delta$.

2.5.2 Refined

We next discuss how the above results can be refined under an additional regularity assumption. We need some preliminary definitions. Let $\mathcal{H}$ be the RKHS defined by k, and $L : L^2(\mathcal{X}, \rho_{\mathcal{X}}) \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$ the integral operator

$$Lf(x) = \int_{\mathcal{X}} k(x, x') f(x') d\rho_{\mathcal{X}}(x'), \qquad \forall f \in L^2(\mathcal{X}, \rho_{\mathcal{X}}), x \in \mathcal{X},$$

where $L^2(\mathcal{X}, \rho_{\mathcal{X}}) = \{f : \mathcal{X} \to \mathbb{R} \ : \ \|f\|_\rho^2 = \int_{\mathcal{X}} |f|^2 d\rho_{\mathcal{X}} < \infty\}$. The above operator is symmetric and positive definite. Moreover, Assumption 1 ensures that the kernel is bounded, which in turn ensures L is trace class, hence compact [SC08]. Define also the operator $C : \mathcal{H} \to \mathcal{H}$ as

$$\langle h, Ch' \rangle_{\mathcal{H}} = \int_{\mathcal{X}} h(x)h'(x)d\rho_{\mathcal{X}}(x), \qquad \forall h, h' \in \mathcal{H}$$

We now define a quantity known as *effective dimension*

Definition 1. *For any $\lambda > 0$, we define the effective dimension the quantity*

$$\mathcal{N}(\lambda) = \mathrm{Tr}((L + \lambda I)^{-1} L). \tag{2.49}$$

This quantity can be seen as a measure of the size of $\mathcal{H}$ and with certain assumptions allows to improve the rates of convergence. For example, one can assume that the effective dimension has a polynomial decrease in λ.

Assumption 4. *For any $\lambda > 0$, assume that there exist $Q > 0$ and $\alpha \in [0, 1]$ such that*

$$\mathcal{N}(\lambda) \leq Q^2 \lambda^{-\alpha}. \tag{2.50}$$

Condition (2.50) describes the *capacity/complexity* of the RKHS $\mathcal{H}$ and the measure ρ. It is equivalent to classic entropy/covering number conditions, see e.g. [SC08]. The case $\alpha = 1$ corresponds to making no assumptions on the kernel, and reduces to the worst case analysis in the previous section. The smaller is α the more stringent is the capacity condition. A classic example is considering $\mathcal{X} = \mathbb{R}^d$ with $d\rho_{\mathcal{X}}(x) = p(x)dx$, where p is a probability density, strictly positive and bounded away from zero, and $\mathcal{H}$ to be a Sobolev space with smoothness $s > d/2$. Indeed, in this case $\alpha = d/2s$ and classical nonparametric statistics assumptions are recovered as a special case. Note that in particular the worst case is $s = d/2$.

We now state a more refined version of Assumption 2.

Assumption 5. *Assume there exists $1/2 \leq r \leq 1$ and $g \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$ such that*

$$f_{\mathcal{H}}(x) = (L^r g)(x), \tag{2.51}$$

with $\|g\| \leq R$, where $R > 0$.

This assumption correspond in the inverse problem literature to what is called *source condition*, where it is expressed in this equivalent form [VRC$^+$05, DVRC06].

Assumption 6. *Assume there exists $1/2 \leq r \leq 1$ and $h \in \mathcal{H}$ such that*

$$f_{\mathcal{H}}(x) = (C^{r-1/2} h)(x), \tag{2.52}$$

with $\|h\| \leq R$, where $R > 0$.

These last two equivalent assumptions are regularity conditions commonly used in approximation theory to control the bias of the estimator. Note that for $r = 1/2$ we recover Assumption 2 (see [SZ03]), but this is a relaxed version which measures the regularity of $f_{\mathcal{H}}$: if r is big $f_{\mathcal{H}}$ is regular and rates are faster. For further discussions on the interpretation of the conditions above see [CDV07, SHS$^+$09, Bac13, RCR15].

We can now state the equivalent bounds and rates under the more refined assumptions.

Proposition 3 (from [CDV07])**.** *Let $\widehat{f}_\lambda$ be the KRR estimator as in* (2.32)*, and let $\delta \in (0,1)$ Under assumption Assumptions 1,3, 4, 6, the following holds with probability at least $1-\delta$*

$$\mathcal{R}(\widehat{f}_\lambda) \lesssim \quad \lambda^{2r} \quad + \quad \frac{\mathcal{N}(\lambda)}{n} \quad \log\frac{1}{\delta}, \tag{2.53}$$

where we ignored the constants which do not depend on n, λ, δ.

Proposition 4 (from [LR17c])**.** *Let $\widehat{f}_{t_{max}}$ be the GD estimator as in* (2.40)*, and let $\delta \in (0,1)$ Under assumption Assumptions 1,3, 4, 6, the following holds with probability at least $1-\delta$*

$$\mathcal{R}(\widehat{f}_{t_{max}}) \lesssim \quad \left(\frac{1}{\gamma t_{max}}\right)^{2r} \quad + \quad \frac{\mathcal{N}(1/(\gamma t_{max}))}{n} \quad \log\frac{1}{\delta}, \tag{2.54}$$

where we ignored the constants which do not depend on n, t_{max}, δ.

Corollary 3. *Let $\widehat{f}_\lambda$ be the KRR estimator as in* (2.32)*, and let $\delta \in (0,1)$ and $\lambda > 0$. Under assumption Assumptions 1,3, 4, 6, choosing λ_n such that*

$$\lambda_n = n^{-\frac{1}{2r+\alpha}} \tag{2.55}$$

the following holds with probability at least $1-\delta$

$$\mathcal{R}(\widehat{f}_{\lambda_n}) \lesssim \quad n^{-\frac{2r}{2r+\alpha}} \quad \log\frac{1}{\delta}, \tag{2.56}$$

where we ignored the constants which do not depend on n, λ^*, δ.

Corollary 4. *Let $\widehat{f}^*_{t_{max}}$ be the GD estimator as in* (2.40)*, and let $\delta \in (0,1)$ and $\gamma = \kappa^{-2}$. Under assumption Assumptions 1,3, 4, 6, choosing t_{max} as*

$$t_{max} = n^{\frac{1}{2r+\alpha}} \tag{2.57}$$

the following holds with probability at least $1-\delta$

$$\mathcal{R}(\widehat{f}_{t_{max}}) \lesssim \quad n^{-\frac{2r}{2r+\alpha}} \quad \log\frac{1}{\delta}, \tag{2.58}$$

where we ignored the constants which do not depend on $n, \gamma, t_{max}, \delta$.

Notice that the results in Section 2.5.1 can be recovered as a special case of these when choosing $r = \frac{1}{2}$ and $\alpha = 1$.

Chapter 3

Stochastic Gradient Descent with Random Features

In this chapter, we study an estimator defined by stochastic gradient [RM51] with mini-batches and random features [RR08] . These latter are typically defined by nonlinear sketching: random projections of the followed by a component-wise nonlinearity [ASW13]. In order to derive an efficient large scale learning algorithm, we investigate its application in the context of nonparametric statistical learning.

For nonparametric learning, we have seen in the previous chapter that classical methods like KRR (see Section 2.3) or L^2-boosting (see Section 2.4) requires both $\mathcal{O}(n^2)$ in memory and respectively $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2 t)$ in time, with n number of samples and t number of iterations of the L^2-boosting algorithm. To trim this computational resources, on the one hand, we use stochastic gradients to process data points individually, or in small batches, keeping good convergence rates, while reducing computational complexity [BB08]. On the other hand, we use sketching techniques to reduce data-dimensionality, hence memory requirements, by random projections [ASW13].

The considered estimator is not explicitly penalized/constrained and regularization is implicit. Indeed in the following analysis, we show how the number of random features, iterations, step-size and mini-batch size control the learning properties of the solution. By deriving finite sample bounds, we show how different parameter choices can be used to derive optimal learning rates. In particular, we show that similarly to ridge regression [SS02, RR17], a number of random features proportional to the square root of the number of samples suffice for $O(1/\sqrt{n})$ error bounds. Further, we show that for certain choices of the free parameters we can derive optimal estimators with a much smaller time and memory complexity with respect to previous methods.

3.1 Learning with Stochastic Gradients and Random Features

We consider the problem of supervised statistical learning with squared loss presented in Chapter 2. As suggested in Section 2.1 and in more details in [DGL13], the search for a solution needs to be restricted to a suitable space of hypothesis to allow efficient computations and reliable estimation. Then in this chapter we consider functions of the form

$$f(x) = \langle w, \phi_M(x) \rangle, \quad \forall x \in \mathcal{X}, \tag{3.1}$$

where $w \in \mathbb{R}^M$ and $\phi_M : \mathcal{X} \to \mathbb{R}^M$, $M \in \mathbb{N}_+$, denotes a family of finite dimensional feature maps (see below). Further, we consider a mini-batch stochastic gradient method to estimate the coefficients from data,

$$\widehat{w}_1 = 0; \qquad \widehat{w}_{t+1} = \widehat{w}_t - \gamma_t \frac{1}{b} \sum_{i=b(t-1)+1}^{bt} \big(\langle \widehat{w}_t, \phi_M(x_{j_i}) \rangle - y_{j_i} \big) \phi_M(x_{j_i}), \qquad t = 1, \ldots, t_{max}. \tag{3.2}$$

Here $t_{max} \in \mathbb{N}_+$ is the number of iterations and $J = \{j_1, \ldots, j_{bt_{max}}\}$ denotes the strategy to select training set points. In particular, in this work we assume the points to be drawn uniformly at random with replacement. Note that given this sampling strategy, one *pass* over the data is reached on average after $\lceil \frac{n}{b} \rceil$ iterations. Our analysis allows to consider multiple as well as single passes. For $b = 1$ the above algorithm reduces to a simple stochastic gradient iteration. For $b > 1$ it is a mini-batch version, where b points are used in each iteration to compute a gradient estimate. The parameter γ_t is the step-size.

The algorithm requires specifying different parameters. In the following, we study how their choice is related and can be performed to achieve optimal learning bounds. Before doing this, we further discuss the class of feature maps we consider.

3.1.1 From Sketching to Random Features, from Shallow Nets to Kernels

In this chapter, we are interested in a particular class of feature maps, namely random features [RR08]. A simple example is obtained by sketching the input data. Assume $\mathcal{X} \subseteq \mathbb{R}^d$ and

$$\phi_M(x) = \left(\langle x, s_1 \rangle, \ldots, \langle x, s_M \rangle \right),$$

where $s_1, \ldots, s_M \in \mathbb{R}^d$ is a set of identical and independent random vectors [Woo14]. More generally, we can consider features obtained by nonlinear sketching

$$\phi_M(x) = \left(\sigma(\langle x, s_1 \rangle), \ldots, \sigma(\langle x, s_M \rangle) \right), \tag{3.3}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear function, for example $\sigma(a) = \cos(a)$ [RR08], $\sigma(a) = |a|_+ = \max(a, 0)$, $a \in \mathbb{R}$ [CS09]. If we write the corresponding function (3.1) explicitly, we get

$$f(x) = \sum_{j=1}^{M} w^j \sigma(\langle s_j, x \rangle), \quad \forall x \in \mathcal{X}. \tag{3.4}$$

that is as shallow neural nets with random weights [CS09] (offsets can be added easily).
For many examples of random features the inner product,

$$\langle \phi_M(x), \phi_M(x') \rangle = \sum_{j=1}^{M} \sigma(\langle x, s_j \rangle) \sigma(\langle x', s_j \rangle), \tag{3.5}$$

can be shown to converge to a corresponding positive definite kernel k as M tends to infinity [RR08, SS15]. We now show some examples of kernels determined by specific choices of random features.

Example 1 (Random features and kernel)**.** *Let $\sigma(a) = \cos(a)$ and consider $(\langle x, s \rangle + b)$ in place of the inner product $\langle x, s \rangle$, with s drawn from a standard Gaussian distribution with variance σ^2, and b uniformly from $[0, 2\pi]$. These are the so called Fourier random features and recover the Gaussian kernel $k(x, x') = e^{-\|x-x'\|^2/2\sigma^2}$ [RR08] as M increases. If instead $\sigma(a) = a$, and the s is sampled according to a standard Gaussian the linear kernel $k(x, x') = \sigma^2 \langle x, x' \rangle$ is recovered in the limit. [HXGD14].*

These last observations allow to establish a connection with kernel methods [SS02] and the theory of reproducing kernel Hilbert spaces [Aro50]. As introduced in Section 2.2, a reproducing kernel Hilbert space $\mathcal{H}$ is a Hilbert space of functions for which there is a symmetric positive definite function[1] $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ called reproducing kernel, such that $k(x, \cdot) \in \mathcal{H}$ and $\langle f, k(x, \cdot) \rangle = f(x)$ for all $f \in \mathcal{H}, x \in \mathcal{X}$. It is also useful to recall that k is a reproducing kernel if and only if there exists a Hilbert (feature) space $\mathcal{F}$ and a (feature) map $\phi : \mathcal{X} \to \mathcal{F}$ such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle, \qquad \forall x, x' \in \mathcal{X}, \tag{3.6}$$

where $\mathcal{F}$ can be infinite dimensional.

The connection to RKHS is interesting in at least two ways. First, it allows to use results and techniques from the theory of RKHS to analyze random features. Second, it shows that random features can be seen as an approach to derive scalable kernel methods [SS02]. Indeed, kernel methods have complexity at least quadratic in the number of points, while random features have complexity which is typically linear in the number of points. From this point of view, the intuition behind random features is to relax (3.6) considering

$$k(x, x') \approx \langle \phi_M(x), \phi_M(x') \rangle, \qquad \forall x, x' \in \mathcal{X}. \tag{3.7}$$

where ϕ_M is finite dimensional.

[1] For all $x_1, \ldots, x_n$ the matrix with entries $k(x_i, x_j)$, $i, j = 1, \ldots, n$ is positive semi-definite.

3.1.2 Computational Complexity

If we assume the computation of the feature map $\phi_M(x)$ to have a constant cost, the iteration (3.2) requires $O(M)$ operations per iteration for $b = 1$, that is $O(Mn)$ for one pass $t_{max} = n$. Note that for $b > 1$ each iteration cost $O(Mb)$ but one pass corresponds to $\lceil \frac{n}{b} \rceil$ iterations so that the cost for one pass is again $O(Mn)$. A main advantage of mini-batching is that gradient computations can be easily parallelized. In the multiple pass case, the time complexity after t_{max} iterations is $O(Mbt_{max})$.
Computing the feature map $\phi_M(x)$ requires to compute M random features. The computation of one random feature does not depend on n, but only on the input space $\mathcal{X}$. If for example we assume $\mathcal{X} \subseteq \mathbb{R}^d$ and consider random features defined as in the previous section, computing $\phi_M(x)$ requires M random projections of d dimensional vectors [RR08], for a total time complexity of $O(Md)$ for evaluating the feature map at one point. For different input spaces and different types of random features computational cost may differ, see for example Orthogonal Random Features [FSC$^+$16] or Fastfood [LSS13] where the cost is reduced from $O(Md)$ to $O(M \log d)$. Note that the analysis presented in his paper holds for random features which are independent, while Orthogonal and Fastfood random features are dependent. Although it should be possible to extend our analysis for Orthogonal and Fastfood random features, further work is needed. To simplify the discussion, in the following we treat the complexity of $\phi_M(x)$ to be $O(M)$.
One of the advantages of random features is that each $\phi_M(x)$ can be computed online at each iteration, preserving $O(Mbt_{max})$ as the time complexity of the algorithm (3.2). Computing $\phi_M(x)$ online also reduces memory requirements. Indeed the space complexity of the algorithm (3.2) is $O(Mb)$ if the mini-batches are computed in parallel, or $O(M)$ if computed sequentially.

3.1.3 Related Approaches

We comment on the connection to related algorithms. Random features are typically used within an empirical risk minimization framework [SC08]. Results considering convex Lipschitz loss functions and ℓ_∞ constraints are given in [RR09], while [Bac17] considers ℓ_2 constraints. A ridge regression framework is considered in [RR17], where it is shown that it is possible to achieve optimal statistical guarantees with a number of random features in the order of $\sqrt{n}$. The combination of random features and gradient methods is less explored. A stochastic coordinate descent approach is considered in [DXH$^+$14], see also [LR17a, TRVR16]. A related approach is based on subsampling and is often called Nyström method [SS00, WS01]. Here a shallow network is defined considering a nonlinearity which is a positive definite kernel, and weights chosen as a subset of training set points. This idea can be used within a penalized empirical risk minimization framework [RCR15, YPW15, AM15a] but also considering gradient [CARR16, RCR17] and stochastic gradient [LR17b] techniques. An empirical comparison between Nyström method, random features and full kernel method is given in [TRVR16], where the empirical risk minimization problem is solved by block coordinate descent. Note that numer-

ous works have combined stochastic gradient and kernel methods with no random projections approximation [DFB17, LR17c, PVRB18a, PVRB18b, RV15, Ora14]. The above list of references is only partial and focusing on papers providing theoretical analysis. In the following, after stating our main results we provide a further quantitative comparison with related results.

3.2 Main Results

In this section, we first discuss our main results under basic assumptions and then more refined results under further conditions.

3.2.1 Worst Case Results

Our results apply to a general class of random features described by the following assumption.

Assumption 7. *Let (Ω, π) be a probability space, $\psi : \mathcal{X} \times \Omega \to \mathbb{R}$ and for all $x \in \mathcal{X}$,*

$$\phi_M(x) = \frac{1}{\sqrt{M}} \left(\psi(x, \omega_1), \ldots, \psi(x, \omega_M)\right), \tag{3.8}$$

where $\omega_1, \ldots, \omega_M \in \Omega$ are sampled independently according to π.

The above class of random features cover all the examples described in Section 3.1.1, as well as many others, see [RR17, Bac17] and references therein. Next we introduce the positive definite kernel defined by the above random features. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be defined by

$$k(x, x') = \int \psi(x, \omega)\psi(x', \omega) d\pi(\omega), \quad \forall, x, x' \in \mathcal{X}.$$

It is easy to check that k is a symmetric and positive definite kernel. To control basic properties of the induced kernel (continuity, boundedness), we require the following assumption, which is again satisfied by the examples described in Section 3.1.1 (see also [RR17, Bac17] and references therein).

Assumption 8. *The function ψ is continuous and there exists $\kappa \geq 1$ such that $|\psi(x, \omega)| \leq \kappa$ for any $x \in \mathcal{X}, \omega \in \Omega$.*

The kernel introduced above allows to compare random feature maps of different size and to express the regularity of the largest function class they induce. In particular, we require Assumption 2 introduced in Section 2.5, the standard assumption on the existence of $f_{\mathcal{H}} \in \mathcal{H}$. In the end, we need an assumption on the data distribution. For all $x \in \mathcal{X}$, denote by $\rho(y|x)$ the conditional probability of ρ and by $\rho_{\mathcal{X}}$ the corresponding marginal probability on $\mathcal{X}$.

Assumption 9. *For any $x \in \mathcal{X}$*

$$\int_{\mathcal{Y}} y^{2l} d\rho(y|x) \leq l! B^l p, \qquad \forall l \in \mathbb{N} \tag{3.9}$$

for constants $B \in (0, \infty)$ and $p \in (1, \infty)$, $\rho_{\mathcal{X}}$-almost surely.

This assumption is slightly different form Assumption 3 presented in Section 2.5, but they both hold for bounded, sub-Gaussian or sub-exponential outputs y.

The next theorem corresponds to our first main result. In the following theorem, we control the *excess risk* of the estimator with respect to the number of points, the number of random features (RF), the step size, the mini-batch size and the number of iterations. We let $\widehat{f}_{t+1} = \langle \widehat{w}_{t+1}, \phi_M(\cdot) \rangle$, with $\widehat{w}_{t+1}$ as in (3.2). Denote with $[a] = \{1, \dots, a\}$ for any $a \in \mathbb{N}_+$, and with $b \wedge c$ and $b \vee c$ respectively the minimum and maximum between any $b, c \in \mathbb{R}$.

Theorem 1. *Let $n, M \in \mathbb{N}_+$, $\delta \in (0,1)$ and $t \in [T]$. Under Assumption 2,7,8,9, for $b \in [n]$, $\gamma_t = \gamma$ s.t. $\gamma \leq \frac{n}{9T \log \frac{n}{\delta}} \wedge \frac{1}{8(1+\log T)}$, $n \geq 32 \log^2 \frac{2}{\delta}$ and $M \gtrsim \gamma T$ the following holds with probability at least $1 - \delta$:*

$$\mathcal{E}_J\big[\mathcal{E}(\widehat{f}_{t+1})\big] - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{\gamma}{b} + \left(\frac{\gamma t}{M} + 1\right) \frac{\gamma t \log \frac{1}{\delta}}{n} + \frac{\log \frac{1}{\delta}}{M} + \frac{1}{\gamma t}. \tag{3.10}$$

The above theorem bounds the excess risk with a sum of terms controlled by the different parameters. The following corollary shows how these parameters can be chosen to derive finite sample bounds.

Corollary 5. *Under the same assumptions of Theorem 1, for one of the following conditions*

($c_{1.1}$). $b = 1$, $\gamma \simeq \frac{1}{n}$, and $T = n\sqrt{n}$ iterations ($\sqrt{n}$ passes over the data);

($c_{1.2}$). $b = 1$, $\gamma \simeq \frac{1}{\sqrt{n}}$, and $T = n$ iterations (1 pass over the data);

($c_{1.3}$). $b = \sqrt{n}$, $\gamma \simeq 1$, and $T = \sqrt{n}$ iterations (1 pass over the data);

($c_{1.4}$). $b = n$, $\gamma \simeq 1$, and $T = \sqrt{n}$ iterations ($\sqrt{n}$ passes over the data);

a number

$$M = \widetilde{O}(\sqrt{n}) \tag{3.11}$$

of random features is sufficient to guarantee with high probability that

$$\mathcal{E}_J\big[\mathcal{E}(\widehat{f}_T)\big] - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{1}{\sqrt{n}}. \tag{3.12}$$

The above learning rate is the same achieved by an exact kernel ridge regression (KRR) estimator [CDV07, SHS$^+$09, LRRC18], which has been proved to be optimal in a minimax sense [CDV07] under the same assumptions. Further, the number of random features required to achieve this bound is the same as the kernel ridge regression estimator with random features [RR17]. Notice that, for the limit case where the number of random features grows to infinity for Corollary 5 under conditions $(c_{1.2})$ and $(c_{1.3})$ we recover the same results for one pass SGD of [SZ13], [DGBSX12]. In this limit, our results are also related to those in [DB16]. Here, however, averaging of the iterates is used to achieve larger step-sizes.
Note that conditions $(c_{1.1})$ and $(c_{1.2})$ in the corollary above show that, when no mini-batches are used ($b = 1$) and $\frac{1}{n} \leq \gamma \leq \frac{1}{\sqrt{n}}$, then the step-size γ determines the number of passes over the data required for optimal generalization. In particular the number of passes varies from constant, when $\gamma = \frac{1}{\sqrt{n}}$, to $\sqrt{n}$, when $\gamma = \frac{1}{n}$. In order to increase the step-size over $\frac{1}{\sqrt{n}}$ the algorithm needs to be run with mini-batches. The step-size can then be increased up to a constant if b is chosen equal to $\sqrt{n}$ (condition $(c_{1.3})$), requiring the same number of passes over the data of the setting $(c_{1.2})$. Interestingly condition $(c_{1.4})$ shows that increasing the mini-batch size over $\sqrt{n}$ does not allow to take larger step-sizes, while it seems to increase the number of passes over the data required to reach optimality.
We now compare the time complexity of algorithm (3.2) with some closely related methods which achieve the same optimal rate of $\frac{1}{\sqrt{n}}$. As seen in Section 2.3, computing the classical KRR estimator (2.32) has a complexity of roughly $O(n^3)$ in time and $O(n^2)$ in memory. Lowering this computational cost is possible with random projection techniques. Both random features and Nyström method on KRR [RR17, RCR15] lower the time complexity to $O(n^2)$ and the memory complexity to $O(n\sqrt{n})$ preserving the statistical accuracy. The same time complexity is achieved by stochastic gradient method solving the full kernel method [LR17c, RV15], but with the higher space complexity of $O(n^2)$. The combination of the stochastic gradient iteration, random features and mini-batches allows our algorithm to achieve a complexity of $O(n\sqrt{n})$ in time and $O(n)$ in space for certain choices of the free parameters (like $(c_{1.2})$ and $(c_{1.3})$). Note that these time and memory complexity are lower with respect to those of stochastic gradient with mini-batches and Nyström approximation which are $O(n^2)$ and $O(n)$ respectively [LR17b]. We will present in the next chapter a method with similar complexity to stochastic gradient descent (SGD) with RF. This method, called FALKON, has indeed a time complexity of $O(n\sqrt{n}\log(n))$ and $O(n)$ space complexity. This method blends together Nyström approximation, a sketched preconditioner and conjugate gradient.

3.2.2 Refined Analysis and Fast Rates

We now discuss how faster rates can be achieved under the more refined assumptions discussed in Section 2.5.2.

The following theorem is a refined version of Theorem 1 where we also consider the assumptions

on the capacity and regularity condition.

Theorem 2. *Let $n, M \in \mathbb{N}_+$, $\delta \in (0,1)$ and $t \in [T]$, under Assumption 4,5,7,8,9, for $b \in [n]$, $\gamma_t = \gamma$ s.t. $\gamma \leq \frac{n}{9T\log\frac{n}{\delta}} \wedge \frac{1}{8(1+\log T)}$, $n \geq 32\log^2\frac{2}{\delta}$ and $M \gtrsim \gamma T$ the following holds with high probability:*

$$\mathcal{E}_J\big[\mathcal{E}(\widehat{f}_{t+1})\big] - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{\gamma}{b} + \left(\frac{\gamma t}{M} + 1\right)\frac{\mathcal{N}\left(\frac{1}{\gamma t}\right)\log\frac{1}{\delta}}{n} + \frac{\mathcal{N}\left(\frac{1}{\gamma t}\right)^{2r-1}\log\frac{1}{\delta}}{M(\gamma t)^{2r-1}} + \left(\frac{1}{\gamma t}\right)^{2r}. \tag{3.13}$$

The main difference is the presence of the effective dimension providing a sharper control of the stability of the considered estimator. As before, explicit learning bounds can be derived considering different parameter settings.

Corollary 6. *Under the same assumptions of Theorem 2, for one of the following conditions*

($c_{2.1}$). $b = 1$, $\gamma \simeq n^{-1}$, and $T = n^{\frac{2r+\alpha+1}{2r+\alpha}}$ iterations ($n^{\frac{1}{2r+\alpha}}$ passes over the data);

($c_{2.2}$). $b = 1$, $\gamma \simeq n^{-\frac{2r}{2r+\alpha}}$, and $T = n^{\frac{2r+1}{2r+\alpha}}$ iterations ($n^{\frac{1-\alpha}{2r+\alpha}}$ passes over the data);

($c_{2.3}$). $b = n^{\frac{2r}{2r+\alpha}}$, $\gamma \simeq 1$, and $T = n^{\frac{1}{2r+\alpha}}$ iterations ($n^{\frac{1-\alpha}{2r+\alpha}}$ passes over the data);

($c_{2.4}$). $b = n$, $\gamma \simeq 1$, and $T = n^{\frac{1}{2r+\alpha}}$ iterations ($n^{\frac{1}{2r+\alpha}}$ passes over the data);

a number

$$M = \widetilde{O}(n^{\frac{1+\alpha(2r-1)}{2r+\alpha}}) \tag{3.14}$$

of random features suffies to guarantee with high probability that

$$\mathcal{E}_J\big[\mathcal{E}(\widehat{w}_T)\big] - \mathcal{E}(f_{\mathcal{H}}) \lesssim n^{-\frac{2r}{2r+\alpha}}. \tag{3.15}$$

The corollary above shows that multi-pass SGD achieves a learning rate that is the same as kernel ridge regression under the regularity Assumption 5 and is again minimax optimal (see [CDV07]). Moreover, we obtain the minimax optimal rate with the same number of random features required for ridge regression with random features [RR17] under the same assumptions. Finally, when the number of random features goes to infinity we also recover the results for the infinite dimensional case of the single-pass and multiple pass stochastic gradient method [LR17c].
It is worth noting that, under the additional regularity Assumption 5, the number of both random features and passes over the data sufficient for optimal learning rates increase with respect to the one required in the worst case (see Corollary 5). The same effect occurs in the context of ridge regression with random features as noted in [RR17]. In this latter paper, it is observed that this issue tackled can be using more refined, possibly more costly, sampling schemes [Bac17].
Finally, we present a general result from which all our previous results follow as special cases. We consider a more general setting where we allow decreasing step-sizes.

Theorem 3. *Let $n, M, T \in \mathbb{N}$, $b \in [n]$ and $\gamma > 0$. Let $\delta \in (0,1)$ and $\widehat{w}_{t+1}$ be the estimator in Eq.* (3.2) *with $\gamma_t = \gamma\kappa^{-2}t^{-\theta}$ and $\theta \in [0,1[$. Under Assumption 4,5,7,8,9, when $n \geq 32\log^2\frac{2}{\delta}$ and*

$$\gamma \leq \frac{n}{9T^{1-\theta}\log\frac{n}{\delta}} \wedge \begin{cases} \frac{\theta\wedge(1-\theta)}{7} & \theta \in]0,1[\\ \frac{1}{8(1+\log T)} & \textit{otherwise}, \end{cases} \tag{3.16}$$

moreover

$$M \geq \left(4 + 18\gamma T^{1-\theta}\right)\log\frac{12\gamma T^{1-\theta}}{\delta}, \tag{3.17}$$

then, for any $t \in [T]$ the following holds with probability at least $1 - 9\delta$

$$\mathcal{E}_J\big[\mathcal{E}(\widehat{w}_{t+1})\big] - \inf_{w\in\mathcal{F}}\mathcal{E}(w) \leq c_1\,\frac{\gamma}{bt^{\min(\theta,1-\theta)}}(\log t \vee 1) \tag{3.18}$$

$$+ \left(c_2 + c_3\frac{1}{M}\log\frac{M}{\delta}\left(\gamma t^{1-\theta}\vee 1\right)\right)\frac{\mathcal{N}\left(\frac{\kappa^2}{\gamma t^{1-\theta}}\right)}{n}\left(\log^2(t)\vee 1\right)\log^2\frac{4}{\delta} \tag{3.19}$$

$$+ c_4\left(\frac{\mathcal{N}(\frac{\kappa^2}{\gamma t^{1-\theta}})^{2r-1}\log\frac{2}{\delta}}{M(\gamma t^{1-\theta}\kappa^{-2})^{2r-1}}\,\log^{2-2r}\left(11\gamma t^{1-\theta}\right) + \left(\frac{1}{\gamma t^{1-\theta}}\right)^{2r}\right), \tag{3.20}$$

with c_1, c_2, c_3, c_4 constants which do not depend on $b, \gamma, n, t, M, \delta$.

We note that as the number of random features M goes to infinity, we recover the same bound of [LR17c] for decreasing step-sizes. Moreover, the above theorem shows that there is no apparent gain in using a decreasing stepsize (i.e. $\theta > 0$) with respect to the regimes identified in Corollaries 5 and 6.

3.2.3 Sketch of the Proof

In this section, we sketch the main ideas in the proof. We relate $\widehat{f}_t$ and $f_{\mathcal{H}}$ introducing several intermediate functions. In particular, the following iterations are useful,

$$\widehat{v}_1 = 0; \qquad \widehat{v}_{t+1} = \widehat{v}_t - \gamma_t\frac{1}{n}\sum_{i=1}^{n}\big(\langle\widehat{v}_t, \phi_M(x_i)\rangle - y_i\big)\phi_M(x_i), \qquad \forall t \in [T]. \tag{3.21}$$

$$\widetilde{v}_1 = 0; \qquad \widetilde{v}_{t+1} = \widetilde{v}_t - \gamma_t\int_{\mathcal{X}}\big(\langle\widetilde{v}_t, \phi_M(x)\rangle - y\big)\phi_M(x)d\rho(x,y), \qquad \forall t \in [T]. \tag{3.22}$$

$$v_1 = 0; \qquad v_{t+1} = v_t - \gamma_t\int_{\mathcal{X}}\big(\langle v_t, \phi_M(x)\rangle - f_{\mathcal{H}}(x)\big)\phi_M(x)d\rho_{\mathcal{X}}(x), \qquad \forall t \in [T]. \tag{3.23}$$

Further, we let

$$\widetilde{u}_\lambda = \operatorname*{argmin}_{u\in\mathbb{R}^M} \int_{\mathcal{X}} \big(\langle u, \phi_M(x)\rangle - f_{\mathcal{H}}(x)\big)^2 d\rho_{\mathcal{X}}(x) + \lambda\|u\|^2, \quad \lambda > 0, \tag{3.24}$$

$$u_\lambda = \operatorname*{argmin}_{u\in\mathcal{F}} \int_{\mathcal{X}} \big(\langle u, \phi(x)\rangle - y\big)^2 d\rho(x,y) + \lambda\|u\|^2, \quad \lambda > 0, \tag{3.25}$$

where $(\mathcal{F}, \phi)$ are feature space and feature map associated to the kernel k. The first three vectors are defined by the random features and can be seen as an empirical and population batch gradient descent iterations. The last two vectors can be seen as a population version of ridge regression defined by the random features and the feature map ϕ, respectively.
Since the above objects (3.21), (3.22), (3.23), (3.24), (3.25) belong to different spaces, instead of comparing them directly we compare the functions in $L^2(\mathcal{X}, \rho_{\mathcal{X}})$ associated to them, letting

$$\widehat{g}_t = \langle \widehat{v}_t, \phi_M(\cdot)\rangle\,, \;\; \widetilde{g}_t = \langle \widetilde{v}_t, \phi_M(\cdot)\rangle\,, \;\; g_t = \langle v_t, \phi_M(\cdot)\rangle\,, \;\; \widetilde{g}_\lambda = \langle \widetilde{u}_\lambda, \phi_M(\cdot)\rangle\,, \;\; g_\lambda = \langle u_\lambda, \phi(\cdot)\rangle\,.$$

Since it is well known [CDV07] that

$$\mathcal{E}(f) - \mathcal{E}(f_{\mathcal{H}}) = \|f - f_{\mathcal{H}}\|_\rho^2,$$

we than can consider the following decomposition

$$\begin{aligned}
\widehat{f}_t - f_{\mathcal{H}} &= \widehat{f}_t - \widehat{g}_t && (3.26)\\
&+ \widehat{g}_t - \widetilde{g}_t && (3.27)\\
&+ \widetilde{g}_t - g_t && (3.28)\\
&+ g_t - \widetilde{g}_\lambda && (3.29)\\
&+ \widetilde{g}_\lambda - g_\lambda && (3.30)\\
&+ g_\lambda - f_{\mathcal{H}}. && (3.31)
\end{aligned}$$

The first two terms control how SGD deviates from the batch gradient descent and the effect of noise and sampling. They are studied in Lemma 1, 2, 3, 4, 5, 6 in the following Section, borrowing and adapting ideas from [LR17c, RV15, RR17]. The following terms account for the approximation properties of random features and the bias of the algorithm. Here the basic idea and novel result is the study of how the population gradient decent and ridge regression are related (3.29) (Lemma 9 in Section 3.3). Then, results from the the analysis of ridge regression with random features are used [RR17].

3.3 Details of the Proof

We start recalling some definitions and define some new operators.

3.3.1 Preliminary Definitions

Let $\mathcal{F}$ be the feature space corresponding to the kernel k given in Assumption 8.

Given $\phi\colon \mathcal{X} \to \mathcal{F}$ (feature map), we define the operator $S\colon \mathcal{F} \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$ as

$$(Sw)(\cdot) = \langle w, \phi(\cdot)\rangle_{\mathcal{F}}, \qquad \forall w \in \mathcal{F}. \tag{3.32}$$

If S^* is the adjoint operator of S, we let $C\colon \mathcal{F} \to \mathcal{F}$ be the linear operator $C = S^*S$, which can be written as

$$C = \int_{\mathcal{X}} \phi(x) \otimes \phi(x) d\rho_{\mathcal{X}}(x), \tag{3.33}$$

where we denote with $\otimes$ the tensor product, in particular

$$(u \otimes v)z = u \langle v, z\rangle_{\mathcal{F}}, \quad \forall u, v, z \in \mathcal{F}.$$

We also define the linear operator $L\colon L^2(\mathcal{X}, \rho_{\mathcal{X}}) \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$ such that $L = SS^*$, that can be represented as

$$(Lf)(\cdot) = \int_{\mathcal{X}} \langle \phi(x), \phi(\cdot)\rangle_{\mathcal{F}}\, f(x) d\rho_{\mathcal{X}}(x), \qquad \forall f \in L^2(\mathcal{X}, \rho_{\mathcal{X}}). \tag{3.34}$$

We now define the analog of the previous operators where we use the feature map ϕ_M instead of ϕ. We have $S_M\colon \mathbb{R}^M \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$ defined as

$$(S_M v)(\cdot) = \langle v, \phi_M(\cdot)\rangle_{\mathbb{R}^M}, \qquad \forall v \in \mathbb{R}^M, \tag{3.35}$$

together with $C_M\colon \mathbb{R}^M \to \mathbb{R}^M$ and $L_M\colon L^2(\mathcal{X}, \rho_{\mathcal{X}}) \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$ defined as $C_M = S_M^* S_M$ and $L_M = S_M S_M^*$ respectively.

We also define the empirical counterpart of the previous operators. The operator $\widehat{S}_M\colon \mathbb{R}^M \to \mathbb{R}^n$ is defined as,

$$\widehat{S}_M^* = \frac{1}{\sqrt{n}} \left(\phi_M(x_1), \ldots, \phi_M(x_n)\right), \tag{3.36}$$

and with $\widehat{C}_M\colon \mathbb{R}^M \to \mathbb{R}^M$ and $\widehat{L}_M\colon \mathbb{R}^n \to \mathbb{R}^n$ are defined as $\widehat{C}_M = \widehat{S}_M^* \widehat{S}_M$ and $\widehat{L}_M = \widehat{S}_M \widehat{S}_M^*$, respectively. We further denote with $A_\lambda = A + \lambda I$, for any linear operator A and $\lambda \in \mathbb{R}$.

Remark 1 (from [CS02, VRC$^+$05])**.** *Let $P : L^2(\mathcal{X}, \rho_{\mathcal{X}}) \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$ be the projection operator whose range is the closure of the range of L. Let $f_\rho : \mathcal{X} \to \mathbb{R}$ be defined as*

$$f_\rho(x) = \int_{\mathcal{Y}} y d\rho(y|x).$$

If there exists $f_{\mathcal{H}} \in \mathcal{H}$ such that

$$\inf_{f \in \mathcal{H}} \mathcal{E}(f) = \mathcal{E}(f_{\mathcal{H}}),$$

then

$$Pf_\rho = Sf_{\mathcal{H}},$$

or equivalently, there exists $g \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$ *such that*

$$Pf_\rho = L^{\frac{1}{2}} g$$

In particular, we have $R := \|f_{\mathcal{H}}\|_{\mathcal{H}} = \|g\|_{L^2(\mathcal{X},\rho_{\mathcal{X}})}$. *The above condition is commonly relaxed in approximation theory as*

$$Pf_\rho = L^r g,$$

with $\frac{1}{2} \leq r \leq 1$ *[SZ03].*

With the operators introduced above and Remark 1, we can rewrite the auxiliary objects (3.21), (3.22), (3.23), (3.24), (3.25) respectively as

$$\widehat{v}_1 = 0; \qquad \widehat{v}_{t+1} = (I - \gamma_t \widehat{C}_M)\widehat{v}_t + \gamma_t \widehat{S}_M^* \widehat{y}, \qquad \forall t \in [T], \tag{3.37}$$

$$\widetilde{v}_1 = 0; \qquad \widetilde{v}_{t+1} = (I - \gamma_t C_M)\widetilde{v}_t + \gamma_t S_M^* f_\rho, \qquad \forall t \in [T], \tag{3.38}$$

$$v_1 = 0; \qquad v_{t+1} = (I - \gamma_t C_M)v_t + \gamma_t S_M^* P f_\rho, \qquad \forall t \in [T]. \tag{3.39}$$

where $\widehat{y} = n^{-1/2}(y_1, \ldots, y_n)$, and

$$\widetilde{u}_\lambda = S_M^* L_{M,\lambda}^{-1} P f_\rho, \tag{3.40}$$

$$u_\lambda = S^* L_\lambda^{-1} P f_\rho. \tag{3.41}$$

By a simple induction argument the three sequences can be written as

$$\widehat{v}_{t+1} = \textstyle\sum_{i=1}^t \gamma_i \prod_{k=i+1}^t (I - \gamma_k \widehat{C}_M)\widehat{S}_M^* \widehat{y}, \tag{3.42}$$

$$\widetilde{v}_{t+1} = \textstyle\sum_{i=1}^t \gamma_i \prod_{k=i+1}^t (I - \gamma_k \widehat{C}_M) S_M^* f_\rho, \tag{3.43}$$

$$v_{t+1} = \textstyle\sum_{i=1}^t \gamma_i \prod_{k=i+1}^t (I - \gamma_k C_M) S_M^* P f_\rho. \tag{3.44}$$

3.3.2 Error Decomposition

We can now rewrite the error decomposition of $\widehat{f}_t - f_{\mathcal{H}}$ using the operators introduced above as

$$\begin{aligned}
S_M \widehat{w}_t - P f_\rho &= S_M \widehat{w}_t - S_M \widehat{v}_t && (3.45)\\
&+ S_M \widehat{v}_t - S_M \widetilde{v}_t && (3.46)\\
&+ S_M \widetilde{v}_t - S_M v_t && (3.47)\\
&+ S_M v_t - L_M L_{M,\lambda}^{-1} P f_\rho && (3.48)\\
&+ L_M L_{M,\lambda}^{-1} P f_\rho - L L_\lambda^{-1} P f_\rho && (3.49)\\
&+ L L_\lambda^{-1} P f_\rho - P f_\rho. && (3.50)
\end{aligned}$$

3.3.3 Lemmas

The first three lemmas we present are some technical lemmas used when bounding the first three terms (3.45), (3.46), (3.47) of the error decomposition.

Lemma 1. *Under Assumption 8 the following holds for any* $t, M, n \in \mathbb{N}$

$$\|\widetilde{v}_t - v_t\| = 0 \quad a.s. \tag{3.51}$$

Proof. Given (3.43), (3.44) and defining $A_{Mt} = \sum_{i=1}^{t} \gamma_i \prod_{k=i+1}^{t} (I - \gamma_k C_M)$, we can write

$$\|\widetilde{v}_t - v_t\| = \|A_{Mt} S_M^* (I - P) f_\rho\| \leq \|A_{Mt}\| \, \|S_M^* (I - P)\| \, \|f_\rho\| \,. \tag{3.52}$$

Under Assumption 8, by Lemma 2 of [RR17], we have $\|S_M^*(I - P)\| = 0$, which completes the proof. □

Lemma 2. *Let* $M \in \mathbb{N}$. *Under Assumption 8 and 5, let* $\gamma_t \kappa^2 \leq 1$, $\delta \in]0, 1]$, *the following holds with probability* $1 - \delta$ *for all* $t \in [T]$

$$\|\widetilde{v}_{t+1}\| \leq 2R\kappa^{2r-1} \left(1 + \sqrt{\frac{9\kappa^2}{M} \log \frac{M}{\delta}} \max \left(\left(\sum_{i=1}^{t} \gamma_t \right)^{\frac{1}{2}}, \kappa^{-1} \right) \right) . \tag{3.53}$$

Proof. Considering (3.38) (3.39), we can write

$$\|\widetilde{v}_{t+1}\| \leq \|\widetilde{v}_{t+1} - v_{t+1}\| + \|v_{t+1}\| = \|v_{t+1}\|, \tag{3.54}$$

where in the last equality we used the result from Lemma 1. Using Assumption 5 (see also Remark 1), we derive

$$\|v_{t+1}\| = \left\| \sum_{i=1}^{t} \gamma_i S_M^* \prod_{k=i+1}^{t} (I - \gamma_k L_M) P f_\rho \right\| \leq R \left\| \sum_{i=1}^{t} \gamma_i S_M^* \prod_{k=i+1}^{t} (I - \gamma_k L_M) L^r \right\| \tag{3.55}$$

Define $Q_{Mt} = \sum_{i=1}^{t} \gamma_i S_M^* \prod_{k=i+1}^{t} (I - \gamma_k L_M)$. Note that $\|L^{r-\frac{1}{2}}\| \leq \kappa^{2r-1}$ for $r \geq \frac{1}{2}$ and that $\|L_{M,\eta}^{-1/2} L^{1/2}\| \leq 2$ holds with probability $1 - \delta$ when $\frac{9\kappa^2}{M} \log \frac{M}{\delta} \leq \eta \leq \|L\|$ (see Lemma 5 in [RCR15]). Moreover, when $\eta \geq \|L\|$, we have that $\|L_{M,\eta}^{-1/2} L^{1/2}\| \leq \eta^{-1/2} \|L^{1/2}\| \leq 1$. So $\|L_{M,\eta}^{-1/2} L^{1/2}\| \leq 2$ with probability $1 - \delta$, when

$$\frac{9\kappa^2}{M} \log \frac{M}{\delta} \leq \eta. \tag{3.56}$$

So when (3.56) holds, with probability $1-\delta$ we can write

$$\begin{aligned} R\|Q_{Mt}L^r\| &= R\|Q_{Mt}L_{M,\eta}^{\frac{1}{2}}L_{M,\eta}^{-\frac{1}{2}}L^{\frac{1}{2}}L^{r-\frac{1}{2}}\| \\ &\leq R\|Q_{Mt}L_{M,\eta}^{\frac{1}{2}}\|\|L_{M,\eta}^{-\frac{1}{2}}L^{\frac{1}{2}}\|\|L^{r-\frac{1}{2}}\| \\ &\leq 2R\kappa^{2r-1}\|Q_{Mt}L_{M,\eta}^{\frac{1}{2}}\| \\ &\leq 2R\kappa^{2r-1}\left(\|Q_{Mt}L_M^{\frac{1}{2}}\| + \eta^{\frac{1}{2}}\|Q_{Mt}\|\right). \end{aligned} \tag{3.57}$$

Now note that for any $a \in [0, 1/2]$,

$$\|Q_{Mt}L_M^a\| \leq \max\left(\kappa^{2a-1}, \left(\sum_{i=1}^t \gamma_i\right)^{\frac{1}{2}-a}\right) \tag{3.58}$$

(see Lemma B.10(i) in [RV15] or Lemma 16 of [LR17c]). We use (3.58) with $a = \frac{1}{2}$ and $a = 0$ to bound $\|Q_{Mt}L_M^{1/2}\|$ and $\|Q_{Mt}\|$ respectively and plug the results in (3.57). To complete the proof we take $\eta = \frac{9\kappa^2}{M}\log\frac{M}{\delta}$. □

Lemma 3. *Let $\lambda > 0$, $R \in \mathbb{N}$ and $\delta \in (0,1)$. Let $\zeta_1, \ldots, \zeta_R$ be i.i.d. random vectors bounded by $\kappa > 0$. Denote with $Q_R = \frac{1}{R}\sum_{j=1}^R \zeta_j \otimes \zeta_j$ and by Q the expectation of Q_R. Then, for any $\lambda \geq \frac{9\kappa^2}{R}\log\frac{R}{\delta}$, we have*

$$\|(Q_R + \lambda I)^{-1/2}(Q + \lambda I)^{1/2}\|^2 \leq 2.$$

Proof. This lemma is a more refined version of a result in [RCR13]. When $\|Q\| \geq \lambda \geq \frac{9\kappa^2}{R}\log\frac{R}{\delta}$, by combining Prop. 8 of [RR17], with Prop. 6 and in particular Rem. 10 point 2 of the same paper, we have that $\|(Q_R+\lambda I)^{-1/2}(Q+\lambda I)^{1/2}\| \leq 2$, with probability at least $1-\delta$. To cover the case $\lambda > \|Q\|$, note that

$$\|(Q_R + \lambda I)^{-1/2}(Q + \lambda I)^{1/2}\| \leq (\|Q\|^{1/2} + \lambda^{1/2})/\lambda^{1/2}.$$

When $\lambda > \|Q\|$, we have that

$$\|(Q_R + \lambda I)^{-1/2}(Q + \lambda I)^{1/2}\| \leq \sup_{\lambda > \|Q\|} (\|Q\|^{1/2} + \lambda^{1/2})/\lambda^{1/2} \leq 2.$$

□

We need the following technical lemma that complements Proposition 10 of [RR17] when $\lambda \geq \|L\|$, and that we will need for the proof of Lemma 6.

Lemma 4. *Let $M \in \mathbb{N}$ and $\delta \in (0,1]$. For any $\lambda > 0$ such that*

$$M \geq \left(4 + \frac{18\kappa^2}{\lambda}\right) \log \frac{12\kappa^2}{\lambda\delta},$$

the following holds with probability $1-\delta$

$$\mathcal{N}_M(\lambda) := \int_{\mathcal{X}} \|(L_M + \lambda I)^{-\frac{1}{2}} \phi_M(x)\|^2 d\rho_{\mathcal{X}}(x) \leq \max\left(2.55, \frac{2\kappa^2}{\|L\|}\right) \mathcal{N}(\lambda).$$

Proof. First of all note that

$$\mathcal{N}_M(\lambda) := \int_{\mathcal{X}} \|(L_M + \lambda I)^{-\frac{1}{2}} \phi_M(x)\|^2 d\rho_{\mathcal{X}}(x) = \mathrm{Tr}(L_{M,\lambda}^{-\frac{1}{2}} L_M L_{M,\lambda}^{-\frac{1}{2}}) = \mathrm{Tr}(L_{M,\lambda}^{-1} L_M).$$

Now consider the case when $\lambda \leq \|L\|$. By applying Proposition 10 of [RR17] we have that under the required condition on M, the following holds with probability at least $1-\delta$

$$\mathcal{N}_M(\lambda) \leq 2.55 \mathcal{N}(\lambda).$$

For the case $\lambda > \|L\|$, note that $\mathrm{Tr}(AA_\lambda^{-1})$ satisfies the following inequality for any trace class positive linear operator A with trace bounded by κ^2 and $\lambda > 0$,

$$\frac{\|A\|}{\|A\| + \lambda} \leq \mathrm{Tr}(AA_\lambda^{-1}) \leq \frac{\mathrm{Tr}(A)}{\lambda}.$$

So, when $\lambda > \|L\|$, since $\mathcal{N}_M(\lambda) = \mathrm{Tr}(C_M C_{M\lambda}^{-1})$ and $\mathcal{N}(\lambda) = \mathrm{Tr}(LL_\lambda^{-1})$, and both L and $\widehat{C}_M$ have trace bounded by κ^2, we have $\mathcal{N}_M(\lambda) \leq \frac{\kappa^2}{\lambda}$ and $\mathcal{N}(\lambda) \geq \frac{\|L\|}{\|L\|+\lambda}$. So by selecting $q = \frac{\kappa^2(\|L\|+\lambda)}{\lambda\|L\|}$, we have

$$\mathcal{N}_M(\lambda) \leq \frac{\kappa^2}{\lambda} = q\frac{\|L\|}{\|L\| + \lambda} \leq q\mathcal{N}(\lambda).$$

Finally note that

$$q \leq \sup_{\lambda > \|L\|} \frac{\kappa^2(\|L\| + \lambda)}{\lambda\|L\|} \leq 2\frac{\kappa^2}{\|L\|}.$$

□

We now start bounding the different parts of the error decomposition. The next two lemmas bound the first two terms (3.45), (3.46). To bound these we require the above lemmas and adapting ideas from [LR17c, RV15, RR17].

Lemma 5. *Under Assumption 8 and 9, let* $\delta \in]0,1[$, $n \geq 32 \log^2 \frac{2}{\delta}$, *and* $\gamma_t = \gamma \kappa^{-2} t^{-\theta}$ *for all* $t \in [T]$, *with* $\theta \in [0,1[$ *and* γ *such that*

$$0 < \gamma \leq \frac{t^{\min(\theta, 1-\theta)}}{8(\log t + 1)}, \qquad \forall t \in [T]. \tag{3.59}$$

When

$$\frac{1}{\gamma t^{1-\theta}} \geq \frac{9}{n} \log \frac{n}{\delta} \tag{3.60}$$

for all $t \in [T]$, *with probability at least* $1 - 2\delta$,

$$\mathcal{E}_{\mathbf{J}} \|S_M(\widehat{w}_{t+1} - \widehat{v}_{t+1})\|^2 \leq \frac{208 Bp}{(1-\theta)b} \left(\gamma t^{-\min(\theta, 1-\theta)}\right) (\log t \vee 1). \tag{3.61}$$

Proof. The proof is derived by applying Proposition 6 in [LR17c] with γ satisfying condition (3.59), $\lambda = \frac{1}{\gamma_t t}$, $\delta_2 = \delta_3 = \delta$, and some changes that we now describe. Instead of the stochastic iteration w_t and the batch gradient iteration ν_t as defined in [LR17c] we consider (3.2) and (3.37) respectively, as well as the operators S_M, C_M, L_M, $\widehat{S}_M, \widehat{C}_M, \widehat{L}_M$ defined in Section 2 instead of $S_\rho, \mathcal{T}_\rho, \mathcal{L}_\rho$, $S_{\mathbf{x}}, \mathcal{T}_{\mathbf{x}}, \mathcal{L}_{\mathbf{x}}$ defined in [LR17c]. Instead of assuming that exists a $\kappa \geq 1$ for which $\langle x, x' \rangle \leq \kappa^2, \forall x, x' \in \mathcal{X}$ we have Assumption 8 which implies the same κ^2 upper bound of the operators used in the proof. To apply this version of Proposition 6 note that their Equation (63) is satisfied by Lemma 25 of [LR17c], while their Equation (47) is satisfied by our Lemma 3, from which we obtain the condition (3.60). □

Lemma 6. *Under Assumptions 8, 9 and 5, let* $\delta \in]0,1[$ *and* $\gamma_t = \gamma \kappa^{-2} t^{-\theta}$ *for all* $t \in [T]$, *with* $\gamma \in]0,1]$ *and* $\theta \in [0,1[$. *When*

$$M \geq \left(4 + 18\gamma t^{1-\theta}\right) \log \frac{12\gamma t^{1-\theta}}{\delta}, \tag{3.62}$$

for all $t \in [T]$ *with probability at least* $1 - 3\delta$

$$\begin{aligned} \|S_M(\widehat{v}_{t+1} - \widetilde{v}_{t+1})\| \leq 4 \left(R\kappa^{2r} \left(1 + \sqrt{\frac{9}{M} \log \frac{M}{\delta}} \left(\sqrt{\gamma t^{1-\theta}} \vee 1\right)\right) + \sqrt{B} \right) \times \\ \times \left(\frac{8}{(1-\theta)} + 4 \log t + 4 + \sqrt{2}\gamma\right) \left(\frac{\sqrt{\gamma t^{1-\theta}}}{n} + \frac{\sqrt{2\sqrt{p} q_0 \mathcal{N}(\frac{\kappa^2}{\gamma t^{1-\theta}})}}{\sqrt{n}} \right) \log \frac{4}{\delta}, \end{aligned} \tag{3.63}$$

where $q_0 = \max\left(2.55, \frac{2\kappa^2}{\|L\|}\right)$.

Proof. The proof can be derived from the one of Theorem 5 in [LR17c] with $\lambda = \frac{1}{\gamma_t t}$, $\delta_1 = \delta_2 = \delta$, and some changes we now describe. Instead of the iteration ν_t and μ_t defined in [LR17c]

we consider (3.37) and (3.38) respectively, as well as the operators S_M, C_M, L_M, $\widehat{S}_M, \widehat{C}_M, \widehat{L}_M$ defined in Section 2 instead of $S_\rho, \mathcal{T}_\rho, \mathcal{L}_\rho$, $S_{\mathbf{x}}, \mathcal{T}_{\mathbf{x}}, \mathcal{L}_{\mathbf{x}}$ defined in [LR17c]. Instead of assuming that exists a $\kappa \geq 1$ for which $\langle x, x' \rangle \leq \kappa^2, \forall x, x' \in \mathcal{X}$ we have Assumption 8 which imply the same $\|C_M\| \leq \kappa^2$ upper bound of the operators used in the proof. Further, when in the proof we need to bound $\|v_{t+1}\|$ we use our Lemma 2 instead of Lemma 16 of [LR17c]. In addition instead of Lemma 18 of [LR17c] we use Lemma 6 of [RR17], together with Lemma 4, obtaining the desired result with probability $1 - 3\delta$, when M satisfies $M \geq (4 + 18\gamma_t t) \log \frac{12\gamma_t t}{\delta}$. Under the assumption that $\gamma_t = \gamma \kappa^{-2} t^{-\theta}$, the two condition above can be rewritten as (3.62). □

The next lemma states that the third term (3.47) of the error decomposition is equal to zero.

Lemma 7. *Under Assumption 5 the following holds for any* $t, M, n \in \mathbb{N}$

$$\|S_M \widetilde{v}_t - S_M v_t\| = 0 \quad a.s. \tag{3.64}$$

Proof. From Lemma 1 and the definition of operator norm the result follows trivially. □

The next Lemma is a known result from Lemma 8 of [RR17] which bounds the distance between the Tikhonov solution with RF and the Tikhonov solution without RF (3.49).

Lemma 8. *Under Assumption 8 and 5 for any* $\lambda > 0$, $\delta \in (0, 1/2]$, *when*

$$M \geq \left(4 + \frac{18\kappa^2}{\lambda}\right) \log \frac{8\kappa^2}{\lambda\delta} \tag{3.65}$$

the following holds with probability at least $1 - 2\delta$,

$$\|L L_\lambda^{-1} P f_\rho - L_M L_{M,\lambda}^{-1} P f_\rho\| \leq 4R\kappa^{2r} \left(\frac{\log \frac{2}{\delta}}{M^r} + \sqrt{\frac{\lambda^{2r-1} \mathcal{N}(\lambda)^{2r-1} \log \frac{2}{\delta}}{M}} \right) q^{1-r}, \tag{3.66}$$

where $q := \log \frac{11\kappa^2}{\lambda}$.

The next lemma is one of our main contributions and studies how the population gradient decent with RF and ridge regression with RF are related (3.48).

Lemma 9. *Under Assumption 5 the following holds with probability* $1 - \delta$ *for* $\lambda = \frac{1}{\sum_{i=1}^t \gamma_i}$ *for all* $t \in [T]$

$$\begin{aligned} \|S_M v_{t+1} - L_M L_{M,\lambda}^{-1} P f_\rho\|_\rho \leq 8R\kappa^{2r} & \left(\frac{\log \frac{2}{\delta}}{M^r} + \sqrt{\frac{\mathcal{N}((\sum_{i=1}^t \gamma_i)^{-1})^{2r-1} \log \frac{2}{\delta}}{M(\sum_{i=1}^t \gamma_i)^{2r-1}}} \right) \times \\ & \times \log^{1-r} \left(11\kappa^2 \sum_{i=1}^t \gamma_i \right) + 2R \left(\sum_{i=1}^t \gamma_i \right)^{-r}, \end{aligned} \tag{3.67}$$

when

$$M \geq \left(4 + 18\sum_{i=1}^{t}\gamma_i\right)\log\left(\frac{8\kappa^2\sum_{i=1}^{t}\gamma_i}{\delta}\right). \tag{3.68}$$

Proof. Denoting $Q_M = \sum_{i=1}^{t}\gamma_i\prod_{k=i+1}^{t}(I-\gamma_k L_M)$ we can write

$$S_M v_{t+1} = Q_M L_M P f_\rho$$

Then

$$\begin{aligned} S_M v_{t+1} - L_M L_{M,\lambda}^{-1} P f_\rho &= Q_M L_{M,\lambda} L_M L_{M,\lambda}^{-1} - L_M L_{M,\lambda}^{-1} P f_\rho \\ &= (Q_M(L_M+\lambda I) - I) L_M L_{M,\lambda}^{-1} P f_\rho. \end{aligned} \tag{3.69}$$

Denote by $A_{i,t}$ the operator $A_{i,t} := \prod_{k=i}^{t}(I-\gamma_k L_M)$, and note that

$$A_{i,t} := (I-\gamma_k L_M)A_{i+1,t}.$$

We can then derive

$$\begin{aligned} Q_M L_M &= \sum_{i=1}^{t}\gamma_i\prod_{k=i+1}^{t}(I-\gamma_k L_M)L_M = \sum_{i=1}^{t}(I-(I-\gamma_i L_M))\prod_{k=i+1}^{t}(I-\gamma_k L_M) \\ &= \sum_{i=1}^{t}(I-(I-\gamma_i L_M))A_{i+1,t} = \sum_{i=1}^{t}A_{i+1,t} - \sum_{i=1}^{t}(I-\gamma_i L_M)A_{i+1,t} \\ &= \sum_{i=1}^{t}A_{i+1,t} - \sum_{i=1}^{t}A_{i,t} = I + \sum_{i=2}^{t}A_{i,t} - \sum_{i=1}^{t}A_{i,t} = I - A_{1,t}. \end{aligned}$$

We now write

$$\begin{aligned} \|(Q_M(L_M+\lambda I)-I)L_M\| &= \|(Q_M L_M + \lambda Q_M - I)L_M\| \\ &= \|(I - A_{1,t} + \lambda Q_M - I)L_M\| \\ &= \|\lambda Q_M L_M - A_{1,t}L_M\| \\ &\leq \|\lambda Q_M L_M\| + \|A_{1,t}L_M\|. \end{aligned} \tag{3.70}$$

For the first term in (3.70) we have

$$\|\lambda Q_M L_M\| = \lambda\|I - A_{1,t}\| \leq \lambda,$$

since L_M is positive operator and $\gamma_i\|L_M\| < 1$, so $A_{1,t}$ is positive with norm smaller than one by construction, implying that $\|I - A_{1,t}\| \leq 1$. The second term in (3.70) can be bounded using Lemma 15 in [LR17c],

$$\|A_{1,t}L_M\| \leq (\sum_{i=1}^{t}\gamma_i)^{-1}$$

Now back to (3.69), we can write

$$\|S_M v_{t+1} - L_M L_{M,\lambda}^{-1} P f_\rho\|_\rho \leq \left(\lambda + \frac{1}{\sum_{i=1}^t \gamma_i}\right) \|L_{M\lambda}^{-1} P f_\rho\|_\rho. \tag{3.71}$$

Setting $\lambda = \frac{1}{\sum_{i=1}^t \gamma_i}$, and calling this quantity $\widetilde{\lambda}$ for the rest of the proof, we can write

$$\begin{aligned}
\|S_M v_{t+1} - L_M L_{M,\widetilde{\lambda}}^{-1} P f_\rho\|_\rho &\leq 2\widetilde{\lambda} \|L_{M\widetilde{\lambda}}^{-1} P f_\rho\|_\rho & (3.72)\\
&= 2\|(\widetilde{\lambda} L_{M\widetilde{\lambda}}^{-1} - \widetilde{\lambda} L_{\widetilde{\lambda}}^{-1} + \widetilde{\lambda} L_{\widetilde{\lambda}}^{-1}) P f_\rho\|_\rho & (3.73)\\
&\leq 2\|(\widetilde{\lambda} L_{M\widetilde{\lambda}}^{-1} - \widetilde{\lambda} L_{\widetilde{\lambda}}^{-1}) P f_\rho\|_\rho + 2\widetilde{\lambda} \|L_{\widetilde{\lambda}}^{-1} P f_\rho\|_\rho. & (3.74)
\end{aligned}$$

Since $A A_\lambda^{-1} = I - \lambda A_\lambda^{-1}$ for any bounded symmetric operator A and $\lambda > 0$, we can write the last term of (3.74) as

$$\widetilde{\lambda} \|L_{\widetilde{\lambda}}^{-1} P f_\rho\|_\rho = \|(L L_{\widetilde{\lambda}}^{-1} - I) P f_\rho\|_\rho.$$

We can then use Lemma 10 to control this quantity as

$$\|(L L_{\widetilde{\lambda}}^{-1} - I) P f_\rho\|_\rho \leq R \widetilde{\lambda}^r. \tag{3.75}$$

For the first term, analogously

$$\begin{aligned}
\|(\widetilde{\lambda} L_{M\widetilde{\lambda}}^{-1} - \widetilde{\lambda} L_{\widetilde{\lambda}}^{-1}) P f_\rho\|_\rho &= \|((I - \widetilde{\lambda} L_{M\widetilde{\lambda}}^{-1}) - (I - \widetilde{\lambda} L_\lambda^{-1})) P f_\rho\|_\rho \\
&= \|(L_M L_{M\widetilde{\lambda}}^{-1} - L L_{\widetilde{\lambda}}^{-1}) P f_\rho\|_\rho \\
&\leq 4R\kappa^{2r} \left(\frac{\log \frac{2}{\delta}}{M^r} + \sqrt{\frac{\widetilde{\lambda}^{2r-1} \mathcal{N}(\widetilde{\lambda})^{2r-1} \log \frac{2}{\delta}}{M}} \right) \left(\log \frac{11\kappa^2}{\widetilde{\lambda}} \right)^{1-r}, \qquad (3.76)
\end{aligned}$$

where the last step holds when $M \geq (4 + 18\widetilde{\lambda}^{-1}) \log(8\kappa^2 (\widetilde{\lambda}\delta)^{-1})$ and consists in the application of Lemma 9. Now recalling the definition of $\widetilde{\lambda}$ we complete the proof. □

The last result is a classical bound of the approximation error for the Tikhonov filter (3.50), see [CDV07].

Lemma 10 (From [CDV07] or Lemma 5 of [RR17])**.** *Under Assumption 5*

$$\|L L_\lambda^{-1} P f_\rho - P f_\rho\| \leq R \lambda^r \tag{3.77}$$

3.3.4 Proofs of Theorems

We now present the proofs of our theorems. Theorem 2 and 1 are specific case of the more general Theorem 3.

Proof of Theorem 3. We start considering Lemma 6, and we note that condition (3.62) is satisfied when

$$M \geq \left(4 + 18\gamma T^{1-\theta}\right) \log \frac{12\gamma T^{1-\theta}}{\delta}. \tag{3.78}$$

Noting that (3.16) imply $\sqrt{2}\gamma \leq 1$, we can derive from (3.63)

$$\begin{aligned} \|S_M(\widehat{v}_{t+1} - v_{t+1})\|^2 \leq & \left(\frac{(17-9\theta)\sqrt{8\sqrt{p}}}{(1-\theta)}\right)^2 \times \\ & \times \left(32B + 64R^2\kappa^{4r}\left(1 + \frac{9}{M}\log\frac{M}{\delta}\left(\gamma t^{1-\theta} \vee 1\right)\right)\right) \times \\ & \times \frac{q_0 \mathcal{N}(\frac{\kappa^2}{\gamma t^{1-\theta}})}{n}\left(\log^2 t \vee 1\right)\log^2\frac{4}{\delta}, \end{aligned} \tag{3.79}$$

when (3.78) holds.

Let $\lambda = \frac{\kappa^2}{\gamma t^{1-\theta}}$. Given Lemma 8 we derive from (3.66) that

$$\begin{aligned} \left\|LL_\lambda^{-1}Pf_\rho - L_M L_{M,\lambda}^{-1} P f_\rho\right\|^2 \leq & 32R^2\kappa^{4r}\left(\frac{\log^2\frac{2}{\delta}}{M^{2r}} + \frac{\mathcal{N}(\frac{\kappa^2}{\gamma t^{1-\theta}})^{2r-1}\log\frac{2}{\delta}}{M(\gamma t^{1-\theta}\kappa^{-2})^{2r-1}}\right) \times \\ & \times \log^{2-2r}\left(11\gamma t^{1-\theta}\right), \end{aligned} \tag{3.80}$$

when (3.78) holds.

Let $\gamma_t = \gamma\kappa^{-2}t^{-\theta}$ for all $t \in [T]$. Given Lemma 9 we derive from (3.67)

$$\begin{aligned} \left\|S_M v_{t+1} - L_M L_{M,\lambda}^{-1} P f_\rho\right\|^2 \leq & 8R^2\kappa^{4r}\left(32\left(\frac{\log^2\frac{2}{\delta}}{M^{2r}} + \frac{\mathcal{N}(\frac{\kappa^2}{\gamma t^{1-\theta}})^{2r-1}\log\frac{2}{\delta}}{M(\gamma t^{1-\theta}\kappa^{-2})^{2r-1}}\right) \times \right. \\ & \left. \times \log^{2-2r}\left(11\gamma t^{1-\theta}\right) + \left(\frac{1}{\gamma t^{1-\theta}}\right)^{2r}\right), \end{aligned} \tag{3.81}$$

when (3.78) holds.

Similarly from Lemma 10

$$\|LL_\lambda^{-1}Pf_\rho - Pf_\rho\|^2 \leq R^2\kappa^{4r}\left(\frac{1}{\gamma t^{1-\theta}}\right)^{2r}. \tag{3.82}$$

The desired result is obtained by gathering the results in (3.61), (3.79), (3.81), (3.80), (3.82). Requiring γ, M to satisfy the associated conditions (3.78), (3.59), (3.60). In particular note that

(3.59) is satisfied when $\theta = 0$ by $\gamma \leq (8(\log T + 1))^{-1}$, while, if $\theta > 0$, we have

$$\begin{aligned}\frac{t^{\min(\theta,1-\theta)}}{8(\log t + 1)} &= e^{-\min(\theta,1-\theta)} \frac{(et)^{\min(\theta,1-\theta)}}{8\log(et)} \geq e^{-\min(\theta,1-\theta)} \inf_{t\in 1} \frac{(et)^{\min(\theta,1-\theta)}}{8\log(et)} \\ &= e^{-\min(\theta,1-\theta)} \inf_{z\geq e^{\min(\theta,1-\theta)}} \frac{z}{\frac{8}{\min(\theta,1-\theta)}\log z} \\ &\geq e^{-\min(\theta,1-\theta)} \inf_{z\geq 1} \frac{z}{\frac{8}{\min(\theta,1-\theta)}\log z} \geq e^{-\min(\theta,1-\theta)} \frac{\min(\theta,1-\theta)}{4},\end{aligned}$$

where we performed the change of variable $t^{\min(\theta,1-\theta)} = z$. Finally note that $e^{-\min(\theta,1-\theta)} \geq e^{-1/2}$, for any $\theta \in (0,1)$. Moreover the (3.78), (3.60) are satisfied for any $t \in [T]$ by requiring them to hold for $t = T$. □

Proof of Theorem 2. Choosing $\theta = 0$ in Theorem 3 we complete the proof. □

Proof of Theorem 1. Considering the case of Assumption 4 with $\alpha = 1$ and Assumption 5 with $r = \frac{1}{2}$, we can bound $\mathcal{N}(1/\gamma t) \leq \gamma t$ in Theorem 3 and complete the proof. □

3.4 Experiments

We study the behavior of the SGD with RF algorithm on subsets of $n = 2 \times 10^5$ points of the SUSY [2] and HIGGS [3] datasets [BSW14]. The measures we show in the following experiments are an average over 10 repetitions of the algorithm. Further, we consider random Fourier features that are known to approximate translation invariant kernels [RR08]. We use random features of the form $\psi(x,\omega) = \cos(w^T x + q)$, with $\omega := (w,q)$, w sampled according to the normal distribution and q sampled uniformly at random between 0 and 2π. Note that the random features defined this way satisfy Assumption 8.

Our theoretical analysis suggests that only a number of RF of the order of $\sqrt{n}$ suffices to gain optimal learning properties. Hence we study how the number of RF affect the accuracy of the algorithm on test sets of 10^5 points. In Figure 3.1 we show the classification error after 5 passes over the data of SGD with RF as the number of RF increases, with a fixed batch size of $\sqrt{n}$ and a step-size of 1. We can observe that over a certain threshold of the order of $\sqrt{n}$, increasing the number of RF does not improve the accuracy, confirming what our theoretical results suggest.

Further, theory suggests that the step-size can be increased as the mini-batch size increases to reach an optimal accuracy, and that after a mini-batch size of the order of $\sqrt{n}$ more than 1 pass over the data is required to reach the same accuracy. We show in Figure 3.2 the classification

[2] `https://archive.ics.uci.edu/ml/datasets/SUSY`

[3] `https://archive.ics.uci.edu/ml/datasets/HIGGS`

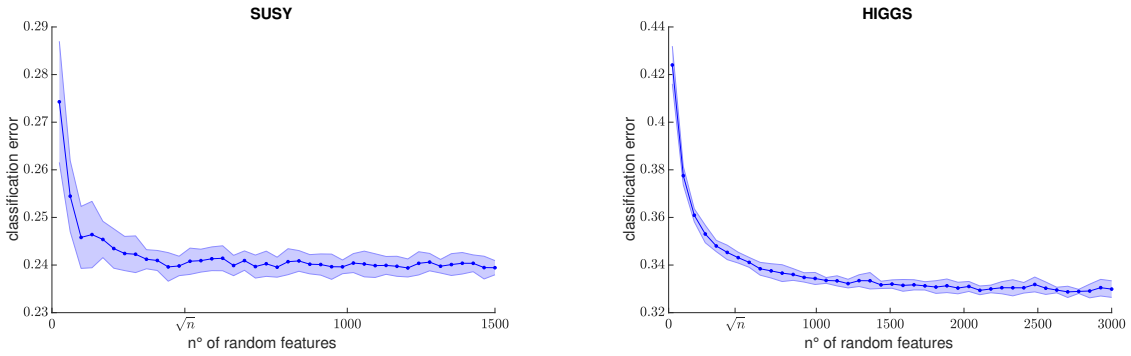

Fig. 3.1: Classification error of SUSY (left) and HIGGS (right) datasets as the number of random features (M) varies

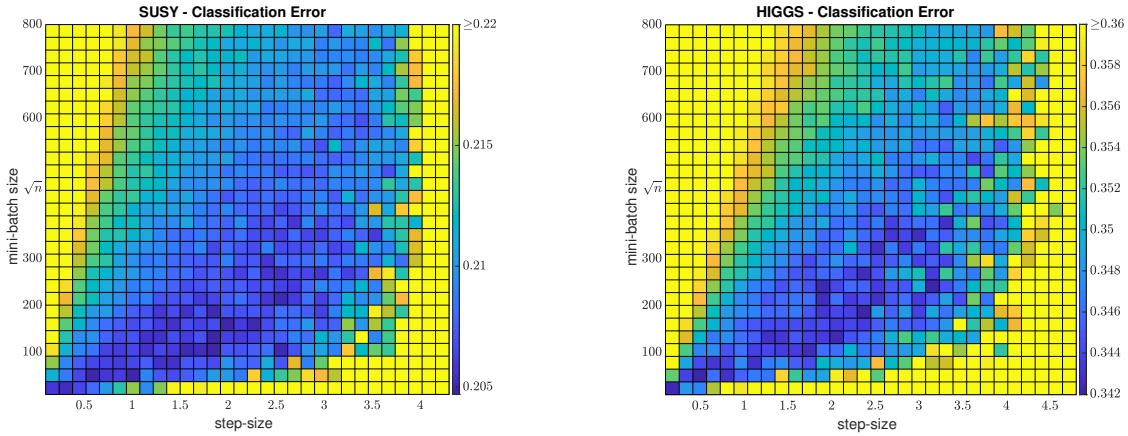

Fig. 3.2: Classification error of SUSY (left) and HIGGS (right) datasets as step-size and mini-batch size vary

error of SGD with RF after 1 pass over the data, with a fixed number of random features $\sqrt{n}$, as mini-batch size and step-size vary, on test sets of 10^5 points. As suggested by theory, to reach the lowest error as the mini-batch size grows the step-size needs to grow as well. Further for mini-batch sizes bigger that $\sqrt{n}$ the lowest error can not be reached in only 1 pass even if increasing the step-size.

Chapter 4

FALKON

In this chapter, we propose and study FALKON, a new algorithm that provides an efficient approach to apply kernel methods on millions of points, and tested on a variety of large scale problems outperforms previously proposed methods while utilizing only a fraction of computational resources.

The state of the art approximation of KRR, for which optimal statistical bounds are known, typically requires complexities that are roughly $\mathcal{O}(n^2)$ in time and memory (or possibly $\mathcal{O}(n)$ in memory, if kernel computations are made on the fly). The new FALKON algorithm is derived combining several algorithmic principles, namely stochastic subsampling, iterative solvers and preconditioning. In particular, it exploits the idea of using Nyström methods [SS00] to approximate the KRR problem, but also to efficiently compute a preconditioner to be used in conjugate gradient. Our theoretical analysis shows that optimal statistical accuracy is achieved requiring essentially $\mathcal{O}(n)$ memory and $\mathcal{O}(n\sqrt{n})$ time. An extensive experimental analysis on large scale datasets shows that, even with a single machine, FALKON outperforms the previous state of the art solutions, which exploit parallel/distributed architectures.

4.1 From Kernel Ridge Regression to Nyström Approximation

We consider the supervised learning problem of estimating a function from random noisy samples introduced in Chapter 2. We are interested in both computational and statistical aspects of this problem. In particular, we investigate the computational resources needed to achieve optimal statistical accuracy, i.e. minimal excess risk. Our focus is on the most popular class of nonparametric methods, namely kernel methods.

Recall for Section 2.3 that Kernel methods consider a space $\mathcal{H}$ of functions

$$f(x) = \sum_{i=1}^{n} k(x, x_i)c_j, \tag{4.1}$$

where k is a positive definite kernel. The coefficients $c = (c_1, \ldots, c_n)$ are typically derived from a convex optimization problem, that for the square loss is

$$\widehat{f}_\lambda = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \left\| f \right\|_{\mathcal{H}}^2, \tag{4.2}$$

and defines the so called kernel ridge regression (KRR) estimator [SS02]. An advantage of least squares approaches is that they reduce computations to a linear system

$$(\widehat{K} + \lambda n I)\, c = \widehat{y}, \tag{4.3}$$

where $\widehat{K}$ is an $n \times n$ matrix defined by $(\widehat{K})_{ij} = K(x_i, x_j)$ and $\widehat{y} = (y_1, \ldots y_n)$. A direct approach to solve (4.3) requires $\mathcal{O}(n^2)$ in space, $\mathcal{O}(n^3)$ in time and $O(n^2)$ in kernel evaluations.

As we have seen in Section 2.5, under basic assumptions, KRR achieves an error $\mathcal{R}(\widehat{f}_{\lambda_n}) = O(n^{-1/2})$, for $\lambda_n = n^{-1/2}$, which is optimal in a minimax sense and can be improved *only* under more stringent assumptions [CDV07, SHS$^+$09]. The question is then if it is possible to achieve the statistical properties of KRR, with less computations.

A natural idea introduced in Section 2.4 is to consider iterative solvers and in particular gradient methods, because of their simplicity and low iteration cost. In the case of the iterative solver described in equation (2.35), if t is the number of iterations, this method requires $O(n^2 t)$ in time, $O(n^2)$ in memory and $O(n^2)$ in kernel evaluations, if the kernel matrix is stored. Note that, the kernel matrix can also be computed on the fly with only $O(n)$ memory, but $O(n^2 t)$ kernel evaluations are required.

We note that, beyond this simple iteration, several variants have been considered including accelerated [CY10, BPR07] and stochastic extensions [DB16].

While the time complexity of these methods dramatically improves over KRR, and computations can be done in blocks, memory requirements (or number of kernel evaluations) still makes the application to large scale setting cumbersome. Randomization provides an approach to tackle this challenge.

4.1.1 Random Projections.

The rough idea is to use random projections to compute $\widehat{K}$ only approximately. The most popular examples in this class of approaches are the random features methods introduced in Section 3.1.1

and the Nyström [SS00] methods. In the following we focus in particular on a basic Nyström approach based on considering functions of the form

$$\widetilde{f}_{\lambda,M}(x) = \sum_{i=1}^{M} k(x, \widetilde{x}_i)\widetilde{c}_i, \quad \text{with} \quad \{\widetilde{x}_1, \dots, \widetilde{x}_M\} \subseteq \{x_1, \dots, x_n\}, \tag{4.4}$$

defined considering only a subset of M training points sampled uniformly. In this case, there are only M coefficients that, following the approach in (4.2), can be derived considering the linear system

$$H\widetilde{c} \;=\; z, \qquad \text{where} \quad H = K_{nM}^\top K_{nM} \;+\; \lambda n K_{MM}, \quad z = K_{nM}^\top \widehat{y}. \tag{4.5}$$

Here $\widehat{K}_{nM}$ is the $n \times M$ matrix with $(\widehat{K}_{nM})_{ij} = K(x_i, \widetilde{x}_j)$ and $\widehat{K}_{MM}$ is the $M \times M$ matrix with $(\widehat{K}_{MM})_{ij} = K(\widetilde{x}_i, \widetilde{x}_j)$. This method consists in subsampling the columns of $\widehat{K}$ and can be seen as a particular form of random projections.

Direct methods for solving (4.5) require $O(nM^2)$ in time to form $\widehat{K}_{nM}^\top \widehat{K}_{nM}$ and $O(M^3)$ for solving the linear system, and only $O(nM)$ kernel evaluations. The naive memory requirement is $O(nM)$ to store $\widehat{K}_{nM}$, however if $\widehat{K}_{nM}^\top \widehat{K}_{nM}$ is computed in blocks of dimension at most $M \times M$ only $O(M^2)$ memory is needed. Iterative approaches can also be combined with random projections [DXH+14, CARR16, TRVR16] to slightly reduce time requirements (see Table 4.1, or Section 4.4, for more details).

The key point though, is that random projections allow to dramatically reduce memory requirements as soon as $M \ll n$ and the question arises of whether this comes at expenses of statistical accuracy. Interestingly, recent results considering this question show that there are large classes of problems for which $M = \widetilde{O}(\sqrt{n})$ suffices for the same optimal statistical accuracy of the exact KRR [Bac13, AM15a, RCR15].

In summary, in this case the computations needed for optimal statistical accuracy are reduced from $O(n^2)$ to $O(n\sqrt{n})$ kernel evaluations, but the best time complexity is basically $O(n^2)$. In the rest of the chapter we discuss how this requirement can indeed be dramatically reduced.

4.2 FALKON

Our approach is based on a novel combination of randomized projections with iterative solvers plus preconditioning. The main novelty is that we use random projections to approximate both the problem *and* the preconditioning.

4.2.1 Preliminaries: Preconditioning and KRR

We begin recalling the basic idea behind preconditioning. The key quantity is the condition number, that for a linear system is the ratio between the largest and smallest singular values of the matrix defining the problem [Saa03]. For example, for problem (4.3) the condition number is given by

$$\text{cond}(\widehat{K} + \lambda nI) = (\sigma_{\text{max}} + \lambda n)/(\sigma_{\text{min}} + \lambda n),$$

with $\sigma_{\text{max}}, \sigma_{\text{min}}$ largest and smallest eigenvalues of $\widehat{K}$, respectively. The importance of the condition number is that it captures the time complexity of iteratively solving the corresponding linear system. For example, if a simple gradient descent is used, the number of iterations needed for an ε accurate solution of problem (4.3) is

$$t = O(\text{cond}(\widehat{K} + \lambda nI)\log(1/\varepsilon)).$$

It is shown in [CARR16] that in this case $t = \sqrt{n}\log n$ are needed to achieve a solution with good statistical properties. Indeed, it can be shown that roughly $t \approx 1/\lambda \log(\frac{1}{\varepsilon})$ are needed where $\lambda = 1/\sqrt{n}$ and $\varepsilon = 1/n$. The idea behind preconditioning is to use a suitable matrix B to define an equivalent linear system with better condition number. For (4.3), an ideal choice is B such that

$$BB^\top = (\widehat{K} + \lambda nI)^{-1} \tag{4.6}$$

and $B^\top(\widehat{K} + \lambda nI)B\ \beta\ =\ B^\top\widehat{y}$. Clearly, if β_* solves the latter problem, $\alpha_* = B\beta_*$ is a solution of problem (4.3). Using a preconditioner B as in (4.6) one iteration is sufficient, but computing the B is typically as hard as the original problem. The problem is to derive preconditioning such that (4.6) might hold only approximately, but that can be computed efficiently. Derivation of efficient preconditioners for the exact KRR problem (4.3) has been the subject of recent studies, [FM12, ACW16, COCF16, GOSS16, MB17]. In particular, [ACW16, COCF16, GOSS16, MB17] consider random projections to approximately compute a preconditioner. Clearly, while preconditioning (4.3) leads to computational speed ups in terms of the number of iterations, requirements in terms of memory/kernel evaluation are the same as standard kernel ridge regression.

The key idea to tackle this problem is to consider an efficient preconditioning approach for problem (4.5) rather than (4.3).

4.2.2 Basic FALKON Algorithm

We begin illustrating a basic version of our approach. The key ingredient is the following preconditioner for Eq. (4.5),

$$BB^\top = \Big(\frac{n}{M}\widehat{K}_{MM}^2 + \lambda n\widehat{K}_{MM}\Big)^{-1}, \tag{4.7}$$

which is itself based on a Nyström approximation[1]. The above preconditioning is a natural approximation of the ideal preconditioning of problem (4.5) that corresponds to

$$BB^\top = (K_{nM}^\top K_{nM} \ + \ \lambda n K_{MM})^{-1}$$

and reduces to it if $M = n$. Our theoretical analysis, shows that $M \ll n$ suffices for deriving optimal statistical rates. In its basic form FALKON is derived combining the above preconditioning and gradient descent,

$$\widehat{f}_{\lambda,M,t}(x) = \sum_{i=1}^{M} k(x, \widetilde{x}_i) c_{t,i}, \quad \text{with} \quad c_t = B\beta_t \quad \text{and} \tag{4.8}$$

$$\beta_s = \beta_{s-1} - \frac{\gamma}{n} B^\top \left[\widehat{K}_{nM}^\top (\widehat{K}_{nM}(B\beta_{s-1}) - \widehat{y}) \ + \ \lambda n \widehat{K}_{MM}(B\beta_{s-1}) \right], \tag{4.9}$$

for $t \in \mathbb{N}$, $\beta_0 = 0$ and $1 \leq s \leq t$ and a suitable chosen γ. In practice, a refined version of FALKON is preferable where a faster gradient iteration is used and additional care is taken in organizing computations.

4.2.3 The Complete Algorithm

The actual version of FALKON we propose is Alg. 1 (see Sect. 4.5, Alg. 2 for the complete algorithm). It consists in solving the system $B^\top H B \beta = B^\top z$ via conjugate gradient [Saa03], since it is a fast gradient method and does not require to specify the step-size. Moreover, to compute B quickly, with reduced numerical errors, we consider the following strategy

$$B = \frac{1}{\sqrt{n}} T^{-1} R^{-1}, \qquad T = \text{chol}(K_{MM}), \quad R = \text{chol}\left(\frac{1}{M} \, T \, T^\top + \lambda I \right), \tag{4.10}$$

where chol() is the Cholesky decomposition (in Sect. 4.5 the strategy for non invertible K_{MM}).

Computations. in Alg. 1, B is never built explicitly and R, T are two upper-triangular matrices, so $R^{-\top} u$, $R^{-1} u$ for a vector u costs M^2, and the same for T. The cost of computing the preconditioner is only $\frac{4}{3} M^3$ floating point operations (consisting in two Cholesky decompositions and one product of two triangular matrices). Then FALKON requires $O(nMt + M^3)$ in time and the same $O(M^2)$ memory requirement of the basic Nyström method, if matrix/vector multiplications at each iteration are performed in blocks. This implies $O(nMt)$ kernel evaluations are needed.

The question remains to characterize M and the number of iterations needed for good statistical accuracy. Indeed, in the next section we show that roughly $O(n\sqrt{n})$ computations and $O(n)$ memory are sufficient for optimal accuracy. This implies that FALKON is currently the most efficient kernel method with the same optimal statistical accuracy of KRR, see Table 4.1.

[1] For the sake of simplicity, here we assume $\widehat{K}_{MM}$ to be invertible and the Nyström centers selected with uniform sampling from the training set, see Sect. 4.5 and Alg. 2 in the appendix for the general algorithm.

Algorithm 1: MATLAB code for FALKON. It requires $O(nMt + M^3)$ in time and $O(M^2)$ in memory. See Sect. 4.5 and Alg. 2 in the appendixes for the complete algorithm.

Input: Dataset $X = (x_i)_{i=1}^n \in \mathbb{R}^{n\times d}$, $\widehat{y} = (y_i)_{i=1}^n \in \mathbb{R}^n$, centers $Cen = (\widetilde{x}_j)_{j=1}^M \in \mathbb{R}^{M\times d}$, KernelMatrix computing the kernel matrix given two sets of points, regularization parameter λ, number of iterations t.
Output: Nyström coefficients c.

```
function c = FALKON(X, Cen, Y, KernelMatrix, lambda, t)
n = size(X,1); M = size(Cen,1); KMM = KernelMatrix(Cen,Cen);
T = chol(KMM + eps*M*eye(M));
R = chol(T*T'/M + lambda*eye(M));

function w = KnM_times_vector(u, v)
w = zeros(M,1); ms = ceil(linspace(0, n, ceil(n/M)+1));
for i=1:ceil(n/M)
Kr = KernelMatrix( X(ms(i)+1:ms(i+1),:), Cen );
w = w + Kr'*(Kr*u + v(ms(i)+1:ms(i+1),:));
end
end

BHB = @(u) R'\(T'\(KnM_times_vector(T\(R\u), zeros(n,1))/n)+lambda*(R\u));
r = R'\(T'\KnM_times_vector(zeros(M,1), Y/n));
c = T\(R\conjgrad(BHB, r, t));
end
```

4.3 Theoretical Analysis

In this section, we characterize the generalization properties of FALKON showing it achieves the optimal generalization error of KRR, with dramatically reduced computations. This result is given in Theorem 6 and derived in two steps. First, we study the difference between the excess risk of FALKON and that of the basic Nyström (4.5), showing it depends on the condition number induced by the preconditioning, hence on M (see Theorem 4). Deriving these results requires some care, since differently to standard optimization results, our goal is to solve (2.4) i.e. achieve small excess risk, not to minimize the empirical error. Second, we show that choosing $M = \widetilde{O}(1/\lambda)$ allows to make this difference as small as $e^{-t/2}$ (see Theorem 5). Finally, recalling that the basic Nyström for $\lambda = 1/\sqrt{n}$ has essentially the same statistical properties of KRR [RCR15], we answer the question posed at the end of the last section and show that roughly $\log n$ iterations are sufficient for optimal statistical accuracy. Following the discussion in the previous section this means that the computational requirements for optimal accuracy are $\widetilde{O}(n\sqrt{n})$ in time/kernel evaluations and $\widetilde{O}(n)$ in space. Later in this section faster rates under further regularity assumptions are also derived and the effect of different selection methods for the Nyström centers considered. The proofs for this section are provided in Sect. 4.9 of the appendixes.

4.3.1 Main Result

The first result is interesting in its own right since it corresponds to translating optimization guarantees into statistical results. In particular, we derive a relation the excess risk of the FALKON algorithm $\widehat{f}_{\lambda,M,t}$ from Algorithm 1 and the Nyström estimator $\widetilde{f}_{\lambda,M}$ from Eq. (4.5) with uniform sampling.

Theorem 4. *Let $n, M \geq 3$, $t \in \mathbb{N}$, $0 < \lambda \leq \lambda_1$ and $\delta \in (0, 1]$. Under Assumption 1, the following inequality holds with probability $1 - \delta$*

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \;\leq\; \mathcal{R}(\widetilde{f}_{\lambda,M})^{1/2} \;+\; 4\widehat{v}\, e^{-\nu t} \sqrt{1 + \frac{9\kappa^2}{\lambda n} \log \frac{n}{\delta}},$$

where $\widehat{v}^2 = \frac{1}{n}\sum_{i=1}^{n} y_i^2$ and $\nu = \log(1 + 2/(\operatorname{cond}(B^\top H B)^{1/2} - 1))$, with $\operatorname{cond}(B^\top H B)$ the condition number of $B^\top H B$. Note that $\lambda_1 > 0$ is a constant not depending on λ, n, M, δ, t.

The additive term in the bound above decreases exponentially in the number of iterations. If the condition number of $B^\top H B$ is smaller than a small universal constant (e.g. 17), then $\nu > 1/2$ and the additive term decreases as $e^{-\frac{t}{2}}$. Next, theorems derive a condition on M that allows to control $\operatorname{cond}(B^\top H B)$, and derive such an exponential decay.

Theorem 5. *Under the same conditions of Thm. 4, if*

$$M \geq 5 \left[1 + \frac{14\kappa^2}{\lambda}\right] \log \frac{8\kappa^2}{\lambda\delta}.$$

then the exponent ν in Thm. 4 satisfies $\nu \geq 1/2$.

The above result gives the desired exponential bound showing that after $\log n$ iterations the excess risk of FALKON is controlled by that of the basic Nyström, more precisely

$$\mathcal{R}(\widehat{f}_{\lambda,M,t}) \leq 2\mathcal{R}(\widetilde{f}_{\lambda,M}) \quad \text{when} \quad t \geq \log \mathcal{R}(\widetilde{f}_{\lambda,M}) \;+\; \log\left(1 + \frac{9\kappa^2}{\lambda n} \log \frac{n}{\delta}\right) \;+\; \log\left(16\widehat{v}^2\right).$$

Finally, we derive an excess risk bound for FALKON. By the no-free-lunch theorem, this requires some conditions on the learning problem. We first consider the standard basic setting introduced in Section 2.5.1 where we only assume it exists $f_{\mathcal{H}} \in \mathcal{H}$ such that $\mathcal{E}(f_{\mathcal{H}}) = \inf_{f \in \mathcal{H}} \mathcal{E}(f)$. We also make a stronger assumption on y being bounded which we will relax in following theorems.

Theorem 6. *Let $\delta \in (0, 1]$. Under Assumption 1 and assuming $y \in [-\frac{a}{2}, \frac{a}{2}]$, almost surely, for $a > 0$, then there exist $n_0 \in \mathbb{N}$ such that for any $n \geq n_0$, if*

$$\lambda = \frac{1}{\sqrt{n}}, \qquad M \geq 75\,\sqrt{n}\,\log\frac{48\kappa^2 n}{\delta}, \qquad t \;\geq\; \frac{1}{2}\log(n) \;+\; 5 + 2\log(a + 3\kappa),$$

Algorithm	train time	kernel evaluations	memory	test time
SVM / KRR + direct method	n^3	n^2	n^2	n
KRR + iterative [CY10, GRO+08]	$n^2\sqrt[4]{n}$	n^2	n^2	n
Doubly stochastic [DXH+14]	$n^2\sqrt{n}$	$n^2\sqrt{n}$	n	n
Pegasos / KRR + sgd [SSSSC11]	n^2	n^2	n	n
KRR + iter + precond [FM12, YPW15, ACW16, GOSS16, MB17]	n^2	n^2	n	n
Divide & Conquer [ZDW13]	n^2	$n\sqrt{n}$	n	n
Nyström, random features [WS01, SS00, RR08]	n^2	$n\sqrt{n}$	n	$\sqrt{n}$
Nyström + iterative [CARR16, TRVR16]	n^2	$n\sqrt{n}$	n	$\sqrt{n}$
Nyström + sgd [LR17b]	n^2	$n\sqrt{n}$	n	$\sqrt{n}$
FALKON (see Thm. 6)	$\boldsymbol{n\sqrt{n}}$	$\boldsymbol{n\sqrt{n}}$	$\boldsymbol{n}$	$\boldsymbol{\sqrt{n}}$

Table 4.1: Computational complexity required by different algorithms, for optimal generalization. Logarithmic terms are not showed.

then with probability $1-\delta$*,*

$$\mathcal{R}(\widehat{f}_{\lambda,M,t}) \leq \frac{c_0 \log^2 \frac{24}{\delta}}{\sqrt{n}}.$$

In particular n_0, c_0 *do not depend on* λ, M, n, t *and* c_0 *do not depend on* δ*.*

The above result provides the desired bound, and all the constants are given in Section 4.9. The obtained learning rate is the same as the full KRR estimator and is known to be optimal in a minmax sense [CDV07], hence not improvable. As mentioned before, the same bound is also achieved by the basic Nyström method but with much worse time complexity. Indeed, as discussed before, using a simple iterative solver typically requires $O(\sqrt{n}\log n)$ iterations, while we need only $O(\log n)$. Considering the choice for M this leads to a computational time of $O(nMt) = O(n\sqrt{n})$ for optimal generalization (omitting logarithmic terms). To the best of our knowledge FALKON currently provides the best time/space complexity to achieve the statistical accuracy of KRR.

Beyond the basic setting considered above, in the next section we show that FALKON can achieve much faster rates under refined regularity assumptions and also consider the potential benefits of leverage score sampling.

4.3.2 Fast Learning Rates and Nyström with Approximate Leverage Scores

Considering fast rates and Nyström with more general sampling is considerably more technical and a heavier notation is needed. Our analysis apply to any approximation scheme (e.g. [DMIMW12, AM15a, CLM+15]) satisfying the definition of q-approximate leverage scores [RCR15].

We recall the definition of approximate leverage scores, and then the sampling method based

on them. Let $n \in \mathbb{N}$, $\lambda > 0$. Let $x_1, \ldots, x_n$ be the training points and define $\widehat{K} \in \mathbb{R}^{n \times n}$ as $(\widehat{K})_{ij} = K(x_i, x_j)$ for $1 \leq i, j \leq n$. The exact leverage scores are defined by

$$\ell(i, \lambda) = \left(\widehat{K}(\widehat{K} + \lambda n I)^{-1}\right)_{ii}, \tag{4.11}$$

for any $i \in 1, \ldots, n$. Any bi-Lipschitz approximation of the exact leverage scores, satisfying the following definition is denoted as approximate leverage scores.

Definition 2 ((q, λ_0, δ)-approximate leverage scores [RCR15])**.** *Let $\delta \in (0, 1]$ and $\lambda_0 > 0$ and $q \in [1, \infty)$. A (random) sequence $(\widetilde{\ell}(i, \lambda))_{i=1}^n$ is denoted as (q, λ_0, δ)-approximate leverage scores when the following holds with probability at least $1 - \delta$*

$$\frac{1}{q}\, \ell(i, \lambda) \;\leq\; \widetilde{\ell}(i, \lambda) \;\leq\; q\, \ell(i, \lambda), \qquad \forall \lambda \geq \lambda_0, t \in \{1, \ldots, n\}.$$

In particular, given $n \in \mathbb{N}$ training points $x_1, \ldots, x_n$, and a sequence of approximate leverage scores $(\widetilde{\ell}(i, \lambda))_{i=1}^n$, the Nyström centers $\widetilde{x}_1, \ldots, \widetilde{x}_M$ are selected in the following way. Let

$$p_i = \frac{\widetilde{\ell}(i, \lambda)}{\sum_{j=1}^n \widetilde{\ell}(j, \lambda)},$$

with $1 \leq i \leq n$. Let $i_1, \ldots, i_M$ be independently sampled from $\{1, \ldots, n\}$ with probability $(p_i)_{i=1}^n$. Then $\widetilde{x}_1 := x_{i_1}, \ldots, \widetilde{x}_M := x_{i_M}$.

We need a few more definitions to state the next theorem for fast rates. Let $k_x = k(x, \cdot)$ for any $x \in \mathcal{X}$ and $\mathcal{H}$ the reproducing kernel Hilbert space [SC08] of functions with inner product defined by $\mathcal{H} = \overline{\operatorname{span}\{k_x \mid x \in \mathcal{X}\}}$ and closed with respect to the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ defined by $\langle k_x, k_{x'} \rangle_{\mathcal{H}} = k(x, x')$, for all $x, x' \in \mathcal{X}$. Define $C : \mathcal{H} \to \mathcal{H}$ to be the linear operator

$$\langle f, Cg \rangle_{\mathcal{H}} = \int_{\mathcal{X}} f(x) g(x) d\rho_{\mathcal{X}}(x),$$

for all $f, g \in \mathcal{H}$. Finally, for any $\lambda > 0$, we recall the definition of *effective dimension* and define a new quantity,

$$\mathcal{N}(\lambda) = \operatorname{Tr}\left(C(C + \lambda I)^{-1}\right), \tag{4.12}$$

$$\mathcal{N}_\infty(\lambda) = \sup_{x \in \mathcal{X}} \left\| (C + \lambda I)^{-1/2} k_x \right\|_{\mathcal{H}}. \tag{4.13}$$

Note that the effective dimension defined in (4.12) is the same as the one defined in (2.49) thanks to the cyclic property of the Trace. The quantity $\mathcal{N}_\infty(\lambda)$ can be seen to provide a uniform bound on the leverage scores. In particular note that $\mathcal{N}(\lambda) \leq \mathcal{N}_\infty(\lambda) \leq \frac{\kappa^2}{\lambda}$ [RCR15]. We can now provide a refined version of Theorem 5.

Theorem 7. *Under the same conditions of Theorem 4, the exponent ν in Theorem 4 satisfies $\nu \geq 1/2$, when*

1. *either Nyström uniform sampling is used with $M \geq 70\,[1 + \mathcal{N}_\infty(\lambda)] \log \frac{8\kappa^2}{\lambda\delta}$.*
2. *or Nyström q-approx. lev. scores [RCR15] is used, with $\lambda \geq \frac{19\kappa^2}{n} \log \frac{n}{2\delta}$, $n \geq 405\kappa^2 \log \frac{12\kappa^2}{\delta}$,*

$$M \geq 215\left[2 + q^2\mathcal{N}(\lambda)\right] \log \frac{8\kappa^2}{\lambda\delta}.$$

Considering now the Assumptions 4 and 6 leading to fast rates presented in Section 2.5.2, we can state our main result on fast rates.

Theorem 8. *Let $\delta \in (0, 1]$ and assume $y \in [-\frac{a}{2}, \frac{a}{2}]$, almost surely, with $a > 0$. Under Assumption 1, 4,6, there exist an $n_0 \in \mathbb{N}$ such that for any $n \geq n_0$ the following holds. When*

$$\lambda = n^{-\frac{1}{2r+\alpha}}, \qquad t \;\geq\; \log(n) \;+\; 5 + 2\log(a + 3\kappa^2),$$

1. *and either Nyström uniform sampling is used with $M \geq 70\,[1 + \mathcal{N}_\infty(\lambda)] \log \frac{8\kappa^2}{\lambda\delta}$,*
2. *or Nyström q-approx. lev. scores [RCR15] is used with $M \geq 220\,[2 + q^2\mathcal{N}(\lambda)] \log \frac{8\kappa^2}{\lambda\delta}$,*

then with probability $1 - \delta$,

$$\mathcal{R}(\widehat{f}_{\lambda,M,t}) \;\leq\; c_0 \log^2 \frac{24}{\delta} \;\; n^{-\frac{2r}{2r+\alpha}}.$$

where $\widehat{f}_{\lambda,M,t}$ is the FALKON estimator (Algorithm. 1 in Section 4.2, and Algorithm. 2 in Section 4.10 for the complete version). In particular n_0, c_0 do not depend on λ, M, n, t and c_0 do not depend on δ.

The above result shows that FALKON achieves the same fast rates as KRR, under the same conditions [CDV07]. For $r = 1/2, \alpha = 1$, the rate in Theorem 6 is recovered. If $\alpha < 1, r > 1/2$, FALKON achieves a rate close to $O(1/n)$. By selecting the Nyström points with uniform sampling, a bigger M could be needed for fast rates (albeit always less than n). However, when approximate leverage scores are used M, smaller than $n^{\alpha/2} \ll \sqrt{n}$ is always enough for optimal generalization. This shows that FALKON with approximate leverage scores is the first algorithm to achieve fast rates with a computational complexity that is $O(n\mathcal{N}(\lambda)) = O(n^{1+\frac{\alpha}{2r+\alpha}}) \leq O(n^{1+\frac{\alpha}{2}})$ in time.

4.4 Comparison with Previous Works

In the literature of KRR there are some papers that propose to solve Eq. (4.3) with iterative preconditioned methods [FM12, ACW16, COCF16, GOSS16, MB17]. In particular the one of

[FM12] is based, essentially, on an incomplete singular value decomposition of the kernel matrix. Similarly, the ones proposed by [GOSS16, MB17] are based on singular value decomposition obtained via randomized linear algebra approaches. The first covers the linear case, while the second deals with the kernel case. [ACW16, COCF16] use a preconditioner based on the solution of a randomized projection problem based respectively on random features and Nyström.

While such preconditioners are suitable in the case of KRR, their computational cost becomes too expensive when applied to the random projection case. Indeed, performing an incomplete svd of the matrix $\widehat{K}_{nM}$ even via randomized linear algebra approaches would require $O(nMk)$ where k is the number of singular values to compute. To achieve a good preconditioning level (and so having $t \approx \log n$) we should choose k such that $\sigma_k(\widehat{K}_{nM}) \approx \lambda$. When the kernel function is bounded, without further assumptions on the eigenvalue decay of the kernel matrix, we need $k \approx \lambda^{-1}$ [CDV07, RCR15]. Since randomized projection requires $\lambda = n^{-1/2}$, $M = O(\sqrt{n})$ to achieve optimal generalization bounds, we have $k \approx \sqrt{n}$ and so the total cost of the incomplete svd preconditioner is $O(n^2)$. On the same lines, applying the preconditioner proposed by [ACW16, COCF16] requires $O(nM^2)$ to be computed and there is no natural way to find a similar sketched preconditioner as the one in Eq. (4.7) in the case of [ACW16], with reduced computational cost. In the case of [COCF16], the preconditioner they use is exactly the matrix H^{-1}, whose computation amounts to solve the original problem in Eq. (4.5) with direct methods and requires $O(nM^2)$.

A similar reasoning hold for methods that solve the Nyström linear system (4.5) with iterative approaches [DXH+14, CARR16, TRVR16]. Indeed on the positive side, they have a computational cost of $O(nMt)$. However they are affected by the poor conditioning of the linear system in Eq. (4.5). Indeed, even if H or $\widehat{K}_{MM}$ in Eq. (4.5) are invertible, their condition number can be arbitrarily large (while in the KRR case it is bounded by λ^{-1}), and so many iterations are often needed to achieve optimal generalization (E.g. by using early stopping in [CARR16] they need $t \approx \lambda^{-1}$).

4.5 Generalized FALKON

In this section we define a generalized version of FALKON. In particular we provide a preconditioner able to deal with non invertible $\widehat{K}_{MM}$ and with Nyström centers selected by using approximate leverage scores. In Definition 4 we state the properties that such preconditioner must satisfy.

First we define a diagonal matrix depending on the used sampling scheme that will be needed for the general preconditioner.

Definition 3. *Let $A \in \mathbb{R}^{M\times M}$ be a diagonal matrix. If the Nyström centers are selected via uniform sampling, then $A_{jj} = 1$, for $1 \leq j \leq M$.*

Otherwise, let $i_1, \dots, i_M \in \{1, \dots, n\}$ be the indexes of the training points sampled via approximate leverage scores. Then for $1 \leq j \leq M$,

$$A_{jj} = np_{i_j}.$$

We note here that by definition A is a diagonal matrix with strictly positive and finite diagonal. Indeed it is true in the uniform case. In the leverage scores case, let $1 \leq j \leq M$. Note that since the index i_j has been sampled, it implies that the probability p_{i_j} is strictly larger than zero. Then, since $0 < p_{i_j} \leq 1$ then $0 < A_{jj}^{-1/2} < \infty$ a.s. .

4.5.1 The Algorithm

We now introduce some matrices needed for the definition of a generalized version of FALKON, able to deal with non invertible $\widehat{K}_{MM}$ and with different sampling schemes, for the Nyström centers. Finally in Definition 5, we define a general form of the algorithm, that will be used in the rest of the Chapter.

Definition 4 (The generalized preconditioner)**.** *Let $M \in \mathbb{N}$. Let $\widetilde{x}_1, \dots, \widetilde{x}_M \in \mathcal{X}$ and $\widehat{K}_{MM} \in \mathbb{R}^{M \times M}$ with $(\widehat{K}_{MM})_{ij} = k(\widetilde{x}_i, \widetilde{x}_j)$, for $1 \leq i, j \leq M$. Let $A \in \mathbb{R}^{M \times M}$ be a diagonal matrix with strictly positive diagonal, defined according to Definition 3.*

Let $\lambda > 0$, $q \leq M$ be the rank of $\widehat{K}_{MM}$, $Q \in \mathbb{R}^{M \times q}$ a partial isometry such that $Q^\top Q = I$ and $T \in \mathbb{R}^{q \times q}$ a triangular matrix. Moreover Q, T satisfy the following equation

$$A^{-1/2} \widehat{K}_{MM} A^{-1/2} = QT^\top T Q^\top.$$

Finally let $R \in \mathbb{R}^{q \times q}$ be a triangular matrix such that

$$R^\top R = \frac{1}{M} TT^\top + \lambda I.$$

Then the generalized preconditioner *is defined as*

$$B \;=\; \frac{1}{\sqrt{n}}\, A^{-1/2} Q T^{-1} R^{-1}.$$

Note that B is right invertible, indeed A is invertible, since is a diagonal matrix, with strictly positive diagonal, T, R are invertible since they are square and full rank and Q is a partial isometry, so $B^{-1} = \sqrt{n} R T Q^\top A^{1/2}$ and $BB^{-1} = I$. Now we provide two ways to compute Q, T, R. We recall that the Cholesky algorithm, denoted by chol, given a square positive definite matrix, $B \in \mathbb{R}^{M \times M}$, produces an upper triangular matrix $R \in \mathbb{R}^{M \times M}$ such that $B = R^\top R$. While the pivoted (or rank revealing) QR decomposition, denoted by qr, given a square matrix B, with rank q, produces a partial isometry $Q \in \mathbb{R}^{M \times q}$ with the same range of M and an upper trapezoidal matrix $R \in \mathbb{R}^{q \times M}$ such that $B = QR$.

Example 2 (precoditioner satisfying Definition 4)**.** *Let $\lambda > 0$, and $\widehat{K}_{MM}, A$ as in Definition 4.*

1. *When $\widehat{K}_{MM}$ is full rank ($q = M$), then the following Q, T, R satisfy Definition 4*

$$Q = I, \quad T = \mathrm{chol}(A^{-1/2}\widehat{K}_{MM}A^{-1/2}), \quad R = \mathrm{chol}\left(\frac{1}{M}TT^\top + \lambda I\right).$$

2. *When $\widehat{K}_{MM}$ is of any rank ($q \leq M$), then the following Q, T, R satisfy Definition 4*

$$(Q, R) = \mathrm{qr}(A^{-1/2}\widehat{K}_{MM}A^{-1/2}), \quad T = \mathrm{chol}(Q^\top A^{-1/2}\widehat{K}_{MM}A^{-1/2}Q),$$
$$R = \mathrm{chol}\left(\frac{1}{M}TT^\top + \lambda I\right).$$

Proof. In the first case, Q, T, R satisfy Definition 4 by construction. In the second case, since $QQ^\top$ is the projection matrix on the range of $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$, then

$$QQ^\top A^{-1/2}\widehat{K}_{MM}A^{-1/2} = A^{-1/2}\widehat{K}_{MM}A^{-1/2}$$

and, since $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$ is symmetric,

$$A^{-1/2}\widehat{K}_{MM}A^{-1/2}QQ^\top = A^{-1/2}\widehat{K}_{MM}A^{-1/2},$$

so

$$QT^\top TQ^\top = QQ^\top A^{-1/2}\widehat{K}_{MM}A^{-1/2}QQ^\top = A^{-1/2}\widehat{K}_{MM}A^{-1/2}.$$

Moreover note that, since the rank of $\widehat{K}_{MM}$ is q, then the range of $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$ is q, and so $Q^\top Q = I$, since it is a partial isometry with dimension $\mathbb{R}^{M\times q}$. Finally R satisfies Definition 4 by construction. □

Instead of rank-revealing QR decomposition, eigen-decomposition can be used.

Example 3 (preconditioner for the deficient rank case, using eig instead of qr)**.** *Let $\lambda > 0$, and $\widehat{K}_{MM}, A$ as in Definition 4. Let $(\lambda_i, u_i)_{1\leq i\leq M}$ be respectively the eigenvalues and the associated eigenvectors from the eigendecomposition of $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$, with $\lambda_1 \geq \cdots \geq \lambda_M \geq 0$. So the following Q, T, R satisfy Definition 4, $Q = (u_1, \ldots, u_q)$ and $T = \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_q})$, while $R = \mathrm{diag}\left(\sqrt{\lambda + \frac{1}{M}\lambda_1}, \ldots, \sqrt{\lambda + \frac{1}{M}\lambda_q}\right)$.*

We recall that this approach to compute Q, T, R is conceptually simpler than the one with QR decomposition, but slower, since the hidden constants in the eigendecomposition are larger than the one of QR.

The following is the general form of the algorithm.

Definition 5 (Generalized FALKON algorithm)**.** *Let $\lambda > 0$, $t \in \mathbb{N}$ and q, Q, T, R as in Definition 4. The generalized FALKON estimator is defined as follows*

$$\widehat{f}_{\lambda,M,t}(x) = \sum_{i=1}^{M} k(x, \widetilde{x}_i)c_i, \quad \textit{with} \quad c = B\beta_t,$$

and $\beta_t \in \mathbb{R}^q$ denotes the vector resulting from t iterations of the conjugate gradient algorithm applied to the following linear system

$$W\beta = b, \quad \textit{where} \quad W = B^\top(\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM})B, \quad b = B^\top \widehat{K}_{nM}^\top \widehat{y}. \tag{4.14}$$

4.6 Definitions and Notation for Proofs

Here we recall some basic facts on linear operators and give some notation that will be used in the rest of the Chapter, then we define the necessary operators to deal with the excess risk of FALKON via functional analytic tools.

Notation Let $\mathcal{H}$ be a Hilbert space, we denote with $\|\cdot\|_{\mathcal{H}}$, the associated norm and with $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ the associated inner product. We denote with $\|\cdot\|$ the operator norm for a bounded linear operator A, defined as $\|A\| = \sup_{\|f\|_{\mathcal{H}}=1} \|Af\|$. Moreover we will denote with $\otimes$ the tensor product, in particular

$$(u \otimes v)z = u \langle v, z \rangle_{\mathcal{H}}, \quad \forall u, v, z \in \mathcal{H}.$$

In the rest of the appendix $A + \lambda I$ is often denoted by A_λ where A is linear operator and $\lambda \in \mathbb{R}$, moreover we denote with A^* the adjoint of the linear operator A, we will use $A^\top$ if A is a matrix. When $\mathcal{H}$ is separable, we denote with Tr the trace, that is $\mathrm{Tr}(A) = \sum_{j=1}^{d} \langle u_i, Au_i \rangle_{\mathcal{H}}$ for any linear operator $A : \mathcal{H} \to \mathcal{H}$, where $(u_i)_{j=1}^d$ is an orthogonal basis for $\mathcal{H}$ and $d \in \mathbb{N} \cup \{\infty\}$ is the dimensionality of $\mathcal{H}$. Moreover we denote with $\|\cdot\|_{\mathrm{HS}}$ the Hilbert-Schmidt norm, that is $\|A\|_{\mathrm{HS}}^2 = \mathrm{Tr}(A^*A)$, for a linear operator A.

In the next proposition we recall the spectral theorem for compact self-adjoint operators on a Hilbert space.

Proposition 5 (Spectral Theorem for compact self-adjoint operators)**.** *Let A be a compact self-adjoint operator on a separable Hilbert space $\mathcal{H}$. Then there exists a sequence $(\lambda_j)_{j=1}^d$ with $\lambda_j \in \mathbb{R}$, and an orthogonal basis of $\mathcal{H}$ $(u_j)_{j=1}^d$ where $d \in \mathbb{N} \cup \{\infty\}$ is the dimensionality of $\mathcal{H}$, such that*

$$A = \sum_{j=1}^{d} \lambda_j u_j \otimes u_j. \tag{4.15}$$

Proof. Thm. VI.16, pag. 203 of [RS80]. □

Let $\mathcal{H}$ be a separable Hilbert space (for the sake of simplicity assume $d = \infty$), and A be a bounded self-adjoint operator on $\mathcal{H}$ that admits a spectral decomposition as in Eq. (4.15). Then the largest and the smallest eigenvalues of A are denoted by

$$\lambda_{\max}(A) = \sup_{j \geq 1} \lambda_j, \quad \lambda_{\min}(A) = \inf_{j \geq 1} \lambda_j.$$

In the next proposition we recall a basic fact about bounded symmetric linear operators on a separable Hilbert space $\mathcal{H}$.

Proposition 6. *Let A be a bounded self-adjoint operator on $\mathcal{H}$, that admits a spectral decomposition as in Eq.* (4.15)*. Then*

$$-\|A\| \leq \lambda_{\min}(A) \leq \lambda_{\max}(A) \leq \|A\|.$$

Proof. By definition of operator norm, we have that $\|Ax\|_{\mathcal{H}}^2 \leq \|A\|^2 \|x\|_{\mathcal{H}}^2 \quad \forall x \in \mathcal{H}$. Let $(\lambda_j, u_j)_{j=1}^d$ be an eigendecomposition of A, with d the dimensionality of $\mathcal{H}$, according to Prop. 5, then, for any $j \geq 1$, we have

$$\lambda_j^2 = \langle Au_j, Au_j \rangle = \|Au_j\|_{\mathcal{H}}^2 \leq \|A\|^2,$$

where we used the fact that $Au_j = \lambda_j u_j$ and that $\|u_j\|_{\mathcal{H}} = 1$. □

4.6.1 Definitions

Let $\mathcal{X}$ be a measurable and separable space and $\mathcal{Y} = \mathbb{R}$. Let ρ be a probability measure on $\mathcal{X} \times \mathbb{R}$. We denote with $\rho_{\mathcal{X}}$ the marginal probability of ρ on $\mathcal{X}$ and with $\rho(y|x)$ the conditional probability measure on $\mathcal{Y}$ given $\mathcal{X}$. Let $L^2(\mathcal{X}, \rho_{\mathcal{X}})$ be the Lebesgue space of $\rho_{\mathcal{X}}$ square integrable functions, endowed with the inner product

$$\langle \phi, \psi \rangle_\rho = \int \phi(x)\psi(x) d\rho_{\mathcal{X}}(x), \quad \forall \phi, \psi \in L^2(\mathcal{X}, \rho_{\mathcal{X}}),$$

and norm $\|\psi\|_\rho = \sqrt{\langle \psi, \psi \rangle_\rho}$ for any $\psi \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$. We now introduce the kernel and its associated space of functions. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel, measurable and uniformly bounded, i.e. there exists $\kappa \in (0, \infty)$, for which $k(x, x) \leq \kappa^2$ almost surely. We denote with k_x the function $k(x, \cdot)$ and with $(\mathcal{H}, \ \langle \cdot, \cdot \rangle_{\mathcal{H}})$, the Hilbert space of functions with the associated inner product induced by k, defined by

$$\mathcal{H} = \overline{\operatorname{span}\{k_x \mid x \in \mathcal{X}\}}, \quad \langle k_x, k_{x'} \rangle_{\mathcal{H}} = k(x, x'), \ \ \forall\, x, x' \in \mathcal{X}.$$

Now we define the linear operators used in the rest of the appendix

Definition 6. *Under the assumptions above, for any $f \in \mathcal{H}, \phi \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$*

- $S : \mathcal{H} \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$*, such that* $Sf = \left\langle f, k_{(\cdot)} \right\rangle_{\mathcal{H}} \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$*, with adjoint*
- $S^* : L^2(\mathcal{X}, \rho_{\mathcal{X}}) \to \mathcal{H}$*, such that* $S^*\phi = \int \phi(x) k_x d\rho_{\mathcal{X}}(x) \in \mathcal{H}$*.*
- $L : L^2(\mathcal{X}, \rho_{\mathcal{X}}) \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$*, such that* $L = SS^*$ *and*
- $C : \mathcal{H} \to \mathcal{H}$*, such that* $C = S^*S$*.*

Let $x_i \in \mathcal{X}$ with $1 \leq i \leq n$ and $n \in \mathbb{N}$, and $\widetilde{x}_j \in \mathcal{X}$ for $1 \leq j \leq M$ and $M \in \mathbb{N}$. We define the following linear operators

Definition 7. *Under the assumptions above, for any $f \in \mathcal{H}, v \in \mathbb{R}^n, w \in \mathbb{R}^M$,*

- $\widehat{S}_n : \mathcal{H} \to \mathbb{R}^n$*, such that* $\widehat{S}_n f = \frac{1}{\sqrt{n}}(\langle f, k_{x_i} \rangle)_{i=1}^n \in \mathbb{R}^n$*, with adjoint*
- $\widehat{S}_n^* : \mathbb{R}^n \to \mathcal{H}$*, such that* $\widehat{S}_n^* v = \frac{1}{\sqrt{n}} \sum_{i=1}^n v_i k_{x_i} \in \mathcal{H}$*.*
- $\widehat{C}_n : \mathcal{H} \to \mathcal{H}$*, such that* $\widehat{C}_n = \widehat{S}_n^* \widehat{S}_n$*.*
- $\widehat{S}_M : \mathcal{H} \to \mathbb{R}^M$*, such that* $\widehat{S}_M f = \frac{1}{\sqrt{M}}(\langle f, k_{\widetilde{x}_i} \rangle)_{i=1}^M \in \mathbb{R}^M$*, with adjoint*
- $\widehat{S}_M^* : \mathbb{R}^M \to \mathcal{H}$*, such that* $\widehat{S}_M^* w = \frac{1}{\sqrt{M}} \sum_{i=1}^M v_i k_{\widetilde{x}_i} \in \mathcal{H}$*.*
- $\widehat{C}_M : \mathcal{H} \to \mathcal{H}$*, such that* $\widehat{C}_M = \widehat{S}_M^* \widehat{S}_M$*.*
- $\widehat{G}_M : \mathcal{H} \to \mathcal{H}$*, such that* $\widehat{G}_M = \widehat{S}_M^* A{-}1 \widehat{S}_M$*, with A defined in Definition 4 (see also Definition 3).*

We now recall some basic facts about $L, C, S, \widehat{K}, \widehat{C}_n, \widehat{S}_n$, $\widehat{K}_{nM}$ and $\widehat{K}_{MM}$.

Proposition 7. *With the notation introduced above,*

1. $\widehat{K}_{nM} = \sqrt{nM} \widehat{S}_n \widehat{S}_M^*, \qquad \widehat{K}_{MM} = M\ \widehat{S}_M \widehat{S}_M^*, \qquad \widehat{K} = n\ \widehat{S}_n \widehat{S}_n^*$
2. $C = \displaystyle\int_{\mathcal{X}} k_x \otimes k_x d\rho_{\mathcal{X}}(x), \quad \mathrm{Tr}(C) = \mathrm{Tr}(L) = \|S\|_{HS}^2 = \int_{\mathcal{X}} \|k_x\|_{\mathcal{H}}^2 d\rho_{\mathcal{X}}(x) \leq \kappa^2,$
3. $\widehat{C}_n = \dfrac{1}{n} \displaystyle\sum_{i=1}^n k_{x_i} \otimes k_{x_i}, \qquad \mathrm{Tr}(\widehat{C}_n) = \mathrm{Tr}(\widehat{K}/n) = \|\widehat{S}_n\|_{HS}^2 = \dfrac{1}{n} \displaystyle\sum_{i=1}^n \|k_{x_i}\|_{\mathcal{H}}^2 \leq \kappa^2,$
4. $\widehat{C}_M = \dfrac{1}{M} \displaystyle\sum_{i=1}^M k_{\widetilde{x}_i} \otimes k_{\widetilde{x}_i}, \quad \mathrm{Tr}(\widehat{C}_M) = \mathrm{Tr}(\widehat{K}_{MM}/M) = \|\widehat{S}_M\|_{HS}^2 = \dfrac{1}{M} \displaystyle\sum_{i=1}^m \|k_{\widetilde{x}_i}\|_{\mathcal{H}}^2 \leq \kappa^2,$
5. $\widehat{G}_M = \dfrac{1}{M} \displaystyle\sum_{i=1}^M A_{ii}^{-1} k_{\widetilde{x}_i} \otimes k_{\widetilde{x}_i}.$

where $\otimes$ denotes the tensor product.

Proof. Note that $(\widehat{K}_{nM})_{ij} = k(x_i, \widetilde{x}_j) = \left\langle k_{x_i}, k_{\widetilde{x}_j} \right\rangle_{\mathcal{H}} = (\sqrt{nM}\widehat{S}_n \widehat{S}_M^*)_{ij}$, for any $1 \leq i \leq n$, $1 \leq j \leq M$, thus $\widehat{K}_{nM} = \sqrt{nM}\widehat{S}_n\widehat{S}_M^*$. The same reasoning holds for $\widehat{K}_{MM}$ and $\widehat{K}$. For the second equation, by definition of $C = S^*S$ we have that, for each $h, h' \in \mathcal{H}$,

$$\langle h, Ch' \rangle_{\mathcal{H}} = \langle Sh, Sh' \rangle_{\rho} = \int_{\mathcal{X}} \langle h, k_x \rangle_{\mathcal{H}} \langle k_x, h' \rangle_{\mathcal{H}} \, d\rho_{\mathcal{X}}(x) = \int_{\mathcal{X}} \left\langle h, \left(k_x \langle k_x, h' \rangle_{\mathcal{H}} \right) \right\rangle_{\mathcal{H}} d\rho_{\mathcal{X}}(x)$$
$$= \int_{\mathcal{X}} \left\langle h, \left(k_x \otimes k_x \right) h' \right\rangle_{\mathcal{H}} d\rho_{\mathcal{X}}(x) = \left\langle h, \left(\int_{\mathcal{X}} k_x \otimes k_x d\rho_{\mathcal{X}}(x) \right) h' \right\rangle_{\mathcal{H}}.$$

Note that, since k is bounded almost surely, then $\|k_x\|_{\mathcal{H}} \leq \kappa$ for any $x \in \mathcal{X}$, thus

$$\mathrm{Tr}(C) = \int_{\mathcal{X}} \mathrm{Tr}(k_x \otimes k_x) d\rho_{\mathcal{X}}(x) = \int_{\mathcal{X}} \|k_x\|_{\mathcal{H}}^2 d\rho_{\mathcal{X}}(x) \leq \kappa^2$$

by linearity of the trace. Thus $\mathrm{Tr}(C) < \infty$ and so

$$\mathrm{Tr}(C) = \mathrm{Tr}(S^*S) = \|S\|_{HS}^2 = \mathrm{Tr}(SS^*) = \mathrm{Tr}(L).$$

The proof for the rest of equations is analogous to the one for the second. □

Now we recall a standard characterization of the excess risk

Proposition 8. *When $\int_{\mathcal{Y}} y^2 d\rho(y|x) < \infty$, then there exist $f_\rho \in L^2(\mathcal{X}, \rho_{\mathcal{X}})$ defined by*

$$f_\rho(x) = \int y d\rho(y|x),$$

almost everywhere. Moreover, for any $\widehat{f} \in \mathcal{H}$ we have,

$$\mathcal{E}(\widehat{f}\,) - \inf_{f \in \mathcal{H}} \mathcal{E}(f) = \left\| S\widehat{f} - P f_\rho \right\|_{\rho_{\mathcal{X}}}^2,$$

where $P : L^2(\mathcal{X}, \rho_{\mathcal{X}}) \to L^2(\mathcal{X}, \rho_{\mathcal{X}})$ is the projection operator whose range is the closure in $L^2(\mathcal{X}, \rho_{\mathcal{X}})$ of the range of S.

Proof. Page 890 of [VRC+05]. □

4.7 Analytic results

The section of analytic results is divided in two subsections, where we bound the condition number of the FALKON preconditioned linear system (4.14) and we decompose the excess risk of FALKON, with respect to analytical quantities that will be controlled in probability in the following sections.

4.7.1 Analytic Results (I): Controlling Condition Number of W

First we characterize the matrix W defining the FALKON preconditioned linear system (4.14), with respect to the operators defined in Definition 7 (see next lemma) and in particular we characterize its condition number with respect to the norm of an auxiliary operator defined in Lemma 12. Finally we bound the norm of such operator with respect to analytical quantities more amenable to be bounded in probability (Lemma 13).

Lemma 11 (Characterization of W)**.** *Let $\lambda \in \mathbb{R}$. The matrix W in Definition 5 is characterized by*

$$W = R^{-\top} V^* (\widehat{C}_n + \lambda I) V R^{-1}, \quad \textit{with} \quad V = \sqrt{nM}\widehat{S}_M^* B R.$$

Moreover V is a partial isometry such that $V^*V = I_{q\times q}$ and VV^* with the same range of $\widehat{S}_M^*$.

Proof. By the characterization of $\widehat{K}_{nM}, \widehat{K}_{MM}$ and $\widehat{C}_n$ in Prop. 7, we have

$$\begin{aligned}\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda \widehat{K}_{MM} &= nM\ (\widehat{S}_M \widehat{S}_n^* \widehat{S}_n \widehat{S}_M^* + \lambda \widehat{S}_M \widehat{S}_M^*) \\ &= nM\ \widehat{S}_M (\widehat{S}_n^* \widehat{S}_n + \lambda I) \widehat{S}_M^* = nM\ \widehat{S}_M (\widehat{C}_n + \lambda I) \widehat{S}_M^*.\end{aligned}$$

Now note that, by definition of B in Definition 4 and of V, we have

$$\sqrt{nM}\widehat{S}_M^* B = \sqrt{nM}\widehat{S}_M^* B R R^{-1} = V R^{-1},$$

so

$$\begin{aligned}W &= B^\top (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda \widehat{K}_{MM}) B \\ &= nM\ \ B^\top \widehat{S}_M (\widehat{C}_n + \lambda I) \widehat{S}_M^* B \\ &= R^{-\top} V^* (\widehat{C}_n + \lambda I) V R^{-1}.\end{aligned}$$

The last step is to prove that V is a partial isometry. First we need a characterization of V that is obtained by expanding the definition of B,

$$V = \sqrt{nM}\widehat{S}_M^* B R = \sqrt{nM}\widehat{S}_M^* \frac{1}{\sqrt{n}} A^{-1/2} Q T^{-1} R^{-1} R = \sqrt{M}\widehat{S}_M^* A^{-1/2} Q T^{-1}. \tag{4.16}$$

By the characterization of V, the characterization of $\widehat{K}_{MM}$ in Prop. 7 and the definition of Q, T in terms of $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$ in Definition 4 , we have

$$\begin{aligned}V^*V &= M T^{-\top} Q^\top A^{-1/2}\ \widehat{S}_M \widehat{S}_M^*\ A^{-1/2} Q T^{-1} \\ &= T^{-\top} Q^\top\ A^{-1/2} \widehat{K}_{MM} A^{-1/2}\ Q T^{-1} \\ &= T^{-\top} Q^\top Q T^\top T Q^\top Q T^{-1} = I.\end{aligned}$$

Moreover, by the characterization of V, of $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$ with respect to $\widehat{S}_M$, and of Q, T (Prop. 7 and Definition 4),

$$\begin{aligned} VV^*\widehat{S}_M^*A^{-1/2} &= M\ \widehat{S}_M^*DQT^{-1}T^{-\top}Q^\top A^{-1/2}\widehat{S}_M\widehat{S}_M^* \\ &= \widehat{S}_M^*A^{-1/2}QT^{-1}T^{-\top}Q^\top A^{-1/2}\widehat{K}_{MM}A^{-1/2} \\ &= \widehat{S}_M^*A^{-1/2}QT^{-1}T^{-\top}Q^\top QT^\top TQ^\top \\ &= \widehat{S}_M^*A^{-1/2}QQ^\top = \widehat{S}_M^*A^{-1/2}, \end{aligned}$$

where the last step is due to the fact that the range of $QQ^\top$ is the one of $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$ by definition (see Definition 4), and since $A^{-1/2}\widehat{K}_{MM}A^{-1/2} = MA^{-1/2}\widehat{S}_M\widehat{S}_M^*A^{-1/2}$ by Proposition 7, it is the same of $A^{-1/2}\widehat{S}_M$. Note finally that the range of $\widehat{S}_M^*A^{-1/2}$ is the same of $\widehat{S}_M^*$ since $A^{-1/2}$ is a diagonal matrix with strictly positive elements on the diagonal (see Definition 4). □

Lemma 12. *Let $\lambda > 0$ and W be as in Eq.* (4.14)*. Let $E = R^{-\top}V^*(\widehat{C}_n - \widehat{G}_M)VA^{-1}$, with V defined in Lemma 11. Then W is characterized by*

$$W = I + E.$$

In particular, when $\|E\| < 1$,

$$\operatorname{cond}(W) \leq \frac{1 + \|E\|}{1 - \|E\|}.$$

Proof. Let Q, T, R, A as in Definition 4, and V as in Lemma 11. According to Lemma 11 we have

$$W = R^{-\top}V^*(\widehat{C}_n + \lambda I)VR^{-1} = R^{-\top}(V^*\widehat{C}_nV + \lambda I)R^{-1}.$$

Now we bound the largest and the smallest eigenvalue of W. First of all note that

$$R^{-\top}(V^*\widehat{C}_nV + \lambda I)R^{-1} = R^{-\top}(V^*\widehat{G}_MV + \lambda I)R^{-1} + R^{-\top}V^*(\widehat{C}_n - \widehat{G}_M)VR^{-1}, \quad (4.17)$$

where $\widehat{G}_M$ is defined in Definition 7. To study the first term, we need a preliminary result, which simplifies $\widehat{S}_MV$. By using the definition of V, the characterization of $\widehat{K}_{MM}$ in terms of $\widehat{S}_M$ (Prop. 7), the definition of B (Definition 4), and finally the characterization of $A^{-1/2}\widehat{K}_{MM}A^{-1/2}$ in terms of Q, T (Definition 4), we have

$$\begin{aligned} A^{-1/2}\widehat{S}_MV &= \sqrt{nM}A^{-1/2}\widehat{S}_M\widehat{S}_M^*BR = \sqrt{\frac{n}{M}}A^{-1/2}\widehat{K}_{MM}BR = \frac{1}{\sqrt{M}}A^{-1/2}\widehat{K}_{MM}A^{-1/2}\,QT^{-1} \\ &= \frac{1}{\sqrt{M}}QT^\top TQ^\top QT^{-1} = \frac{1}{\sqrt{M}}QT^\top. \end{aligned}$$

Now we can simplify the first term. We express $\widehat{G}_M$ with respect to $\widehat{S}_M$, then we apply the identity above on $A^{-1/2}\widehat{S}_M V$ and on its transpose, finally we recall the identity $R^\top R = \frac{1}{M}TT^\top + \lambda I$ from Definition 4, obtaining

$$R^{-\top}(V^*\widehat{G}_M V + \lambda I)R^{-1} = R^{-\top}(V^*\widehat{S}_M^* A^{-1}\widehat{S}_M V + \lambda I)R^{-1} = R^{-\top}(\frac{1}{M}TQ^\top QT^\top + \lambda I)R^{-1} \tag{4.18}$$

$$= R^{-\top}(\frac{1}{M}TT^\top + \lambda I)R^{-1} = R^{-\top}R^\top RR^{-1} = I. \tag{4.19}$$

So, by defining $E := R^{-\top}V^*(\widehat{C}_n - \widehat{G}_M)VR^{-1}$, we have

$$W = I + E.$$

Note that E is compact and self-adjoint, by definition. Then, by Proposition 5, 6 we have that W admits a spectral decomposition as in Eq. (4.15). Let $\lambda_{\max}(W)$ and $\lambda_{\min}(W)$ be respectively the largest and the smallest eigenvalues of W, by Proposition 6, and considering that $-\|E\| \leq \lambda_j(E) \leq \|E\|$ (see Proposition 5) we have

$$\lambda_{\max}(W) = \sup_{j\in\mathbb{N}} 1 + \lambda_j(E) = 1 + \sup_{j\in\mathbb{N}} \lambda_j(E) = 1 + \lambda_{\max}(E) \leq 1 + \|E\|,$$
$$\lambda_{\min}(W) = \inf_{j\in\mathbb{N}} 1 + \lambda_j(E) = 1 + \inf_{j\in\mathbb{N}} \lambda_j(E) = 1 + \lambda_{\min}(E) \geq 1 - \|E\|.$$

Since W is self-adjoint and positive, when $\|E\| < 1$, by definition of condition number, we have

$$\operatorname{cond}(W) = \frac{\lambda_{\max}(W)}{\lambda_{\min}(W)} \leq \frac{1 + \|E\|}{1 - \|E\|}.$$

□

Lemma 13. *Let E be defined as in Lemma 12 and let $\widehat{G}_M$ as in Definition 7, then*

$$\|E\| \leq \left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\|. \tag{4.20}$$

Proof. By multiplying and dividing by $\widehat{G}_{M\lambda} = \widehat{G}_M + \lambda I$ we have

$$\|E\| = \left\|R^{-\top}V^*\widehat{G}_{M\lambda}^{1/2}\ \widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\ \widehat{G}_{M\lambda}^{1/2}VR^{-1}\right\|$$
$$\leq \left\|R^{-\top}V^*\widehat{G}_{M\lambda}^{1/2}\right\|^2 \left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\|.$$

Now, considering that $V^*V = I$ and the identity in Eq. (4.18), we have

$$\left\|R^{-\top}V^*\widehat{G}_{M\lambda}^{1/2}\right\|^2 = \left\|R^{-\top}V^*(\widehat{G}_M + \lambda I)VR^{-1}\right\| = \left\|R^{-\top}(V^*\widehat{G}_M V + \lambda I)R^{-1}\right\| = 1. \tag{4.21}$$

□

4.7.2 Analytic Results (II): The Computational Oracle Inequality

In this subsection (Lemma 18) we bound the excess risk of FALKON with respect to the one of the exact Nyström estimator. First we prove that FALKON is equal to the exact Nyström estimator as the iterations go to infinity (Lemma 14, 15). Then in Lemma 18 (via Lemma 16, 17) we use functional analytic tools, together with results from operator theory to relate the weak convergence result of the conjugate gradient method on the chosen preconditioned problem, with the excess risk.

Lemma 14 (Representation of the FALKON estimator as vector in $\mathcal{H}$)**.** *Let $\lambda > 0$, $M, t \in \mathbb{N}$ and B as in Definition 4. The FALKON estimator as in Definition 5 is characterized by the vector $\widehat{f} \in \mathcal{H}$ as follows,*

$$\widehat{f}_{\lambda,M,t} \;=\; \sqrt{M}\, \widehat{S}_M^* B \beta_t, \tag{4.22}$$

where $\beta_t \in \mathbb{R}^q$ denotes the vector resulting from t iterations of the conjugate gradient algorithm applied to the linear system in Definition 5.

Proof. According to the definition of $\widehat{f}_{\lambda,M,t}(\cdot)$ in Definition 5 and the definition of the operator $\widehat{S}_M$ in Definition 7, denoting with $\alpha \in \mathbb{R}^M$ the vector $B\beta_t$, we have that

$$\widehat{f}_{\lambda,M,t}(x) = \sum_{i=1}^{M} k(x, \widetilde{x}_i) c_i = \left\langle k_x, \sum_{i=1}^{M} c_i k_{\widetilde{x}_i} \right\rangle_{\mathcal{H}} = \left\langle k_x,\ \sqrt{M}\, \widehat{S}_M^* c \right\rangle_{\mathcal{H}},$$

for any $x \in \mathcal{X}$. Then the vector in $\mathcal{H}$ representing the function $\widehat{f}_{\lambda,M,t}(\cdot)$ is

$$\widehat{f}_{\lambda,M,t} = \sqrt{M}\, \widehat{S}_M^* c = \sqrt{M}\, \widehat{S}_M^* B \beta_t.$$

□

Lemma 15 (Representation of the Nyström estimator as a vector in $\mathcal{H}$)**.** *Let $\lambda > 0, M \in \mathbb{N}$, and B as in Definition 4. The exact Nyström estimator, in Eq.*(4.4) *and Eq.* (4.5) *is characterized by the vector $\widetilde{f} \in \mathcal{H}$ as follows*

$$\widetilde{f}_{\lambda,M} \;=\; \sqrt{M}\, \widehat{S}_M^* B \beta_\infty, \tag{4.23}$$

where $\beta_\infty = W^{-1} B^\top \widehat{K}_{nM}^\top \widehat{y}$ is the vector resulting from infinite iterations of the conjugate gradient algorithm applied to the linear system in Eq. (4.14)*.*

Proof. For the same reasoning in the proof of Lemma 14, we have that the FALKON estimator with infinite iterations is characterized by the following vector in $\mathcal{H}$

$$\widetilde{f}_{\lambda,M} = \sqrt{M}\, \widehat{S}_M^* B \beta_\infty.$$

To complete the proof, we need to prove 1) that $\beta_\infty = W^{-1}B^\top \widehat{K}_{nM}\widehat{y}$ and 2) that $\widetilde{f}_{\lambda,M}$ above, corresponds to the exact Nyström estimator, as in Eq. (4.5).

Now we characterize β_∞. First, by the characterization of W in Lemma 11 and the fact that $V^*V = I$, we have

$$W = R^{-\top}V^*(\widehat{C}_n + \lambda I)VR^{-1} = R^{-\top}(V^*\widehat{C}_nV + \lambda I)R^{-1}. \tag{4.24}$$

Since $\widehat{C}_n$ is a positive operator (see Definition 7) R is invertible and $\lambda > 0$, then W is a symmetric and positive definite matrix. The positive definiteness of W implies that it is invertible and that is has a finite condition number, making the conjugate gradient algorithm to converge to the solution of the system in Eq. (4.14) (Thm. 6.6 of [Saa03] and Eq. 6.107). So we can explicitly characterize β_∞, by the solution of the system in Eq. (4.14), that is

$$\beta_\infty = W^{-1}B^\top \widehat{K}_{nM}^\top \widehat{y}. \tag{4.25}$$

So we proved that $\widetilde{f}_{\lambda,M} \in \mathcal{H}$, with the above characterization of β_∞, corresponds to FALKON with infinite iterations. Now we show that $\widetilde{f}_{\lambda,M}$ is equal to the Nyström estimator given in [RCR15]. First we need to study $\widehat{S}_M^* B W^{-1} B^\top \widehat{S}_M$. By the characterization of W in Eq. (4.24), the identity $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$, valid for any A, B, C bounded invertible operators, and the definition of V (Lemma 11),

$$\begin{aligned}
\widehat{S}_M^* B W^{-1} B^\top \widehat{S}_M &= \widehat{S}_M^* B \left(R^{-\top}(V^*\widehat{C}_nV + \lambda I)R^{-1}\right)^{-1} B^\top \widehat{S}_M && (4.26)\\
&= \widehat{S}_M^* B R (V^*\widehat{C}_nV + \lambda I)^{-1} R^\top B^\top \widehat{S}_M && (4.27)\\
&= \frac{1}{Mn} V (V^*\widehat{C}_nV + \lambda I)^{-1} V^*. && (4.28)
\end{aligned}$$

By expanding β_∞, $\widehat{K}_{nM}$ (see Lemma 7) in $\widetilde{f}_{\lambda,M}$,

$$\begin{aligned}
\widetilde{f}_{\lambda,M} &= \sqrt{M}\ \widehat{S}_M^* B \beta_\infty = \sqrt{M}\ \widehat{S}_M^* B W^{-1} B^\top \widehat{K}_{nM}^\top \widehat{y} = \sqrt{n}M\ \widehat{S}_M^* B W^{-1} B^\top \widehat{S}_M \widehat{S}_n^* \widehat{y} && (4.29)\\
&= \frac{1}{\sqrt{n}}\ V (V^*\widehat{C}_nV + \lambda I)^{-1} V^* \widehat{S}_n^* \widehat{y}. && (4.30)
\end{aligned}$$

Now by Lemma 2 of [RCR15] with $Z_m = \widehat{S}_M$, we know that the exact Nyström solution is characterized by the vector $\overline{f} \in \mathcal{H}$ defined as follows

$$\overline{f} = \frac{1}{\sqrt{n}} \overline{V} (\overline{V}^* \widehat{C}_n \overline{V} + \lambda I)^{-1} \overline{V}^* \widehat{S}_n^* \widehat{y},$$

with $\overline{V}$ a partial isometry, such that $\overline{V}^*\overline{V} = I$ and $\overline{V}\overline{V}^*$ with the same range of $\widehat{S}_M^*$. Note that, by definition of V in Lemma 11, we have that it is a partial isometry such that $V^*V = I$ and VV^* with the same range of $\widehat{S}_M^*$. This implies that $\overline{V} = VG$, for an orthogonal matrix $G \in \mathbb{R}^{q\times q}$.

Finally, exploiting the fact that $G^{-1} = G^\top$, that $GG^\top = G^\top G = I$ and that for three invertible matrices A, B, C we have $(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$,

$$\begin{aligned}\overline{f} &= \frac{1}{\sqrt{n}}\overline{V}(\overline{V}^*\widehat{C}_n\overline{V} + \lambda I)^{-1}\overline{V}^*\widehat{S}_n^*\widehat{y} = \frac{1}{\sqrt{n}}VG\left(G^\top(V^*\widehat{C}_nV + \lambda I)G\right)^{-1}G^\top V^*\widehat{S}_n^*\widehat{y} \\ &= \frac{1}{\sqrt{n}}VGG^\top\left(V^*\widehat{C}_nV + \lambda I\right)^{-1}GG^\top V^*\widehat{S}_n^*\widehat{y} = \frac{1}{\sqrt{n}}V\left(V^*\widehat{C}_nV + \lambda I\right)^{-1}V^*\widehat{S}_n^*\widehat{y} = \widetilde{f}_{\lambda,M}.\end{aligned}$$

□

The next lemma is necessary to prove Lemma 18.

Lemma 16. *When $\lambda > 0$ and B is as in Definition 4. then*

$$\sqrt{M}\left\|S\widehat{S}_M^*BW^{-1/2}\right\| \leq n^{-1/2}\left\|S\widehat{C}_{n\lambda}^{-1/2}\right\|.$$

Proof. By the fact that identity $\|Z\|^2 = \|ZZ^*\|$ valid for any bounded operator Z and the identity in Eq. (4.26), we have

$$\begin{aligned}M\left\|S\widehat{S}_M^*BW^{-1/2}\right\|^2 &= M\left\|S\widehat{S}_M^*BW^{-1}B^\top\widehat{S}_MS^*\right\| = \frac{1}{n}\left\|SV(V^*\widehat{C}_nV + \lambda I)^{-1}V^*S^*\right\| \\ &= \frac{1}{n}\left\|SV(V^*\widehat{C}_nV + \lambda I)^{-1/2}\right\|^2.\end{aligned}$$

Denote with $\widehat{C}_{n\lambda}$ the operator $\widehat{C}_n + \lambda I$, by dividing and multiplying for $\widehat{C}_{n\lambda}^{-1/2}$, we have

$$SV(V^*\widehat{C}_nV + \lambda I)^{-1/2} = S\widehat{C}_{n\lambda}^{-1/2}\ \widehat{C}_{n\lambda}^{1/2}V(V^*\widehat{C}_nV + \lambda I)^{-1/2}.$$

The second term is equal to 1, indeed, since $V^*\widehat{C}_{n\lambda}V = V^*\widehat{C}_nV + \lambda I$, and $\|Z\|^2 = \|Z^*Z\|$, for any bounded operator Z, we have

$$\begin{aligned}\left\|\widehat{C}_{n\lambda}^{1/2}V(V^*\widehat{C}_nV + \lambda I)^{-1/2}\right\|^2 &= \left\|(V^*\widehat{C}_nV + \lambda I)^{-1/2}V^*\widehat{C}_{n\lambda}V(V^*\widehat{C}_nV + \lambda I)^{-1/2}\right\| &(4.31)\\ &= \left\|(V^*\widehat{C}_nV + \lambda I)^{-1/2}(V^*\widehat{C}_nV + \lambda I)(V^*\widehat{C}_nV + \lambda I)^{-1/2}\right\| &(4.32)\\ &= 1. &(4.33)\end{aligned}$$

Finally

$$\begin{aligned}\sqrt{M}\left\|S\widehat{S}_M^*BW^{-1/2}\right\| &= \frac{1}{\sqrt{n}}\left\|SV(V^*\widehat{C}_nV + \lambda I)^{-1/2}\right\| \\ &\leq \frac{1}{\sqrt{n}}\left\|S\widehat{C}_{n\lambda}^{-1/2}\right\|\left\|\widehat{C}_{n\lambda}^{1/2}V(V^*\widehat{C}_nV + \lambda I)^{-1/2}\right\| \\ &\leq n^{-1/2}\left\|S\widehat{C}_{n\lambda}^{-1/2}\right\|.\end{aligned}$$

□

The next lemma is necessary to prove Lemma 18.

Lemma 17. *For any $\lambda > 0$, let β_∞ be the vector resulting from infinite iterations of the conjugate gradient algorithm applied to the linear system in Eq.* (4.14)*. Then*

$$\left\|W^{1/2}\beta_\infty\right\|_{\mathbb{R}^q} \leq \|\widehat{y}\|_{\mathbb{R}^n}.$$

Proof. First we recall the characterization of β_∞ from Lemma 15,

$$\beta_\infty = W^{-1}B^\top \widehat{K}_{nM}^\top \widehat{y}.$$

So, by the characterization of $\widehat{K}_{nM}$ in terms of $\widehat{S}_n, \widehat{S}_M$ (Prop. 7),

$$W^{1/2}\beta_\infty = W^{1/2}W^{-1}B^\top \widehat{K}_{nM}^\top \widehat{y} = \sqrt{nM}\, W^{-1/2}B^\top \widehat{S}_M \widehat{S}_n^* \widehat{y}.$$

Then, by applying the characterization of $\widehat{S}_M^* B W^{-1} B^\top \widehat{S}_M$ in terms of V, in Eq. (4.26)

$$\begin{aligned}\left\|W^{1/2}\beta_\infty\right\|_{\mathbb{R}^q}^2 = nM \quad \left\|W^{-1/2}B^\top \widehat{S}_M \widehat{S}_n^* \widehat{y}\right\|_{\mathbb{R}^q}^2 &= nM\, \widehat{y}^\top \widehat{S}_n \widehat{S}_M^* B W^{-1} B^\top \widehat{S}_M \widehat{S}_n^* \widehat{y} \\ = \widehat{y}^\top \widehat{S}_n V (V^* \widehat{C}_n V + \lambda I)^{-1} V^* \widehat{S}_n^* \widehat{y} &= \left\|(V^* \widehat{C}_n V + \lambda I)^{-1/2} V^* \widehat{S}_n^* \widehat{y}\right\|_{\mathbb{R}^q}^2.\end{aligned}$$

Finally

$$\left\|(V^* \widehat{C}_n V + \lambda I)^{-1/2} \widehat{S}_n^* \widehat{y}\right\|_{\mathbb{R}^q} \leq \left\|(V^* \widehat{C}_n V + \lambda I)^{-1/2} \widehat{S}_n^*\right\| \|\widehat{y}\|_{\mathbb{R}^n}.$$

Note that

$$\left\|(V^* \widehat{C}_n V + \lambda I)^{-1/2} \widehat{S}_n^*\right\| \leq 1,$$

indeed

$$\left\|(V^* \widehat{C}_n V + \lambda I)^{-1/2} \widehat{S}_n^*\right\| \leq \left\|(V^* \widehat{C}_n V + \lambda I)^{-1/2} \widehat{C}_{n\lambda}^{1/2}\right\| \left\|\widehat{C}_{n\lambda}^{-1/2} \widehat{S}_n^*\right\|,$$

and the first term is equal to 1 by Eq. (4.31), moreover by definition of $\widehat{C}_n$ (Definition 7),

$$\left\|\widehat{C}_{n\lambda}^{-1/2} \widehat{S}_n^*\right\|^2 = \left\|\widehat{C}_{n\lambda}^{-1/2} \widehat{C}_n \widehat{C}_{n\lambda}^{-1/2}\right\| = \left\|\widehat{C}_{n\lambda}^{-1/2} \widehat{C}_n^{1/2}\right\|^2 = \sup_{\sigma \in \sigma(\widehat{C}_n)} \frac{\sigma}{\sigma + \lambda} \leq 1,$$

where $\sigma(\widehat{C}_n) \subset [0, \left\|\widehat{C}_n\right\|]$ is the set of eigenvalues of $\widehat{C}_n$. □

Lemma 18. *Let $M \in \mathbb{N}$, $\lambda > 0$ and B satisfying Definition 5. Let $\widehat{f}_{\lambda,M,t}$ be the FALKON estimator after $t \in \mathbb{N}$ iterations and $\widetilde{f}_{\lambda,M}$ the exact Nyström estimator as in Eq.* (4.4)*, 4.5. Let $c_0 \geq 0$ such that*

$$\left\|S \widehat{C}_{n\lambda}^{-1/2}\right\| \leq c_0,$$

then

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \;\leq\; \mathcal{R}(\widetilde{f}_{\lambda,M})^{1/2} \;+\; 2c_0\, \widehat{v} \left(1 - \frac{2}{\sqrt{cond(W)} + 1}\right)^t,$$

where $\widehat{v}^2 = \frac{1}{n}\sum_{i=1}^n y_i^2$.

Proof of Lemma 18. By Prop. 8 we have that for any $f \in \mathcal{H}$

$$(\mathcal{E}(f) - \inf_{f \in \mathcal{H}} \mathcal{E}(f))^{1/2} = \|Sf - Pf_\rho\|_{\rho\mathcal{X}},$$

with $P : L^2(\mathcal{X}, \rho_\mathcal{X}) \to L^2(\mathcal{X}, \rho_\mathcal{X})$ the orthogonal projection operator whose range is the closure of the range of S in $L^2(\mathcal{X}, \rho_\mathcal{X})$. Let $\widehat{f}_{\lambda,M,t} \in \mathcal{H}$ and $\widetilde{f}_{\lambda,M} \in \mathcal{H}$ be respectively the Hilbert vector representation of the FALKON estimator and of the exact Nyström estimator (Lemma 14 and Lemma 15). By adding and subtracting $\widetilde{f}_{\lambda,M}$ we have

$$\begin{aligned}
|\mathcal{E}(\widehat{f}\,) - \inf_{f \in \mathcal{H}} \mathcal{E}(f)|^{1/2} &= \left\|S\widehat{f}_{\lambda,M,t} - Pf_\rho\right\|_{\rho\mathcal{X}} = \left\|S(\widehat{f}_{\lambda,M,t} - \widetilde{f}_{\lambda,M}) \;+\; (S\widetilde{f}_{\lambda,M} - Pf_\rho)\right\|_{\rho\mathcal{X}} \\
&\le \left\|S(\widehat{f}_{\lambda,M,t} - \widetilde{f}_{\lambda,M})\right\|_{\rho\mathcal{X}} + \left\|S\widetilde{f}_{\lambda,M} - Pf_\rho\right\|_{\rho\mathcal{X}} \\
&= \left\|S(\widehat{f}_{\lambda,M,t} - \widetilde{f}_{\lambda,M})\right\|_{\rho\mathcal{X}} + |\mathcal{E}(\widetilde{f}_{\lambda,M}) - \inf_{f \in \mathcal{H}} \mathcal{E}(f)|^{1/2}.
\end{aligned}$$

In particular, by expanding the definition of $\widehat{f}_{\lambda,M,t}, \widetilde{f}_{\lambda,M}$ from Lemma 14 and Lemma 15, we have

$$\left\|S(\widehat{f}_{\lambda,M,t} - \widetilde{f}_{\lambda,M})\right\|_{\rho\mathcal{X}} = \sqrt{M}\left\|S\widehat{S}_M^* B(\beta_t - \beta_\infty)\right\|_{\rho\mathcal{X}},$$

where $\beta_t \in \mathbb{R}^q$ and $\beta_\infty \in \mathbb{R}^q$ denote respectively the vector resulting from t iterations and infinite iterations of the conjugate gradient algorithm applied to the linear system in Eq. (4.14). Since W is symmetric positive definite when $\lambda > 0$ (see proof of Lemma 15), we can apply the standard convergence results for the conjugate gradient algorithm (Thm. 6.6 of [Saa03], in particular Eq. (6.107)), that is the following

$$\left\|W^{1/2}(\beta_t - \beta_\infty)\right\|_{\mathbb{R}^q} \le q(W,t)\left\|W^{1/2}\beta_\infty\right\|_{\mathbb{R}^q}, \quad \text{with} \quad q(W,t) = 2\left(1 - \frac{2}{\sqrt{\text{cond(W)}} + 1}\right)^t.$$

So by dividing and multiplying by $W^{1/2}$ we have

$$\begin{aligned}
\left\|S(\widehat{f}_{\lambda,M,t} - \widetilde{f}_{\lambda,M})\right\|_{\rho\mathcal{X}} &= \sqrt{M}\left\|S\widehat{S}_M^* B(\beta_t - \beta_\infty)\right\|_{\rho\mathcal{X}} = \sqrt{M}\left\|S\widehat{S}_M^* BW^{-1/2}W^{1/2}(\beta_t - \beta_\infty)\right\|_{\rho\mathcal{X}} \\
&\le \sqrt{M}\left\|S\widehat{S}_M^* BW^{-1/2}\right\|\left\|W^{1/2}(\beta_t - \beta_\infty)\right\|_{\mathbb{R}^q} \\
&\le q(W,t)\sqrt{M}\left\|S\widehat{S}_M^* BW^{-1/2}\right\|\left\|W^{1/2}\beta_\infty\right\|_{\mathbb{R}^q}.
\end{aligned}$$

Finally, the term $\sqrt{M}\left\|S\widehat{S}_M^* BW^{-1/2}\right\|$ is bounded in Lemma 16 as

$$\sqrt{M}\left\|S\widehat{S}_M^* BW^{-1/2}\right\| \le \frac{1}{\sqrt{n}}\left\|S\widehat{C}_{n\lambda}^{-1/2}\right\| \le \frac{c_0}{\sqrt{n}},$$

while, for the term $\left\|W^{1/2}\beta_\infty\right\|_{\mathbb{R}^q}$, by Lemma 17, we have

$$\left\|W^{1/2}\beta_\infty\right\|_{\mathbb{R}^q} \leq \|\widehat{y}\|_{\mathbb{R}^n} = (\sum_{i=1} y_i^2)^{1/2} = \sqrt{n}\sqrt{\frac{\sum_{i=1}^n y_i^2}{n}} = \sqrt{n}\widehat{v}.$$

□

4.8 Probabilistic Estimates

In Lemma 19, 20 we provide probabilistic estimates of $\|E\|$, the quantity needed to bound the condition number of the preconditioned linear system of FALKON (see Lemma 11, 13). In particular Lemma 19, analyzes the case when the Nyström centers are selected with uniform sampling, while Lemma 20, considers the case when the Nyström centers are selected via approximate leverage scores sampling.

Now we are ready to provide probabilistic estimates for uniform sampling.

Lemma 19. *Let $\eta \in [0, 1)$ and $\delta \in (0, 1]$. When $\widetilde{x}_1, \ldots, \widetilde{x}_M$ are selected via Nyström uniform sampling (see Sect. 4.5), $0 < \lambda \leq \|C\|$, $M \leq n$ and*

$$M \geq 4\left[\frac{1}{2} + \frac{1}{\eta} + \left(\frac{3+7\eta}{3+3\eta}\right)\left(1+\frac{2}{\eta}\right)^2 \mathcal{N}_\infty(\lambda)\right]\log\frac{8\kappa^2}{\lambda\delta}, \tag{4.34}$$

then the following hold with probability at least $1-\delta$,

$$\left\|C_\lambda^{-1/2}(C-\widehat{C}_n)C_\lambda^{-1/2}\right\| < \eta, \quad \left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n-\widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\| < \eta.$$

Proof. First of all, note that the Nyström centers are selected by uniform sampling. Then $\widetilde{x}_1, \ldots, \widetilde{x}_M$ are independently and identically distributed according to $\rho_{\mathcal{X}}$ and moreover A is the identity matrix. So

$$\widehat{G}_M = \widehat{S}_M^* A^{-1}\widehat{S}_M = \widehat{S}_M^*\widehat{S}_M = \widehat{C}_M.$$

Note that, by multiplying and dividing by C_λ,

$$\begin{aligned}
\left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n-\widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\| &= \left\|\widehat{C}_{M\lambda}^{-1/2}(\widehat{C}_n-\widehat{C}_M)\widehat{C}_{M\lambda}^{-1/2}\right\| \\
&= \left\|\widehat{C}_{M\lambda}^{-1/2}C_\lambda^{1/2}C_\lambda^{-1/2}(\widehat{C}_n-\widehat{C}_M)C_\lambda^{-1/2}C_\lambda^{1/2}\widehat{C}_{M\lambda}^{-1/2}\right\| \\
&\leq \left\|\widehat{C}_{M\lambda}^{-1/2}C_\lambda^{1/2}\right\|^2 \left\|C_\lambda^{-1/2}(\widehat{C}_n-\widehat{C}_M)C_\lambda^{-1/2}\right\| \\
&\leq (1-\lambda_{\max}(C_\lambda^{-1/2}(C-\widehat{C}_M)C_\lambda^{-1/2}))^{-1}\left\|C_\lambda^{-1/2}(\widehat{C}_n-\widehat{C}_M)C_\lambda^{-1/2}\right\|
\end{aligned}$$

where the last step is due to Proposition 9 of [RR17]. Moreover note that

$$\lambda_{\max}(C_\lambda^{-1/2}(C-\widehat{C}_M)C_\lambda^{-1/2}) \leq \left\|C_\lambda^{-1/2}(C-\widehat{C}_M)C_\lambda^{-1/2}\right\|.$$

Let $\mu = \frac{\delta}{2}$. Note that $\widehat{C}_M = \frac{1}{M}\sum_{i=1}^M v_i \otimes v_i$ with v_i the random variable $v_i = k_{\widetilde{x}_i}$ (see Proposition 7) and, since $\widetilde{x}_1, \ldots, \widetilde{x}_M$ are i.i.d. w.r.t. $\rho_{\mathcal{X}}$, by the characterization of C in Proposition 7, we have for any $1 \leq i \leq M$,

$$\mathbb{E}[v_i \otimes v_i] = \int_{\mathcal{X}} k_x \otimes k_x d\rho_{\mathcal{X}}(x) = C.$$

Then, by considering that $\|v\| = \|k_x\| \leq \kappa^2$, we can apply Proposition 7 of [RR17], obtaining

$$\left\|C_\lambda^{-1/2}(C-\widehat{C}_M)C_\lambda^{-1/2}\right\| \leq \frac{2\zeta(1+\mathcal{N}_\infty(\lambda))}{3M} + \sqrt{\frac{2\zeta\mathcal{N}_\infty(\lambda)}{3M}}, \quad \zeta = \log\frac{4\kappa^2}{\lambda\mu},$$

with probability at least $1-\mu$. Note that, when M satisfies Eq (4.34), we have

$$\left\|C_\lambda^{-1/2}(C-\widehat{C}_M)C_\lambda^{-1/2}\right\| < \eta/(2+\eta).$$

By repeating the same reasoning for C_n, we have

$$\left\|C_\lambda^{-1/2}(C-\widehat{C}_n)C_\lambda^{-1/2}\right\| \leq \frac{2\zeta(1+\mathcal{N}_\infty(\lambda))}{3n} + \sqrt{\frac{2\zeta\mathcal{N}_\infty(\lambda)}{3n}}, \quad \zeta = \log\frac{4\kappa^2}{\lambda\mu},$$

with probability $1-\mu$. Since $n \geq M$ and M satisfying Eq. (4.34), we have automatically that $\left\|C_\lambda^{-1/2}(C-\widehat{C}_n)C_\lambda^{-1/2}\right\| < \eta/(2+\eta)$.

Finally note that, by adding and subtracting C,

$$\begin{aligned}\left\|C_\lambda^{-1/2}(\widehat{C}_n-\widehat{C}_M)C_\lambda^{-1/2}\right\| &= \left\|C_\lambda^{-1/2}((\widehat{C}_n-C)+(C-\widehat{C}_M))C_\lambda^{-1/2}\right\| \\ &\leq \left\|C_\lambda^{-1/2}(C-\widehat{C}_n)C_\lambda^{-1/2}\right\| + \left\|C_\lambda^{-1/2}(C-\widehat{C}_M)C_\lambda^{-1/2}\right\|.\end{aligned}$$

So by performing the intersection bound of the two previous events, we have

$$\begin{aligned}\|\widehat{C}_{M\lambda}^{-1/2}(\widehat{C}_n-\widehat{C}_M)\widehat{C}_{M\lambda}^{-1/2}\| &\leq (1-\left\|C_\lambda^{-1/2}(C-\widehat{C}_n)C_\lambda^{-1/2}\right\|)^{-1}\times \\ &\times\left(\left\|C_\lambda^{-1/2}(C-\widehat{C}_n)C_\lambda^{-1}\right\| + \left\|C_\lambda^{-1/2}(C-\widehat{C}_M)C_\lambda^{-1/2}\right\|\right) < \eta,\end{aligned}$$

with probability at least $1-2\mu$. The last step consists in substituting μ with $\delta/2$. □

The next lemma gives probabilistic estimates for $\|E\|$, that is the quantity needed to bound the condition number of the preconditioned linear system of FALKON (see Lemma 11, 13), when the Nyström centers are selected via approximate leverage scores sampling.

Lemma 20. *Let $\eta > 0$, $\delta \in (0, 1]$, $n, M \in \mathbb{N}$, $q \geq 1$ and $\lambda_0 > 0$. Let $x_1, \dots, x_n$ be independently and identically distributed according to $\rho_{\mathcal{X}}$. Let $\widetilde{x}_1, \dots, \widetilde{x}_M$ be randomly selected from $x_1 \dots, x_n$, by using the (q, λ_0, δ)-approximate leverage scores (see Definition 2 and discussion below), with $\lambda_0 \vee \frac{19\kappa^2}{n} \log \frac{n}{2\delta} \leq \lambda \leq \|C\|$. When $n \geq 405\kappa^2 \vee 67\kappa^2 \log \frac{12\kappa^2}{\delta}$ and*

$$M \geq \left[2 + \frac{2}{\eta} + \frac{18(\eta^2 + 5\eta + 4)q^2}{\eta^2}\mathcal{N}(\lambda)\right] \log \frac{8\kappa^2}{\lambda\delta}, \tag{4.35}$$

then the following hold with probability at least $1 - \delta$,

$$\left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\| < \eta, \quad \left\|C_\lambda^{-1/2}(C - \widehat{C}_n)C_\lambda^{-1/2}\right\| < \eta.$$

Proof. By multiplying and dividing by $\widehat{C}_{n\lambda} = \widehat{C}_n + \lambda I$, we have

$$\begin{aligned}
\left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\| &= \left\|\widehat{G}_{M\lambda}^{-1/2}\widehat{C}_{n\lambda}^{1/2}\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\widehat{C}_{n\lambda}^{1/2}\widehat{G}_{M\lambda}^{-1/2}\right\| \\
&\leq \left\|\widehat{G}_{M\lambda}^{-1/2}\widehat{C}_{n\lambda}^{1/2}\right\|^2 \left\|\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\right\| \\
&\leq (1 - \lambda_{\max}(\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}))^{-1} \left\|\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\right\|
\end{aligned}$$

where the last step is due to Proposition 9 of [RR17]. Note that

$$\lambda_{\max}(\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}) \leq \left\|\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\right\|,$$

thus

$$\left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\| \leq \frac{t}{1-t},$$

with $t = \left\|\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\right\|$. Now we bound t. We denote with $\ell(j, \lambda)$, $\widetilde{\ell}(j, \lambda)$, respectively the leverage scores and the (q, λ_0, δ)-approximate leverage score associated to the point x_j, as in Definition 2 and discussion above. First we need some considerations on the leverage scores. By the spectral theorem and the fact that $\widehat{K} = n\,\widehat{S}_n\widehat{S}_n^*$ (see Proposition 7), we have

$$\begin{aligned}
\ell(j, \lambda) &= (\widehat{K}(\widehat{K} + \lambda n I)^{-1})_{jj} = e_j^\top \widehat{S}_n\widehat{S}_n^*(\widehat{S}_n\widehat{S}_n^* + \lambda I)^{-1}e_j = e_j^\top \widehat{S}_n(\widehat{S}_n^*\widehat{S}_n + \lambda I)^{-1}\widehat{S}_n^* e_j \\
&= \frac{1}{n}\left\langle k_{x_j}, \widehat{C}_{n\lambda}^{-1}k_{x_j}\right\rangle = \frac{1}{n}\left\|\widehat{C}_{n\lambda}^{-1/2}k_{x_j}\right\|^2.
\end{aligned}$$

for any $1 \leq j \leq n$. Moreover, by the characterization of $\widehat{C}_n$ in Prop. 7, we have

$$\frac{1}{n}\sum_{j=1}^{n}\ell(j,\lambda) = \frac{1}{n}\sum_{j=1}^{n}\left\langle k_{x_j}, (\widehat{C}_n+\lambda)^{-1}k_{x_j}\right\rangle_{\mathcal{H}} = \frac{1}{n}\sum_{j=1}^{n}\mathrm{Tr}((\widehat{C}_n+\lambda)^{-1}(k_{x_j}\otimes k_{x_j}))$$
$$= \mathrm{Tr}((\widehat{C}_n+\lambda)^{-1}\frac{1}{n}\sum_{j=1}^{n}(k_{x_j}\otimes k_{x_j})) = \mathrm{Tr}(\widehat{C}_{n\lambda}^{-1}\widehat{C}_n).$$

Since the Nyström points are selected by using the (q,λ_0,δ)-approximate leverage scores, then $\widetilde{x}_t = x_{i_t}$ for $1 \leq t \leq M$, where $i_1,\ldots,i_M \in \{1,\ldots,n\}$ is the sequence of indexes obtained by approximate leverage scores sampling (see Section 4.3.2). Note that $i_1,\ldots,i_M$ are independent random indexes, distributed as follows: for $1 \leq t \leq M$,

$$i_t = j, \quad \text{with probability} \quad p_j = \frac{\widetilde{\ell}(j,\lambda)}{\sum_{h=1}^{n}\widetilde{\ell}(h,\lambda)}, \quad \forall\, 1 \leq j \leq n.$$

Then, by recalling the definition of $\widehat{G}_M$ with respect to the matrix A defined as in Definition 3 and by Prop. 7 we have,

$$\widehat{G}_M = \widehat{S}_M^* A^{-1}\widehat{S}_M = \frac{1}{M}\sum_{t=1}^{M}\frac{1}{np_{i_t}}k_{x_{i_t}}\otimes k_{x_{i_t}}.$$

Consequently $\widehat{G}_M = \frac{1}{M}\sum_{i=1}^{M} v_i \otimes v_i$, where $(v_i)_{i=1}^{M}$ are independent random variables distributed in the following way

$$v_i = \frac{1}{\sqrt{p_j n}}k_{x_j}, \quad \text{with probability } p_j, \quad \forall\, 1 \leq j \leq n.$$

Now we study the moments of $\widehat{G}_M$ as a sum of independent random matrices, to apply non-commutative Bernstein inequality (e.g. Proposition 7 of [RR17]).
We have that, for any $1 \leq i \leq M$

$$\mathbb{E}v_i \otimes v_i = \sum_{j=1}^{n} p_j\left(\frac{1}{p_j n}k_{x_j}\otimes k_{x_j}\right) = \widehat{C}_n,$$

$$\left\langle v_i, \widehat{C}_{n\lambda}^{-1}v_i\right\rangle_{\mathcal{H}} \leq \sup_{1\leq j\leq n}\frac{\left\|\widehat{C}_{n\lambda}^{-1/2}k_{x_j}\right\|^2}{p_j n} = \sup_{1\leq j\leq n}\frac{\ell(j,\lambda)}{p_j n} = \sup_{1\leq j\leq n}\frac{\ell(j,\lambda)}{\widetilde{\ell}(j,\lambda)}\frac{1}{n}\sum_{h=1}^{n}\widetilde{\ell}(h,\lambda)$$
$$\leq q\frac{1}{n}\sum_{h=1}^{n}\widetilde{\ell}(h,\lambda) \leq q^2\frac{1}{n}\sum_{h=1}^{n}\ell(h,\lambda) = q^2\,\mathrm{Tr}(\widehat{C}_{n\lambda}^{-1}\widehat{C}_n),$$

for all $1 \leq j \leq n$. Denote with $\widehat{\mathcal{N}}(\lambda)$, the quantity $\mathrm{Tr}(\widehat{C}_{n\lambda}^{-1}\widehat{C}_n)$, by applying Prop. 7 of [RR17], we have

$$\left\|\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\right\| \leq \frac{2\zeta(1+q^2\widehat{\mathcal{N}}(\lambda))}{3M} + \sqrt{\frac{2\zeta q^2\widehat{\mathcal{N}}(\lambda)}{M}}, \quad \zeta = \log\frac{\kappa^2}{\lambda\mu}.$$

with probability at least $1-\mu$. The final step consist in bounding the empirical intrinsic dimension $\widehat{\mathcal{N}}(\lambda)$ with respect to intrinsic dimension $\mathcal{N}(\lambda)$, for which we use Proposition 1 of [RCR15], obtaining

$$\widehat{\mathcal{N}}(\lambda) \leq 2.65\mathcal{N}(\lambda),$$

with probability at least $1-\mu$, when $n \geq 405\kappa^2 \vee 67\kappa^2 \log\frac{6\kappa^2}{\mu}$ and $\frac{19\kappa^2}{n}\log\frac{n}{4\mu} \leq \lambda \leq \|C\|$. By intersecting the events, we have

$$\left\|\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\right\| \leq \frac{5.3\zeta(1+q^2\mathcal{N}(\lambda))}{3M} + \sqrt{\frac{5.3\zeta q^2\mathcal{N}(\lambda)}{M}}, \quad \zeta = \log\frac{\kappa^2}{\lambda\mu}.$$

with probability at least $1-2\mu$. The last step consist in substituting μ with $\mu = \delta/2$. Thus, by selecting M as in Eq. (4.35), we have

$$t = \left\|\widehat{C}_{n\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{C}_{n\lambda}^{-1/2}\right\| < \frac{\eta}{1+\eta}.$$

That implies,

$$\left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\| < \frac{t}{1-t} < \eta.$$

□

4.9 Proof of Main Results

In this section we prove the main results of the chapter. This section is divided in three subsections. In the first, we specify the computational oracle inequality for Nyström with uniform sampling, in the second we specify the computational oracle inequality for Nyström with approximate leverage scores sampling (see Section 4.8 for a definition), while the third subsection contains the proof of the main theorem presented in the chapter.

Now we give a short sketch of the structure of the proofs. The definition of the general version of the FALKON algorithm (taking into account leverage scores and non invertible $\widehat{K}_{MM}$) is given in Section 4.5. In Section 4.6 the notation and basic definition required for the rest of the analysis are provided.

Our starting point is the analysis of the basic Nyström estimator given in [RCR15]. The key novelty is the quantification of the approximations induced by the preconditioned iterative solver by relating its excess risk to the one of the basic Nyström estimator.

A computational oracle inequality. First we prove that FALKON is equal to the exact Nyström estimator as the iterations go to infinity (Lemma 15, Section 5.2.4). Then, in Lemma 18 (see also Lemma 16, 17, Section 5.2.4) we show how optimization guarantees can be used to derive statistical results. More precisely, while optimization results in machine learning typically derives guarantees on empirical minimization problems, we show, using analytic and probabilistic tools, how these results can be turned into guarantees on the expected risks. Finally, in the proof of Theorem 4 we concentrate the terms of the inequality. The other key point is the study of the behavior of the condition number of $B^\top HB$ with B given in (4.7).

Controlling the condition number of $B^\top HB$. Let C_n, C_M be the empirical correlation operators in $\mathcal{H}$ associated respectively to the training set and the Nyström points $C_n = \frac{1}{n}\sum_{i=1}^n k_{x_i} \otimes k_{x_i}$, $C_M = \frac{1}{M}\sum_{j=1}^M k_{\widetilde{x}_j} \otimes k_{\widetilde{x}_j}$. In Lemma 11, Sect. 5.2.4, we prove that $B^\top HB$ is equivalent to $R^{-\top}V^*(C_n + \lambda I)VR^{-1}$ for a suitable partial isometry V. Then in Lemma 12, Sect. 5.2.4, we split it in two components

$$B^\top HB = R^{-\top}V^*(C_M + \lambda I)VR^{-1} \;+\; R^{-\top}V^*(C_n - C_M)VR^{-1}, \tag{4.36}$$

and prove that the first component is just the identity matrix. By denoting the second component with E, Eq. (4.36), Section 5.2.4, implies that the condition number of $B^\top HB$ is bounded by $(1+\|E\|)/(1-\|E\|)$, when $\|E\| < 1$. In Lemma 13 we prove that $\|E\|$ is analytically bounded by a suitable distance between $C_n - C_M$ and in Lemma 19, 20, Section 4.8, we bound in probability such distance, when the Nyström centers are selected uniformly at random and with approximate leverage scores. Finally in Lemma 21, 22, Section 4.8, we give a condition on M for the two kind of sampling, such that the condition number is controlled and the error term in the oracle inequality decays as $e^{-t/2}$, leading to Theorem 5, 7.

Now we provide the preliminary result necessary to prove a computational oracle inequality for FALKON.

Theorem 4 *Let $0 \le \lambda \le \|C\|$, B as in Definition 4 and $n, M, t \in \mathbb{N}$. Let $\widehat{f}_{\lambda,M,t}$ be the FALKON estimator, with preconditioner B, after t iterations Definition 5 and let $\widetilde{f}_{\lambda,M}$ be the exact Nyström estimator as in Eq.* (4.5)*. Let $\delta \in (0,1]$ and $n \ge 3$, then following holds with probability $1-\delta$*

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \;\le\; \mathcal{R}(\widetilde{f}_{\lambda,M})^{1/2} \;+\; 4\widehat{v}\, e^{-\nu t}\sqrt{1 + \frac{9\kappa^2}{\lambda n}\log\frac{n}{\delta}},$$

where $\widehat{v}^2 = \frac{1}{n}\sum_{i=1}^n y_i^2$ and $\nu = \log\frac{\sqrt{\operatorname{cond}(W)}+1}{\sqrt{\operatorname{cond}(W)}-1}$. In particular $\nu \ge 1/2$, when $\operatorname{cond}(W) \le (\frac{e^{1/2}+1}{e^{1/2}-1})^2$.

Proof. By applying Lemma 18, we have

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \;\leq\; \mathcal{R}(\widetilde{f}_{\lambda,M}\,)^{1/2} \;+\; 2c_0 \left\| S\widehat{C}_{n\lambda}^{-1/2} \right\| \widehat{v}\, e^{-\nu t}.$$

To complete the theorem we need to study the quantity $\left\| S\widehat{C}_{n\lambda}^{-1/2} \right\|$. In particular, define $\lambda_0 = \frac{9\kappa^2}{n} \log \frac{n}{\delta}$. By dividing and multiplying for $\widehat{C}_{n\lambda_0}^{1/2}$, we have

$$\left\| S\widehat{C}_{n\lambda}^{-1/2} \right\| = \left\| S\widehat{C}_{n\lambda_0}^{-1/2} \widehat{C}_{n\lambda_0}^{1/2} \widehat{C}_{n\lambda}^{-1/2} \right\| \leq \left\| S\widehat{C}_{n\lambda_0}^{-1/2} \right\| \left\| \widehat{C}_{n\lambda_0}^{1/2} \widehat{C}_{n\lambda}^{-1/2} \right\|.$$

Now, for the first term, since $\|Z\|^2 = \|Z^*Z\|$, and the fact that $C = S^*S$ (see Prop. 7), we have

$$\left\| S\widehat{C}_{n\lambda_0}^{-1/2} \right\|^2 = \left\| \widehat{C}_{n\lambda_0}^{-1/2} C \widehat{C}_{n\lambda_0}^{-1/2} \right\| = \left\| C^{1/2} \widehat{C}_{n\lambda_0}^{-1/2} \right\|,$$

moreover by Lemma 5 of [RCR15] (or Lemma 7.6 of [RCR13]), we have

$$\left\| C^{1/2} \widehat{C}_{n\lambda_0}^{-1/2} \right\| \leq 2,$$

with probability $1 - \delta$. Finally, by denoting with $\sigma(C)$ the set of eigenvalues of the positive operator C, recalling that $\sigma(C) \subset [0, \kappa^2]$ (see Proposition 7), we have

$$\left\| \widehat{C}_{n\lambda_0}^{1/2} C_{n\lambda}^{-1/2} \right\| = \sup_{\sigma \in \sigma(C)} \sqrt{\frac{\sigma + \lambda_0}{\sigma + \lambda}} \leq \sup_{\sigma \in [0,\kappa^2]} \sqrt{\frac{\sigma + \lambda_0}{\sigma + \lambda}} \leq \sqrt{1 + \frac{\lambda_0}{\lambda}}.$$

□

4.9.1 Main Result (I): Computational Oracle Inequality for FALKON with Uniform Sampling

Lemma 21. *Let $\delta \in (0, 1]$, $0 < \lambda \leq \|C\|$, $n, M \in \mathbb{N}$, the matrix W as in Eq. (4.14) with B satisfying Definition 4 and the Nyström centers selected via uniform sampling. When*

$$M \geq 5\,[1 + 14\mathcal{N}_\infty(\lambda)] \log \frac{8\kappa^2}{\lambda\delta}, \tag{4.37}$$

then the following holds with probability $1 - \delta$

$$\operatorname{cond}(W) \leq \left(\frac{e^{1/2} + 1}{e^{1/2} - 1} \right)^2.$$

Proof. By Lemma 11 we have that

$$\operatorname{cond}(W) \leq \frac{1+\|E\|}{1-\|E\|},$$

with the operator E defined in the same lemma. By Lemma 13, we have

$$\|E\| \leq \left\| \widehat{G}_{M\lambda}^{-1/2} (\widehat{C}_n - \widehat{G}_M) \widehat{G}_{M\lambda}^{-1/2} \right\|.$$

Lemma 19 proves that when the Nyström centers are selected with uniform sampling and M satisfies Eq. (4.34) for a given parameter $\eta \in (0,1]$, then $\left\| \widehat{G}_{M\lambda}^{-1/2} (\widehat{C}_n - \widehat{G}_M) \widehat{G}_{M\lambda}^{-1/2} \right\| \leq \eta$, with probability $1-\delta$. In particular we select $\eta = \frac{2e^{1/2}}{e+1}$. The condition on M in Eq. (4.37) is derived by Eq. (4.34) by substituting η with $\frac{2e^{1/2}}{e+1}$. □

Theorem 9. *Let $\delta \in (0,1]$, $0 < \lambda \leq \|C\|$, $n, M \in \mathbb{N}$ and the Nyström centers be selected via uniform sampling. Let $\widehat{f}_{\lambda,M,t}$ be the FALKON estimator, after t iterations (Definition 5) and let $\widetilde{f}_{\lambda,M}$ be the exact Nyström estimator in Eq.* (4.5)*. When*

$$M \geq 5\left[1 + 14\mathcal{N}_\infty(\lambda)\right] \log \frac{8\kappa^2}{\lambda\delta},$$

then, with probability $1-2\delta$,

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \;\leq\; \mathcal{R}(\widetilde{f}_{\lambda,M})^{1/2} \;+\; 4\widehat{v}\, e^{-\frac{t}{2}} \sqrt{1 + \frac{9\kappa^2}{\lambda n} \log \frac{n}{\delta}},$$

Proof. By applying Lemma 21 we have that

$$\operatorname{cond}(W) \leq (e^{1/2}+1)^2/(e^{1/2}-1)^2,$$

with probability $1-\delta$ under the condition on M. Then apply the computational oracle inequality in Theorem 4 and take the union bound of the two events. □

Theorem 5. *Under the same conditions of Theorem 4, the exponent ν in Theorem 4 satisfies $\nu \geq 1/2$, with probability $1-2\delta$, when the Nyström centers are selected via uniform sampling, and*

$$M \geq 5\left[1 + \frac{14\kappa^2}{\lambda}\right] \log \frac{8\kappa^2}{\lambda\delta}.$$

Proof. It is a direct application of Theorem 9. Indeed note that $\mathcal{N}_\infty(\lambda) \leq \frac{\kappa^2}{\lambda}$ by definition. □

4.9.2 Main Result (II): Computational Oracle Inequality for FALKON with Leverage Scores

Lemma 22. *Let $\delta \in (0,1]$ and the matrix W be as in Eq.* (4.14) *with B satisfying Definition 4 and the Nyström centers selected via (q, λ_0, δ)-approximated leverage scores sampling (see Definition 2 and discussion below), with $\lambda_0 = \frac{19\kappa^2}{n}\log\frac{n}{2\delta}$. When $\lambda_0 \le \lambda \le \|C\|$, $n \ge 405\kappa^2 \vee 67\kappa^2 \log\frac{12\kappa^2}{\delta}$ and*

$$M \ge 5\left[1 + 43q^2\mathcal{N}(\lambda)\right]\log\frac{8\kappa^2}{\lambda\delta}, \tag{4.38}$$

then the following holds with probability $1-\delta$

$$\operatorname{cond}(W) \le \left(\frac{e^{1/2}+1}{e^{1/2}-1}\right)^2.$$

Proof. By Lemma 11 we have that

$$\operatorname{cond}(W) \le \frac{1+\|E\|}{1-\|E\|},$$

with the operator E defined in the same lemma. By Lemma 13 we have

$$\|E\| \le \left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\|.$$

Lemma 20 proves that when the Nyström centers are selected via q-approximate leverage scores and M satisfies Eq. (4.35) for a given parameter $\eta \in (0,1]$, then $\left\|\widehat{G}_{M\lambda}^{-1/2}(\widehat{C}_n - \widehat{G}_M)\widehat{G}_{M\lambda}^{-1/2}\right\| \le \eta$, with probability $1-\delta$. In particular we select $\eta = \frac{2e^{1/2}}{e+1}$. The condition on M in Eq. (4.38) is derived by Eq. (4.35) by substituting η with $\frac{2e^{1/2}}{e+1}$. □

Theorem 10. *Let $\delta \in (0,1], M, n \in \mathbb{N}$ and the Nyström centers be selected via (q, λ_0, δ)-approximated leverage scores sampling (see Definition 2 and discussion below), with $\lambda_0 = \frac{19\kappa^2}{n}\log\frac{n}{2\delta}$. Let $t \in \mathbb{N}$. Let $\widehat{f}_{\lambda,M,t}$ be the FALKON estimator, after t iterations (Definition 5) and let $\widetilde{f}_{\lambda,M}$ be the exact Nyström estimator in Eq.* (4.5)*. When $\lambda_0 \le \lambda \le \|C\|$, $n \ge 405\kappa^2 \vee 67\kappa^2 \log\frac{12\kappa^2}{\delta}$ and*

$$M \ge 5\left[1 + 43q^2\mathcal{N}(\lambda)\right]\log\frac{8\kappa^2}{\lambda\delta},$$

then, with probability $1-2\delta$,

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \;\le\; \mathcal{R}(\widetilde{f}_{\lambda,M})^{1/2} \;+\; 4\widehat{v}\, e^{-\frac{t}{2}}\sqrt{1 + \frac{9\kappa^2}{\lambda n}\log\frac{n}{\delta}},$$

Proof. By applying Lemma 22 we have that

$$\operatorname{cond}(W) \leq (e^{1/2}+1)^2/(e^{1/2}-1)^2,$$

with probability $1-\delta$ under the conditions on λ, n, M. Then apply the computational oracle inequality in Theorem 4 and take the union bound of the two events. □

Theorem 7. *Under the same conditions of Theorem 4, the exponent ν in Theorem 4 satisfies $\nu \geq 1/2$, with probability $1-2\delta$, when*

1. *either Nyström uniform sampling (see Sect. 4.5) is used with $M \geq 70\,[1+\mathcal{N}_\infty(\lambda)]\log\frac{8\kappa^2}{\lambda\delta}$.*
2. *or Nyström (q,λ_0,δ)-appr. lev. scores (see Sect. 4.5) is used, with $\lambda \geq \frac{19\kappa^2}{n}\log\frac{n}{2\delta}$, $n \geq 405\kappa^2\log\frac{12\kappa^2}{\delta}$, and*

$$M \geq 215\left[2+q^2\mathcal{N}(\lambda)\right]\log\frac{8\kappa^2}{\lambda\delta}.$$

Proof. It is a merge of Theorem 9 and Theorem 10. □

4.9.3 Main Results (III): Optimal Generalization Bounds

We now provide Theorem 11, from which we obtain Theorem 6 and Theorem 8.

Theorem 11. *Let $\delta \in (0,1]$. Let n, λ, M satisfy $n \geq 1655\kappa^2 + 223\kappa^2\log\frac{24\kappa^2}{\delta}$, $M \geq 334\log\frac{192n}{\delta}$ and $\frac{19\kappa^2}{n}\log\frac{24n}{\delta} \leq \lambda \leq \|C\|$. Let $\widehat{f}_{\lambda,M,t}$ be the FALKON estimator in Definition 5, after $t \in \mathbb{N}$ iterations. Under the Assumptions 1,3, 6 the following holds with probability at least $1-\delta$,*

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \leq 6R\left(\frac{b\sqrt{\mathcal{N}_\infty(\lambda)}}{n}+\sqrt{\frac{\sigma^2\mathcal{N}(\lambda)}{n}}\right)\log\frac{24}{\delta}+7R\lambda^r, \tag{4.39}$$

1. *either, when the Nyström points are selected uniformly sampled and*

$$M \geq 70\Big[1\ +\ \mathcal{N}_\infty(\lambda)\Big]\log\frac{48\kappa^2}{\lambda\delta}, \qquad t \geq 2\log\frac{8(b+\kappa\,\|f_{\mathcal{H}}\|_{\mathcal{H}})}{R\lambda^r}, \tag{4.40}$$

2. *or, when the Nyström points are selected by means of (q,λ_0,δ)-approximate leverage scores, with $q \geq 1$, $\lambda_0 = \frac{19\kappa^2}{n}\log\frac{48n}{\delta}$ and*

$$M \geq 215\Big[1\ +\ q^2\mathcal{N}(\lambda)\Big]\log\frac{192\kappa^2 n}{\lambda\delta}, \qquad t \geq 2\log\frac{8(b+\kappa\,\|f_{\mathcal{H}}\|_{\mathcal{H}})}{R\lambda^r}. \tag{4.41}$$

Proof. Let $\mu = \delta/4$. By Proposition 2 of [RCR15], under the Assumptions 3 and 6, when $n \geq 1655\kappa^2 + 223\kappa^2 \log \frac{6\kappa^2}{\mu}$, $M \geq 334 \log \frac{48n}{\mu}$, and $\frac{19\kappa^2}{n} \log \frac{6n}{\mu} \leq \lambda \leq \|C\|$, we have with probability $1 - \mu$

$$\mathcal{R}(\widetilde{f}_{\lambda,M})^{1/2} \leq 6R \left(\frac{b\sqrt{\mathcal{N}_\infty(\lambda)}}{n} + \sqrt{\frac{\sigma^2 \mathcal{N}(\lambda)}{n}} \right) \log \frac{6}{\mu} + 3R\mathcal{C}(M)^r + 3R\lambda^r,$$

where

$$\mathcal{C}(M) = \min \left\{ t > 0 \;\middle|\; (67 + 5\mathcal{N}_\infty(t)) \log \frac{12\kappa^2}{t\mu} \leq M \right\},$$

when the Nyström centers are selected with uniform sampling, otherwise

$$\mathcal{C}(M) = \min \left\{ \lambda_0 \leq t \leq \|C\| \;\middle|\; 78q^2 \mathcal{N}(t) \log \frac{48n}{\mu} \leq M \right\},$$

when the Nyström centers are selected via approximate sampling, with $\lambda_0 = \frac{19\kappa^2}{n} \log \frac{12n}{\mu}$. In particular, note that $\mathcal{C}(M) \leq \lambda$, in both cases, when M satisfies Eq. (4.40) for uniform sampling, or Eq. (4.41) for approximate leverage scores. Now, by applying the computational oracle inequality in Theorem 9, for uniform sampling, or Theorem 10, for approximate leverage scores, the following holds with probability $1 - 2\mu$

$$R(\widehat{f}_{\lambda,M,t})^{1/2} \leq \mathcal{R}(\widetilde{f}_{\lambda,M})^{1/2} + 4\widehat{v}\, e^{-\frac{t}{2}} \sqrt{1 + \frac{9\kappa^2}{\lambda n} \log \frac{n}{\mu}},$$

with $\widehat{v}^2 := \frac{1}{n} \sum_{i=1}^n y_i^2$. In particular, note that, since we require $\lambda \geq \frac{19\kappa^2}{n} \log \frac{12n}{\mu}$, we have

$$4\sqrt{1 + \frac{9\kappa^2}{\lambda n} \log \frac{n}{\mu}} \leq 5.$$

Now, we choose t such that $5\widehat{v}e^{-t/2} \leq R\lambda^r$, that is $t \geq 2 \log \frac{5\widehat{v}}{R\lambda^r}$. The last step consists in bounding $\widehat{v}$ in probability. Since it depends on the random variables $y_1, \ldots, y_n$ we bound it in the following way. By recalling that

$$|f_{\mathcal{H}}(x)| = |\langle k_x, f_{\mathcal{H}} \rangle_{\mathcal{H}}| \leq \|k_x\|_{\mathcal{H}} \|f_{\mathcal{H}}\|_{\mathcal{H}} \leq \kappa \|f_{\mathcal{H}}\|_{\mathcal{H}} \tag{4.42}$$

for any $x \in \mathcal{X}$, we have

$$\widehat{v} = \frac{1}{\sqrt{n}} \|\widehat{y}\| \leq \sqrt{\sum_{i=1}^n \frac{(y_i - f_{\mathcal{H}}(x_i))^2}{n}} + \sqrt{\sum_{i=1}^n \frac{f_{\mathcal{H}}(x_i)^2}{n}} \leq \sqrt{\sum_{i=1}^n \frac{(y_i - f_{\mathcal{H}}(x_i))^2}{n}} + \kappa \|f_{\mathcal{H}}\|_{\mathcal{H}}.$$

Since the training set examples $(x_i, y_i)_{i=1}^n$ are i.i.d. with probability ρ we can apply the Bernstein inequality [BLB04] to the random variables $z_i = (y_i - f_{\mathcal{H}}(x_i))^2 - s$, with $s = \mathbb{E}(y_i - f_{\mathcal{H}}(x_i))^2$

(since x_i, y_i are i.i.d. each z_i has the same distribution and so the same expected value s). In particular, we need to bound the moments of z_i's. By the assumption in Eq. (2.42), z_i are zero mean and

$$\mathbb{E}|z_i|^{2p} \leq \frac{1}{2}(2p)!\sigma^2 b^{2p-2} \leq \frac{1}{2}p!(4\sigma b)^2(4b^2)^{p-2}, \qquad p \geq 2$$

and so, by applying the Bernstein inequality, the following holds with probability $1-\mu$

$$\left|\sum_{i=1}^{n}\frac{z_i}{n}\right| \leq \frac{8b^2\log\frac{2}{\mu}}{3n} + \sqrt{\frac{8\sigma^2 b^2\log\frac{2}{\mu}}{n}} \leq \frac{1}{4}b^2,$$

where the last step is due to the fact that we require $n \geq 223\kappa^2\log\frac{6}{\mu}$, that $b \geq \sigma$ and that $\kappa \geq 1$ by definition. So, by noting that $s \leq \sigma^2 \leq b^2$ (see Eq. (2.42)), we have

$$\widehat{v} \;\leq\; \kappa\left\|f_{\mathcal{H}}\right\|_{\mathcal{H}} + \sqrt{s+\sum_{i=1}^{n}\frac{z_i}{n}} \;\leq\; \kappa\left\|f_{\mathcal{H}}\right\|_{\mathcal{H}} + \sqrt{s} + \frac{1}{2}b \;\leq\; \frac{3}{2}b + \kappa\left\|f_{\mathcal{H}}\right\|_{\mathcal{H}},$$

with probability at least $1-\mu$. Now by taking the intersection of the three events, the following holds with probability at least $1-4\mu$

$$\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} \leq 6R\left(\frac{b\sqrt{\mathcal{N}_\infty(\lambda)}}{n} + \sqrt{\frac{\sigma^2\mathcal{N}(\lambda)}{n}}\right)\log\frac{6}{\mu} + 7R\lambda^r.$$

□

Now we provide the generalization error bounds for the setting where we only assume the existence of $f_{\mathcal{H}}$.

Theorem. 6 *Let $\delta \in (0,1]$. Let the outputs y be bounded in $[-\frac{a}{2},\frac{a}{2}]$, almost surely, with $a > 0$. For any $n \geq \max(\frac{1}{\|C\|}, 82\kappa^2\log\frac{373\kappa^2}{\sqrt{\delta}})^2$ the following holds. When*

$$\lambda = \frac{1}{\sqrt{n}}, \qquad M \geq 5(67+20\sqrt{n})\log\frac{48\kappa^2 n}{\delta}, \qquad t \;\geq\; \frac{1}{2}\log(n) \;+\; 5 + 2\log(a+3\kappa),$$

then with probability $1-\delta$,

$$\mathcal{R}(\widehat{f}_{\lambda,M,t}\,) \leq \frac{c_0\log^2\frac{24}{\delta}}{\sqrt{n}},$$

where $\widehat{f}_{\lambda,M,t}$ is the FALKON estimator in Definition 5 (see also Section 4.2 Algorithm 1) with Nyström uniform sampling, and the constant $c_0 = 49\left\|f_{\mathcal{H}}\right\|_{\mathcal{H}}^2(1+a\kappa+2\kappa^2\left\|f_{\mathcal{H}}\right\|_{\mathcal{H}})^2$.

Proof. Here we assume $y \in [-\frac{a}{2}, \frac{a}{2}]$ a.s., so Eq. (2.42) is satisfied with $\sigma = b = a + 2\kappa \|f_{\mathcal{H}}\|_{\mathcal{H}}$, indeed

$$\mathbb{E}[|y - f_{\mathcal{H}}(x)|^p \mid x] \;\leq\; \mathbb{E}[2^{p-1}|y|^p \mid x] + 2^{p-1}|f_{\mathcal{H}}(x)|^p \;\leq\; \frac{1}{2}(a^p + 2^p\kappa^p \|f_{\mathcal{H}}\|_{\mathcal{H}}^p) \leq \frac{1}{2}p!(a + 2\kappa \|f_{\mathcal{H}}\|_{\mathcal{H}})^p,$$

where we used Eq. (4.42). Moreover, Eq. (2.52) is satisfied with $r = 1/2$ and $g = f_{\mathcal{H}}$, while $R = \max(1, \|f_{\mathcal{H}}\|_{\mathcal{H}})$.

To complete the proof we show that the assumptions on λ, M, n satisfy the condition required by Theorem 11, then we apply it and derive the final bound. Set $\lambda = n^{-1/2}$ and define $n_0 = \max(\|C\|^{-1}, 82\kappa^2 \log \frac{373\kappa^2}{\sqrt{\delta}})^2$. The condition $n \geq n_0$, satisfies the condition on n required by Theorem 11. Moreover both $\lambda = n^{-1/2}$ and $M \geq 75\sqrt{n}\log\frac{48\kappa^2 n}{\delta}$ satisfy respectively the conditions on λ, M required by Theorem 11, when $n \geq n_0$. Finally note that the condition on t implies the condition required by Theorem 11, indeed, since $R = \max(1, \|f\|_{\mathcal{H}})$, we have $a/R \leq a$ and $\|f_{\mathcal{H}}\|_{\mathcal{H}}/R \leq 1$, so

$$\begin{aligned} 2\log\frac{8(a + \kappa\|f_{\mathcal{H}}\|_{\mathcal{H}})}{R\lambda^r} \;&=\; \log\left[64\left(\frac{a}{R} + \frac{3\kappa\|f_{\mathcal{H}}\|_{\mathcal{H}}}{R}\right)^2\sqrt{n}\right] \\ &\leq\; \log(64(a+3\kappa)^2\sqrt{n}) \;\leq\; \log 64 + \frac{1}{2}\log n + 2\log(a+3\kappa). \end{aligned}$$

So, by applying Theorem 11 with R, r defined as above and recalling that $\mathcal{N}(\lambda) \leq \mathcal{N}_\infty(\lambda) \leq \frac{\kappa^2}{\lambda}$, we have

$$\begin{aligned} \mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} &\leq 6R\left(\frac{b\sqrt{\mathcal{N}_\infty(\lambda)}}{n} + \sqrt{\frac{\sigma^2\mathcal{N}(\lambda)}{n}}\right)\log\frac{24}{\delta} + 7R\lambda^r \\ &\leq 6R\left(\frac{b\kappa}{\sqrt{\lambda}n} + \frac{\sigma\kappa}{\sqrt{\lambda n}}\right)\log\frac{24}{\delta} + 7R\lambda^{\frac{1}{2}} \\ &= 6Rb\kappa(1 + n^{-1/2})n^{-1/4}\log\frac{24}{\delta} + 7Rn^{-1/4} \leq \frac{7R(b\kappa+1)\log\frac{24}{\delta}}{n^{1/4}} \end{aligned}$$

with probability $1 - \delta$. For the last step we used the fact that $b = \sigma$, that $6(1 + n^{-1/2}) \leq 7$, since $n \geq n_0$, and that $\log\frac{24}{\delta} > 1$. □

To state the result for fast rates, we need the assumption on the *capacity condition* (see Assumption 4)

Theorem 8 *Let $\delta \in (0, 1]$. Let the outputs y be bounded in $[-\frac{a}{2}, \frac{a}{2}]$, almost surely, with $a > 0$. Under the Assumptions 1, 3, 4, 6 and $n \geq \|C\|^{-s} \vee \left(\frac{102\kappa^2 s}{s-1}\log\frac{912}{\delta}\right)^{\frac{s}{s-1}}$, with $s = 2r + \alpha$, the following holds. When*

$$\lambda = n^{-\frac{1}{2r+\alpha}}, \qquad t \;\geq\; \log(n) \;+\; 5 + 2\log(a + 3\kappa^2),$$

1. *and either Nyström uniform sampling is used with*

$$M \geq 70\,[1 \,+\, \mathcal{N}_\infty(\lambda)] \;\log \frac{8\kappa^2}{\lambda\delta}, \tag{4.43}$$

2. *and or Nyström (q, λ_0, δ)-approximate leverage scores (Definition 2), with $q \geq 1$, $\lambda_0 = \frac{19\kappa^2}{n} \log \frac{48n}{\delta}$ and*

$$M \geq 215\left[1 \,+\, q^2\mathcal{N}(\lambda)\right] \;\log \frac{8\kappa^2}{\lambda\delta}, \tag{4.44}$$

then with probability $1-\delta$,

$$\mathcal{R}(\widehat{f}_{\lambda,M,t}) \;\leq\; c_0 \log^2 \frac{24}{\delta} \;\; n^{-\frac{2r}{2r+\alpha}},$$

where $\widehat{f}_{\lambda,M,t}$ is the FALKON estimator in Section 4.2 (Algorithm 1). In particular n_0, c_0 do not depend on λ, M, n and c_0 do not depend on δ.

Proof. The proof is similar to the one for the slow learning rate (Theorem 6), here we take into account the additional assumption in Eq. (2.52), (2.50) and the fact that r may be bigger than $1/2$. Moreover we assume $y \in [-\frac{a}{2}, \frac{a}{2}]$ a.s., so Eq. (2.42) is satisfied with $\sigma = b = a + 2\kappa \|f_\mathcal{H}\|_\mathcal{H}$, indeed

$$\mathbb{E}[|y - f_\mathcal{H}(x)|^p \,|\, x] \;\leq\; \mathbb{E}[2^{p-1}|y|^p \,|\, x] + 2^{p-1}|f_\mathcal{H}(x)|^p \;\leq\; \frac{1}{2}(a^p + 2^p\kappa^p \|f_\mathcal{H}\|_\mathcal{H}^p) \leq \frac{1}{2}p!(a + 2\kappa\|f_\mathcal{H}\|_\mathcal{H})^p,$$

where we used Eq. (4.42).

To complete the proof we show that the assumptions on λ, M, n satisfy the required conditions to apply Theorem 11. Then we apply it and derive the final bound. Set $\lambda = n^{-1/(2r+\alpha)}$ and define $n_0 = \|C\|^{-s} \vee \left(\frac{102\kappa^2 s}{s-1} \log \frac{912}{\delta}\right)^{\frac{s}{s-1}}$, with $s = 2r + \alpha$. Since $1 < s \leq 3$, the condition $n \geq n_0$, satisfies the condition on n required to apply Theorem 11. Moreover, for any $n \geq n_0$, both $\lambda = n^{-1/(2r+\alpha)}$ and M satisfying Eq. (4.43) for Nyström uniform sampling, and Eq. (4.44) for Nyström leverage scores, satisfy respectively the conditions on λ, M required to apply Theorem 11. Finally note that the condition on t implies the condition required by Theorem 11, indeed, since $2r/(2r+\alpha) \leq 1$,

$$\begin{aligned}
2\log\frac{8(b + \kappa\|f_\mathcal{H}\|_\mathcal{H})}{R\lambda^r} \;&=\; \log\left[64\left(\frac{a}{R} + \frac{3\kappa\|f_\mathcal{H}\|_\mathcal{H}}{R}\right)^2 n^{\frac{2r}{2r+\alpha}}\right] \\
&\leq\; \log 64 + 2\log\frac{a + 3\kappa\|f_\mathcal{H}\|_\mathcal{H}}{R} + \frac{2r}{2r+\alpha}\log n \\
&\leq\; \log 64 + 2\log\frac{a + 3\kappa\|f_\mathcal{H}\|_\mathcal{H}}{R} + \log n, \\
&\leq\; \log 64 + 2\log(a + 3\kappa^2) + \log n.
\end{aligned}$$

where the last step is due to the fact that $a/R \leq 1$ and $\|f_h\|_{\mathcal{H}}/R \leq \left\|C^{r-1/2}\right\| \leq \|C\|^{1/2} \leq \kappa$, since $R := \max(1, \|g\|_{\mathcal{H}})$, and $\|f_{\mathcal{H}}\|_{\mathcal{H}} \leq \left\|C^{r-1/2}\right\| \|g\|_{\mathcal{H}}$, by definition. So, by applying Theorem 11 with R, r defined as above and recalling that $\mathcal{N}_\infty(\lambda) \leq \frac{\kappa^2}{\lambda}$ by construction and that $\mathcal{N}(\lambda) \leq Q^2\lambda^{-\alpha}$ by the capacity condition in Eq. (2.50), we have

$$\begin{aligned}
\mathcal{R}(\widehat{f}_{\lambda,M,t})^{1/2} &\leq 6R\left(\frac{b\sqrt{\mathcal{N}_\infty(\lambda)}}{n} + \sqrt{\frac{\sigma^2\mathcal{N}(\lambda)}{n}}\right)\log\frac{24}{\delta} + 7R\lambda^r \\
&\leq 6R\left(\frac{b\kappa}{\sqrt{\lambda}n} + \frac{Q\sigma}{\sqrt{\lambda^\alpha n}}\right)\log\frac{24}{\delta} + 7R\lambda^r \\
&= 6Rb\left(\kappa n^{-\frac{r+\alpha-1/2}{2r+\alpha}} + Q\right)n^{-\frac{r}{2r+\alpha}}\log\frac{24}{\delta} + 7Rn^{-\frac{r}{2r+\alpha}} \\
&\leq\ 7R(b(\kappa+Q)+1)\log\frac{24}{\delta}\ \ n^{-\frac{r}{2r+\alpha}}.
\end{aligned}$$

with probability $1-\delta$. For the last step we used the fact that $b = \sigma$, that $r + \alpha - 1/2 \geq 0$, since $r \geq 1/2$ by definition, and that $\log\frac{24}{\delta} > 1$. □

4.10 Experiments

We present FALKON's performance on a range of large scale datasets.

As shown in Table 4.2, 4.3, FALKON achieves state of the art accuracy and typically outperforms previous approaches in all the considered large scale datasets including IMAGENET. This is remarkable considering FALKON required only a fraction of the competitor's computational resources. Indeed we used a single machine equipped with two Intel Xeon E5-2630 v3, one NVIDIA Tesla K40c and 128 GB of RAM and a basic MATLAB FALKON implementation, while typically the results for competing algorithm have been performed on clusters of GPU workstations (accuracies, times and used architectures are cited from the corresponding papers).

A minimal MATLAB implementation of FALKON is presented in Algorithm 2. The code necessary to reproduce the following experiments, plus a FALKON version that is able to use the GPU, is available on GitHub at `https://github.com/LCSL/FALKON_paper` .

The error is measured with MSE, RMSE or relative error for regression problems, and with classification error (c-err) or AUC for the classification problems, to be consistent with the literature. For datasets which do not have a fixed test set, we set apart 20% of the data for testing. For all datasets, but YELP and IMAGENET, we normalize the features by their z-score. From now on we denote with n the cardinality of the dataset, d the dimensionality.

MillionSongs [BMEWL11] (Table 4.2, $n = 4.6 \times 10^5$, $d = 90$, regression).

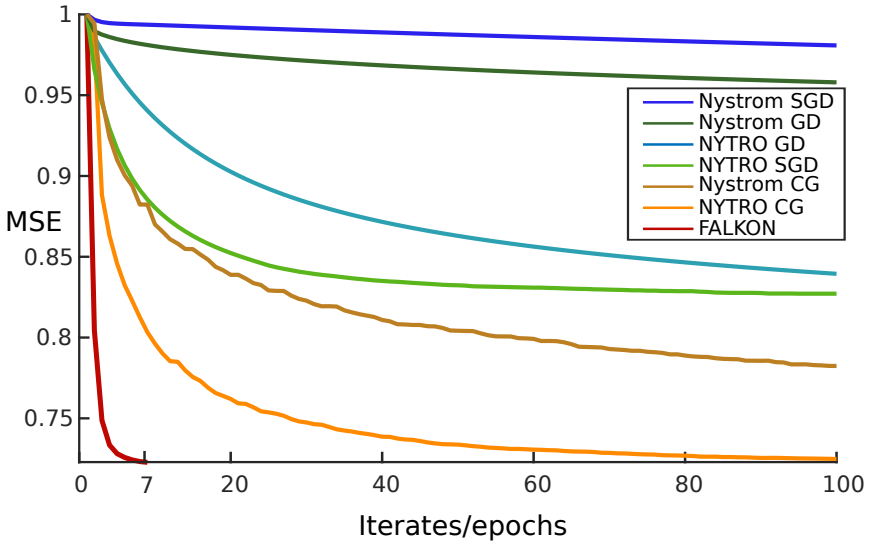

Fig. 4.1: Falkon is compared to stochastic gradient, gradient descent and conjugate gradient applied to Problem (4.5), while NYTRO refer to the variants described in [CARR16]. The graph shows the test error on the HIGGS dataset (1.1×10^7 examples) with respect to the number of iterations (epochs for stochastic algorithms).

We used a Gaussian kernel with $\sigma = 6$, $\lambda = 10^{-6}$ and 10^4 Nyström centers. Moreover with 5×10^4 center, FALKON achieves a 79.20 MSE, and 4.49×10^{-3} rel. error in 630 *sec*.

TIMIT (Table 4.2, $n = 1.2 \times 10^6$, $d = 440$, multiclass classification).

We used the same preprocessed dataset of [MB17] and Gaussian Kernel with $\sigma = 15$, $\lambda = 10^{-9}$ and 10^5 Nyström centers.

YELP (Table 4.2, $n = 1.5 \times 10^6$, $d = 6.52 \times 10^7$, regression).

We used the same dataset of [TRVR16]. We extracted the 3-grams from the plain text with the same pipeline as [TRVR16], then we mapped them in a sparse binary vector which records if the 3-gram is present or not in the example. We used a linear kernel with 5×10^4 Nyström centers. With 10^5 centers, we get a RMSE of 0.828 in 50 minutes.

SUSY (Table 4.3, $n = 5 \times 10^6$, $d = 18$, binary classification).

We used a Gaussian kernel with $\sigma = 4$, $\lambda = 10^{-6}$ and 10^4 Nyström centers.

HIGGS (Table 4.3, $n = 1.1 \times 10^6$, $d = 28$, binary classification).

Each feature has been normalized subtracting its mean and dividing for its variance. We used a Gaussian kernel with diagonal matrix width learned with cross validation on a small validation set, $\lambda = 10^{-8}$ and 10^5 Nyström centers. If we use a single $\sigma = 5$ we reach an AUC of 0.825.

Table 4.2: Architectures: ‡ cluster 128 EC2 r3.2xlarge machines, † cluster 8 EC2 r3.8xlarge machines, ≀ single machine with two Intel Xeon E5-2620, one Nvidia GTX Titan X GPU, 128GB RAM, ⋆ cluster with IBM POWER8 12-core processor, 512 GB RAM, ∗ unknown platform.

	MillionSongs			YELP		TIMIT	
	MSE	Relative error	Time(s)	RMSE	Time(m)	c-err	Time(h)
FALKON	**80.10**	$\mathbf{4.51 \times 10^{-3}}$	**55**	**0.833**	**20**	32.3%	**1.5**
Prec. KRR [ACW16]	-	4.58×10^{-3}	$289^{\dagger}$	-	-	-	-
Hierarchical [CAS16]	-	4.56×10^{-3}	$293^{\star}$	-	-	-	-
D&C [ZDW13]	80.35	-	737^{*}	-	-	-	-
Rand. Feat. [ZDW13]	80.93	-	772^{*}	-	-	-	-
Nyström [ZDW13]	80.38	-	876^{*}	-	-	-	-
ADMM R. F.[ACW16]	-	5.01×10^{-3}	$958^{\dagger}$	-	-	-	-
BCD R. F. [TRVR16]	-	-	-	0.949	$42^{\ddagger}$	34.0%	$1.7^{\ddagger}$
BCD Nyström [TRVR16]	-	-	-	0.861	$60^{\ddagger}$	33.7%	$1.7^{\ddagger}$
EigenPro [MB17]	-	-	-	-	-	32.6%	$3.9^{\wr}$
KRR [CAS16] [TRVR16]	-	4.55×10^{-3}	-	0.854	$500^{\ddagger}$	33.5%	$8.3^{\ddagger}$
Deep NN [MGL+17]	-	-	-	-	-	32.4%	-
Sparse Kernels [MGL+17]	-	-	-	-	-	**30.9%**	-
Ensemble [HAS+14]	-	-	-	-	-	33.5%	-

IMAGENET (Table 4.3, $n = 1.3 \times 10^6$, $d = 1536$, multiclass classification).

We report the top 1 c-err over the validation set of ILSVRC 2012 with a single crop. The features are obtained from the convolutional layers of pre-trained Inception-V4 [SIVA17]. We used Gaussian kernel with $\sigma = 19$, $\lambda = 10^{-9}$ and 5×10^4 Nyström centers. Note that with linear kernel we achieve c-err $= 22.2\%$.

Table 4.3: Architectures: † cluster with IBM POWER8 12-core cpu, 512 GB RAM, ≀ single machine with two Intel Xeon E5-2620, one Nvidia GTX Titan X GPU, 128GB RAM, ‡ single machine [Alv16]

	SUSY			HIGGS		IMAGENET	
	c-err	AUC	Time(m)	AUC	Time(h)	c-err	Time(h)
FALKON	**19.6%**	0.877	**4**	0.833	**3**	20.7%	**4**
EigenPro [MB17]	19.8%	-	$6^{\wr}$	-	-	-	-
Hierarchical [CAS16]	20.1%	-	$40^{\dagger}$	-	-	-	-
Boosted Decision Tree [BSW14]	-	0.863	-	0.810	-	-	-
Neural Network [BSW14]	-	0.875	-	0.816	-	-	-
Deep Neural Network [BSW14]	-	**0.879**	$4680^{\ddagger}$	**0.885**	$78^{\ddagger}$	-	-
Inception-V4 [SIVA17]	-	-	-	-	-	**20.0%**	-

Algorithm 2: Complete `MATLAB` code for FALKON. It requires $O(nMt + M^3)$ in time and $O(M^2)$ in memory. See Sect. 4.2 for more details, and Sect. 4.3 for theoretical properties.

Input: Dataset $X = (x_i)_{i=1}^n \in \mathbb{R}^{n \times d}$, $\widehat{y} = (y_i)_{i=1}^n \in \mathbb{R}^n$, $M \in \mathbb{N}$ numbers of Nyström centers to select, lev_scores $\in \mathbb{R}^n$ approximate leverage scores (set lev_scores = [ ] for selecting Nyström centers via uniform sampling), function KernelMatrix computing the kernel matrix of two sets of points, regularization parameter λ, number of iterations t.
Output: Nyström coefficients α.

```
function alpha = FALKON(X, Y, lev_scores, M, KernelMatrix, lambda, t)
    n = size(X,1);
    [C, D] = selectNystromCenters(X, lev_scores, M, n);

    KMM = KernelMatrix(C,C);
    T = chol(D*KMM*D + eps*M*eye(M));
    R = chol(T*T'/M + lambda*eye(M));

        function w = KnMtimesVector(u, v)
            w = zeros(M,1); ms = ceil(linspace(0, n, ceil(n/M)+1));
            for i=1:ceil(n/M)
                Kr = KernelMatrix( X(ms(i)+1:ms(i+1),:), C );
                w = w + Kr'*(Kr*u + v(ms(i)+1:ms(i+1),:));
            end
        end

        function w = BHB(u)
            w = R'\(T'\(KnMtimesVector(T\(R\u), zeros(n,1))/n) + lambda*(R\u));
        end

    r = R'\(T'\KnMtimesVector(zeros(M,1), Y/n));

    beta = conjgrad(@BHB, r, t);
    alpha = T\(R\beta);
end

function beta = conjgrad(funA, r, tmax)
    p = r; rsold = r'*r; beta = zeros(size(r,1), 1);

    for i = 1:tmax
        Ap = funA(p);
        a = rsold/(p'*Ap);
        beta = beta + a*p;
        r = r - a*Ap; rsnew = r'*r;
        p = r + (rsnew/rsold)*p;
        rsold = rsnew;
    end
end

function [C, D] = selectNystromCenters(X, lev_scores, M, n)
    if isempty(lev_scores) %Uniform Nystrom
        D = eye(M);
        C = X(randperm(n,M),:);
    else % Appr. Lev. Scores Nystrom
        prob = lev_scores(:)./sum(lev_scores(:));
        [count, ind] = discrete_prob_sample(M, prob);
        D = diag(1./sqrt(n*prob(ind).*count));
        C = X(ind,:);
    end
end

function [count, ind] = discrete_prob_sample(M, prob)
    bins = histcounts(rand(M,1), [0; cumsum(prob(:))]);
    ind = find(bins > 0);
    count = bins(ind);
end
```

Chapter 5

Fast and Accurate Leverage Score Sampling

In this chapter, we are going to expand some of the ideas we have seen in the previous chapter. In particular, we study how to derive fast and provably accurate algorithms for approximate leverage score sampling in the case of positive semi-definite matrices.

Leverage score sampling provides an appealing way to perform approximate computations for large matrices [AM15a]. Indeed, it allows deriving faithful approximations with a complexity adapted to the problem at hand. Yet, performing leverage scores sampling is a challenge in its own right requiring further approximations. The state of the art approximation of leverage score sampling requires $\mathcal{O}(n\widehat{\mathcal{N}}^2)$ time complexity, where $\widehat{\mathcal{N}}$ is the effective dimension of the problem. In this chapter, we first provide a novel algorithm for leverage score sampling that reduces the time complexity removing its linear dependence on n. We then exploit the proposed method to further speed up the FALKON algorithm, resulting in a learning pipeline with $\widetilde{\mathcal{O}}(n\widehat{\mathcal{N}})$ time and $\mathcal{O}(\widehat{\mathcal{N}}^2)$ memory complexity. In our theoretical analysis, we show that the proposed algorithms are currently the most efficient and accurate for solving these problems.

5.1 Leverage Score Sampling with BLESS

In the previous chapter in Section 4.3.2, we briefly introduce the concept of leverage scores and approximate leverage scores. In this section we first recall their definition more rigorously and focus on their computational aspects. We then state some previous algorithms for sampling according to leverage scores and present our approach and first theoretical results.

5.1.1 Leverage Score Sampling

Suppose $\widehat{K} \in \mathbb{R}^{n\times n}$ is symmetric and positive semidefinite. A basic question is deriving memory efficient approximation of $\widehat{K}$ [WS01, CLV17a] or related quantities, e.g. approximate projections on its range [MM17], or associated estimators, as in kernel ridge regression [RCR15, RCR17]. The eigendecomposition of $\widehat{K}$ offers a natural, but computationally demanding solution. Subsampling columns (or rows) is an appealing alternative. A basic approach is uniform sampling, whereas a more refined approach is leverage scores sampling. This latter procedure corresponds to sampling columns with probabilities proportional to the leverage scores

$$\ell(i,\lambda) = \left(\widehat{K}(\widehat{K}+\lambda n I)^{-1}\right)_{ii}, \qquad i \in [n], \tag{5.1}$$

where $[n] = \{1,\dots,n\}$. The advantage of leverage score sampling, is that potentially very few columns can suffice for the desired approximation. Indeed, letting

$$\widehat{\mathcal{N}}_\infty(\lambda) = n \max_{i=1,\dots,n} \ell(i,\lambda), \qquad \widehat{\mathcal{N}}(\lambda) = \sum_{i=1}^{n} \ell(i,\lambda),$$

for $\lambda > 0$, it is easy to see that $\widehat{\mathcal{N}}(\lambda) \le \widehat{\mathcal{N}}_\infty(\lambda) \le 1/\lambda$ for all λ, and previous results show that the number of columns required for accurate approximation are $\widehat{\mathcal{N}}_\infty$ for uniform sampling and $\widehat{\mathcal{N}}$ for leverage score sampling [Bac13, AM15a]. However, it is clear from definition (5.1) that an exact leverage scores computation would require the same order of computations as an eigendecomposition, hence approximations are needed. The accuracy of approximate leverage scores is typically measured by $t > 0$ in multiplicative bounds of the form

$$\frac{1}{1+t}\ell(i,\lambda) \le \widetilde{\ell}(i,\lambda) \le (1+t)\ell(i,\lambda), \quad \forall i \in [n]. \tag{5.2}$$

Before proposing a new improved solution, we briefly discuss relevant previous works. To provide a unified view, some preliminary discussion is useful.

5.1.2 Approximate Leverage Scores

First, we recall how a subset of columns can be used to compute approximate leverage scores. For $M \le n$, let $J = \{j_i\}_{i=1}^M$ with $j_i \in [n]$, and $\widehat{K}_{J,J} \in \mathbb{R}^{M\times M}$ with entries $(\widehat{K}_{J,J})_{lm} = \widehat{K}_{j_l,j_m}$. For $i \in [n]$, let $\widehat{K}_{J,i} = (\widehat{K}_{j_1,i},\dots,\widehat{K}_{j_M,i})$ and consider for $\lambda > 1/n$,

$$\widetilde{\ell}_J(i,\lambda) = (\lambda n)^{-1}(\widehat{K}_{i,i} - \widehat{K}_{J,i}^\top(\widehat{K}_{J,J}+\lambda n A)^{-1}\widehat{K}_{J,i}), \tag{5.3}$$

where $A \in \mathbb{R}^{M\times M}$ is a matrix to be specified [1] (see later for details). The above definition is motivated by the observation that if $J = [n]$, and $A = I$, then $\widetilde{\ell}_J(i,\lambda) = \ell(i,\lambda)$, by the following

[1] Clearly, $\widetilde{\ell}_J$ depends on the choice of the matrix A, but we omit this dependence to simplify the notation.

identity

$$\widehat{K}(\widehat{K}+\lambda nI)^{-1} = (\lambda n)^{-1}(\widehat{K} - \widehat{K}(\widehat{K}+\lambda nI)^{-1}\widehat{K}).$$

In the following, it is also useful to consider a subset of leverage scores computed as in (5.3). For $M \leq R \leq n$, let $U = \{u_i\}_{i=1}^{R}$ with $u_i \in [n]$, and

$$L_J(U,\lambda) = \{\widetilde{\ell}_J(u_1,\lambda),\ldots,\widetilde{\ell}_J(u_R,\lambda)\}. \tag{5.4}$$

Also in the following we will use the notation

$$L_J(U,\lambda) \mapsto J' \tag{5.5}$$

to indicate the leverage score sampling of $J' \subset U$ columns based on the leverage scores $L_J(U,\lambda)$, that is the procedure of sampling columns from U according to their leverage scores 5.1, computed using J, to obtain a new subset of columns J'.
We end noting that leverage score sampling (5.5) requires $\mathcal{O}(M^2)$ memory to store $\widehat{K}_{J,J}$, and $\mathcal{O}(M^3 + RM^2)$ time to invert $\widehat{K}_{J,J}$, and compute R leverage scores via (5.3).

5.1.3 Previous Algorithms for Leverage Scores Computations

We discuss relevant previous approaches using the above quantities.

TWO-PASS sampling [AM15a]. This is the first approximate leverage score sampling proposed, and is based on using directly (5.5) as $L_{J_1}(U_2,\lambda) \mapsto J_2$, with $U_2 = [n]$ and J_1 a subset taken uniformly at random. Here we call this method TWO-PASS sampling since it requires two rounds of sampling on the whole set $[n]$, one uniform to select J_1 and one using leverage scores to select J_2.

RECURSIVE-RLS [MM17]. This is a development of TWO-PASS sampling based on the idea of recursing the above construction. In our notation, let $U_1 \subset U_2 \subset U_3 = [n]$, where U_1, U_2 are uniformly sampled and have cardinalities $n/4$ and $n/2$, respectively. The idea is to start from $J_1 = U_1$, and consider first

$$L_{J_1}(U_2,\lambda) \mapsto J_2,$$

but then continue with

$$L_{J_2}(U_3,\lambda) \mapsto J_3.$$

Indeed, the above construction can be made recursive for a family of nested subsets $(U_h)_{h=1}^{H}$ of cardinalities $n/2^h$, considering $J_1 = U_1$ and

$$L_{J_h}(U_{h+1},\lambda) \mapsto J_{h+1}. \tag{5.6}$$

SQUEAK [CLV17a]. This approach follows a different iterative strategy. Consider a partition U_1, U_2, U_3 of $[n]$, so that $U_j = n/3$, for $j = 1,\ldots 3$. Then, consider $J_1 = U_1$, and

$$L_{J_1 \cup U_2}(J_1 \cup U_2,\lambda) \mapsto J_2,$$

and then continue with

$$L_{J_2 \cup U_3}(J_2 \cup U_3, \lambda) \mapsto J_3.$$

Similarly to the other cases, the procedure is iterated considering H subsets $(U_h)_{h=1}^H$ each with cardinality n/H. Starting from $J_1 = U_1$ the iterations is

$$L_{J_h \cup U_{h+1}}(J_h \cup U_{h+1}, \lambda). \tag{5.7}$$

We note that all the above procedures require specifying the number of iteration to be performed, the weights matrix to compute the leverage scores at each iteration, and a strategy to select the subsets $(U_h)_{h=1}^H$. In all the above cases the selection of U_h is based on uniform sampling, while the number of iterations and weight choices arise from theoretical considerations (see [AM15a, CLV17a, MM17] for details).

Note that TWO-PASS SAMPLING uses a set J_1 of cardinality roughly $1/\lambda$ (an upper bound on $\widehat{\mathcal{N}}_\infty(\lambda)$) and incurs in a computational cost of $RM^2 = n/\lambda^2$. In comparison, RECURSIVE-RLS [MM17] leads to essentially the same accuracy while improving computations. In particular, the sets J_h are never larger than $\widehat{\mathcal{N}}(\lambda)$. Taking into account that at the last iteration performs leverage score sampling on $U_h = [n]$, the total computational complexity is $n\widehat{\mathcal{N}}(\lambda)^2$. SQUEAK [CLV17a] recovers the same accuracy, size of J_h, and $n\widehat{\mathcal{N}}(\lambda)^2$ time complexity when $|U_h| \simeq \widehat{\mathcal{N}}(\lambda)$, but only requires a single pass over the data. We also note that a distributed version of SQUEAK is discussed in [CLV17a], which allows to reduce the computational cost to $n\widehat{\mathcal{N}}(\lambda)^2/p$, provided p machines are available.

5.1.4 Bottom-up Leverage Score Sampling with BLESS

The procedure we propose, dubbed BLESS, has similarities to the one proposed in [MM17] (see (5.6)), but also some important differences. The main difference is that, rather than a fixed λ, we consider a decreasing sequence of parameters $\lambda_0 > \lambda_1 > \cdots > \lambda_H = \lambda$ resulting in different algorithmic choices. For the construction of the subsets U_h we do not use nested subsets, but rather each $(U_h)_{h=1}^H$ is sampled uniformly and independently, with a size smoothly increasing as $1/\lambda_h$. Similarly, as in [MM17] we proceed iteratively, but at each iteration a different decreasing parameter λ_h is used to compute the leverage scores. Using the notation introduced above, the iteration of BLESS is given by

$$L_{J_h}(U_{h+1}, \lambda_{h+1}) \mapsto J_{h+1}, \tag{5.8}$$

where the initial set $J_1 = U_1$ is sampled uniformly with size roughly $1/\lambda_0$.
BLESS has two main advantages. The first is computational: each of the sets U_h, including the final U_H, has cardinality smaller than $1/\lambda$. Therefore, denoting with R_H the cardinality of U_H, the overall runtime has a cost of only $R_H M^2 \leq M^2/\lambda$, which can be dramatically smaller than

Algorithm 3: Bottom-up Leverage Scores Sampling (BLESS)

Input: dataset $\{x_i\}_{i=1}^n$, regularization λ, step q, starting reg. λ_0, constants q_1, q_2 controlling the approximation level.

Output: $M_h \in [n]$ number of selected points, J_h set of indexes, A_h weights.

$J_0 = \emptyset,\ A_0 = [],\ H = \frac{\log(\lambda_0/\lambda)}{\log q}$

for $h = 1 \dots H$ **do**

  $\lambda_h = \lambda_{h-1}/q$

  set constant $R_h = q_1 \min\{\kappa^2/\lambda_h,\ n\}$

  sample $U_h = \{u_1, \dots, u_{R_h}\}$ i.i.d. $u_i \sim Uniform([n])$

  compute $\widetilde{\ell}_{J_{h-1}}(x_{u_k}, \lambda_h)$ for all $u_k \in U_h$ using Eq. (5.3)

  set $P_h = (p_{h,k})_{k=1}^{R_h}$ with $p_{h,k} = \widetilde{\ell}_{J_{h-1}}(x_{u_k}, \lambda_h)/(\sum_{u \in U_h} \widetilde{\ell}_{J_{h-1}}(x_u, \lambda_h))$

  set constant $M_h = q_2 d_h$ with $d_h = \frac{n}{R_h} \sum_{u \in U_h} \widetilde{\ell}_{J_{h-1}}(x_u, \lambda_h)$, and

  sample $J_h = \{j_1, \dots, j_{M_h}\}$ i.i.d. $j_i \sim Multinomial(P_h, U_h)$

  $A_h = \frac{R_h M_h}{n} \text{diag}\left(p_{h,j_1}, \dots, p_{h,j_{M_h}}\right)$

end for

the nM^2 cost achieved by the methods in [MM17], [CLV17a] and is comparable to the distributed version of SQUEAK using $p = \lambda/n$ machines. The second advantage is that a whole *path* of leverage scores $\{\ell(i, \lambda_h)\}_{h=1}^H$ is computed at once, in the sense that at each iteration accurate approximate leverage scores at scale λ_h are computed. This is extremely useful in practice, as it can be used when cross-validating λ_h. As a comparison, for all previous method a full run of the algorithm is needed for each value of λ_h.

In this chapter we consider two variations of the above general idea leading to Algorithm 3 and Algorithm 4. The main difference in the two algorithms lies in the way in which sampling is performed: with and without replacement, respectively. In particular, considering sampling without replacement (see 4) it is possible to take the set $(U_h)_{h=1}^H$ to be nested and also to obtain slightly improved results, as shown in the next section.

The derivation of BLESS rests on some basic ideas. First, note that, since sampling uniformly a set U_λ of size $\widehat{\mathcal{N}}_\infty(\lambda) \leq 1/\lambda$ allows a good approximation, then we can replace $L_{[n]}([n], \lambda) \mapsto J$ by

$$L_{U_\lambda}(U_\lambda, \lambda) \mapsto J, \tag{5.9}$$

where J can be taken to have cardinality $\widehat{\mathcal{N}}(\lambda)$. However, this is still costly, and the idea is to repeat and couple approximations at multiple scales. Consider $\lambda' > \lambda$, a set $U_{\lambda'}$ of size $\widehat{\mathcal{N}}_\infty(\lambda') \leq 1/\lambda'$ sampled uniformly, and $L_{U_{\lambda'}}(U_{\lambda'}, \lambda') \mapsto J'$. The basic idea behind BLESS is to replace (5.9) by

$$L_{J'}(U_\lambda, \lambda) \mapsto \widetilde{J}.$$

Algorithm 4: Bottom-up Leverage Scores Sampling without Replacement (BLESS-R)

Input: dataset $\{x_i\}_{i=1}^n$, regularization λ, step q, starting reg. λ_0, constant q_2 controlling the approximation level.

Output: $M_h \in [n]$ number of selected points, J_h set of indexes, A_h weights.

$J_0 = \emptyset,\ A_0 = [],\ H = \frac{\log(\lambda_0/\lambda)}{\log q}$,
for $h = 1 \dots H$ **do**
  $\lambda_h = \lambda_{h-1}/q$
  set constant $\beta_h = \min\{q_2\kappa^2/(\lambda_h n),\ 1\}$
  initialize $U_h = \emptyset$
  for $i \in [n]$ **do**
    add i to U_h with probability β_h
  end for
  for $j \in U_h$ **do**
    compute $p_{h,j} = \min\{q_2 \widetilde{\ell}_{J_{h-1}}(x_j, \lambda_{h-1}),\ 1\}$
    add j to J_h with probability $p_{h,j}/\beta_h$
  end for
  $J_h = \{j_1, \dots, j_{M_h}\}$, and $A_h = \text{diag}\left(p_{h,j_1}, \dots, p_{h,j_{M_h}}\right)$.
end for

The key result is that taking $\widetilde{J}$ of cardinality

$$(\lambda'/\lambda)\widehat{\mathcal{N}}(\lambda) \tag{5.10}$$

suffice to achieve the same accuracy as J. Now, if we take λ' sufficiently large, it is easy to see that $\widehat{\mathcal{N}}(\lambda') \sim \widehat{\mathcal{N}}_\infty(\lambda') \sim 1/\lambda'$, so that we can take J' uniformly at random. However, the factor (λ'/λ) in (5.10) becomes too big. Taking multiple scales fix this problem and leads to the iteration in (5.8).

5.1.5 BLESS and BLESS-R in Details

BLESS (Algorithm 3). Here we describe our bottom-up algorithm in detail (see Algorithm 3). The central element is using a decreasing list of $\{\lambda_h\}_{h=1}^H$, from a given $\lambda_0 \gg \lambda$ up to λ. The idea is to iteratively construct a leverage score generator (LSG) set that approximates well the RLS for a given λ_h, based on the accurate RLS computed using a LSG set for λ_{h-1}. The crucial observation of the proposed algorithm is that when $\lambda_{h-1} \geq \lambda_h$ then

$$\forall i : \ell(i, \lambda_h) \leq \frac{\lambda_h}{\lambda_{h-1}} \ell(i, \lambda_{h-1}), \qquad \widehat{\mathcal{N}}(\lambda_h) \leq \frac{\lambda_h}{\lambda_{h-1}} \widehat{\mathcal{N}}(\lambda_{h-1}),$$

(see Lemma 24, for more details). By smoothly decreasing λ_h, the LSG at step h will only be a λ_h/λ_{h-1} factor worse than our previous estimate, which is automatically compensated by

a λ_h/λ_{h-1} increase in the size of the LSG. Therefore, to maintain an accuracy level for the leverage scores approximation as in Eq. (5.2) and small space complexity, it is sufficient to select a logaritmically spaced list of λ's from $\lambda_0 = \kappa^2$ to λ (see Theorem 12), in order to keep λ_h/λ_{h-1} as a small constant. This implies an extra multiplicative computational cost for the whole algorithm of only $\log(\kappa^2/\lambda)$.

More in detail, we initialize the Algorithm setting $D_0 = (\emptyset, [])$ to the empty LSG. Afterwards, we begin our main loop where at every step we reduce λ_h by a q factor, and then use D_{h-1} to construct a new LSG D_h. Note that at each iteration we construct a set J_h larger than J_{h-1}, which requires computing $\widetilde{\ell}_{D_{h-1}}(i, \lambda_h)$ for samples that are not in J_{h-1}, and therefore not computed at the previous step. Computing approximate leverage scores for the whole dataset would be highly inefficient, requiring $\mathcal{O}(nM_h^2)$ time which makes it unfeasible for large n. Instead, we show that to achieve the desired accuracy it is sufficient to restrict all our operations to a sufficiently large intermediate subset U_h sampled uniformly from $[n]$. After computing $\widetilde{\ell}_{D_{h-1}}(i, \lambda_h)$ only for points in U_h, we select M_h points with replacements according to their RLS to generate J_h. With a similar procedure we update the weights in A_h. We will see in Theorem 12, $|U_h| \propto 1/\lambda_h$ is sufficient to guarantee that this intermediate step produces a set satisfying Equation (5.2), and also takes care of increasing $|U_h|$ to increase accuracy as λ_h decreases. Moreover the algorithm uses a $M_h \propto \sum_{u \in U_h} \widetilde{\ell}_{D_{h-1}}(i, \lambda_h)$ that we prove in Theorem 12, to be in the order of $\widehat{\mathcal{N}}(\lambda_h)$. In the end, we return either the final LSG D_H to compute approximations of $\ell(i, \lambda)$, or any of the intermediate D_h if we are interested in the RLSs along the regularization path $\{\lambda_h\}_{h=1}^H$.

BLESS-R (Algorithm 4) The second algorithm we propose, is based on the same principles of Algorithm 3, while simplifying some steps of the procedure. In particular it removes the need to explicitly track the normalization constant d_h and the intermediate uniform sampling set, by replacing it with *rejection* sampling. At each iteration $h \in [H]$, instead of drawing the set U_h from a uniform distribution, and then sampling J_h, from U_h, Algorithm 4 performs a single round of rejection sampling for each column according to the following identity

$$\mathbb{P}(z_{h,i} = 1) = \mathbb{P}(z_{h,i} = 1 | u_{h,i} \leq \beta_h)\mathbb{P}(u_{h,i} \leq \beta_h) = \beta_h p_{h,i}/\beta_h = p_{h,i} \propto \widetilde{\ell}_{D_{h-1}}(x_i, \lambda_{h-1}),$$

where $z_{h,i}$ is the r.v. which is 1 if $i \in [n]$, while $u_{h,i}$ is the probability that the column i passed the rejection sampling step, while β_h a suitable treshold which mimik the effect of the set U_h.

Space and time complexity. Note that at each iteration constructing the generator $\widetilde{\ell}_{D_{h-1}}$, requires computing the inverse $(K_{J_h} + \lambda_h n I)^{-1}$, with M_h^3 time complexity, while each of the R_h evaluations $\widetilde{\ell}_{D_{h-1}}(i, \lambda_h)$ takes only M_h^2 time. Summing over the H iterations Algorithm 3 runs in $\mathcal{O}(\sum_{h=1}^H M_h^3 + R_h M_h^2)$ time. Noting that $R_h \simeq 1/\lambda_h$, that $M_h \simeq d_h \leq 1/\lambda_h$, and that $\sum_h \lambda_h^{-1} = \sum_h q^{h-H}\lambda^{-1} = \frac{q-q^{-H}}{q-1}\lambda^{-1}$, the final cost is $\mathcal{O}\left(\lambda^{-1} \max_h M_h^2\right)$ time, and $\mathcal{O}\left(\max_h M_h^2\right)$ space. Similarly, Algorithm 4 only evaluates $\widetilde{\ell}_{D_{h-1}}$ for the points that pass the rejection steps which w.h.p. happens only $\mathcal{O}(n\beta_h) = \mathcal{O}(1/\lambda)$ times, so we have the same time and space complexity of Algorithm 3.

5.1.6 Theoretical Guarantees

Our first main result establishes in a precise and quantitative way the advantages of BLESS.

Theorem 12. *Let $n \in \mathbb{N}$, $\lambda > 0$ and $\delta \in (0, 1]$. Given $t > 0, q > 1$ and $H \in \mathbb{N}$, $(\lambda_h)_{h=1}^H$ defined as in Algorithms 3,4, when $(J_h, A_h)_{h=1}^H$ are computed*

1. *by Algorithm 3 with parameters $\lambda_0 = \frac{\kappa^2}{\min(t,1)}$, $q_1 \geq \frac{5\kappa^2 q_2}{q(1+t)}$, $q_2 \geq 12q\frac{(2t+1)^2}{t^2}(1+t)\log\frac{12Hn}{\delta}$,*
2. *by Algorithm 4 with parameters $\lambda_0 = \frac{\kappa^2}{\min(t,1)}$, $q_1 \geq 54\kappa^2\frac{(2t+1)^2}{t^2}\log\frac{12Hn}{\delta}$,*

let $\widetilde{\ell}_{J_h}(i, \lambda_h)$ as in Eq. (5.3) depending on J_h, A_h, then with probability at least $1 - \delta$:

(a) $$\frac{1}{1+t}\ell(i, \lambda_h) \;\leq\; \widetilde{\ell}_{J_h}(i, \lambda_h) \;\leq\; (1 + \min(t, 1))\ell(i, \lambda_h), \quad \forall i \in [n], h \in [H],$$

(b) $$|J_h| \leq q_2\widehat{\mathcal{N}}(\lambda_h), \quad \forall h \in [H].$$

The above result confirms that the subsets J_h computed by BLESS are accurate in the desired sense, see (5.2), and the size of all J_h is small and proportional to $\widehat{\mathcal{N}}(\lambda_h)$, leading to a computational cost of only $\mathcal{O}\left(\min\left(\frac{1}{\lambda}, n\right)\widehat{\mathcal{N}}(\lambda)^2\log^2\frac{1}{\lambda}\right)$ in time and $O\left(\widehat{\mathcal{N}}(\lambda)^2\log^2\frac{1}{\lambda}\right)$ in space (for additional properties of J_h see Theorem 14 in Section 5.2.5). Table 5.1 compares the complexity and number of columns sampled by BLESS with other methods. The crucial point is that in most applications, the parameter λ is chosen as a decreasing function of n, e.g. $\lambda = 1/\sqrt{n}$, resulting in potentially massive computational gains. Indeed, since BLESS computes leverage scores for sets of size at most $1/\lambda$, this allows to perform leverage scores sampling on matrices with millions of rows/columns, as shown in the experiments. In the next section, we illustrate the impact of BLESS in the context of supervised statistical learning.

5.2 Theoretical Analysis for BLESS

In this section, Theorem 14 and Theorem 15 provide guarantees for the two methods, from which Theorem 12 is derived. In particular in Section 5.2.4 some important properties about (out-of-sample) leverage scores, that will be used in the proofs, are derived.

We now present notation, definitions and some preliminary results necessary to proof the main results.

Algorithm	Runtime	$\lvert J\rvert$
Uniform Sampling [Bac13]	$-$	$1/\lambda$
Exact RLS Sampl.	n^3	$\widehat{\mathcal{N}}(\lambda)$
Two-Pass Sampling [AM15a]	n/λ^2	$\widehat{\mathcal{N}}(\lambda)$
Recursive RLS [MM17]	$n\widehat{\mathcal{N}}(\lambda)^2$	$\widehat{\mathcal{N}}(\lambda)$
SQUEAK [CLV17a]	$n\widehat{\mathcal{N}}(\lambda)^2$	$\widehat{\mathcal{N}}(\lambda)$
BLESS/BLESS-R(Alg. 3 and 4)	$\boldsymbol{1/\lambda\,\widehat{\mathcal{N}}(\lambda)^2}$	$\widehat{\mathcal{N}}(\lambda)$

Table 5.1: The proposed algorithms are compared with the state of the art (in $\widetilde{\mathcal{O}}$ notation), in terms of time complexity and cardinality of the set J required to satisfy the approximation condition in Eq. (5.2).

5.2.1 Notation

Let $\mathcal{X}$ be a Polish space and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive semidefinite function on $\mathcal{X}$, we denote $\mathcal{H}$ the Hilbert space obtained by the completion of

$$\mathcal{H} = \overline{\text{span}\{k(x,\cdot) \mid x \in \mathcal{X}\}}$$

according to the norm induced by the inner product $\langle k(x,\cdot), k(x',\cdot)\rangle_{\mathcal{H}} = k(x,x')$. Spaces $\mathcal{H}$ constructed in this way are known as *reproducing kernel Hilbert spaces* and there is a one-to-one relation between a kernel k and its associated RKHS. For more details on RKHS we refer the reader to [Aro50, SC08]. Given a kernel k, in the following we will denote with $k_x = k(x,\cdot) \in \mathcal{H}$ for all $x \in \mathcal{X}$. We say that a kernel is bounded if $\|k_x\|_{\mathcal{H}} \leq \kappa$ with $\kappa > 0$. In the following we will always assume k to be continuous and bounded by $\kappa > 0$. The continuity of k with the fact that $\mathcal{X}$ is Polish implies $\mathcal{H}$ to be separable [SC08].

In the rest of the Chapter we denote with A_λ, the operator $A + \lambda I$, for any symmetric linear operator A, $\lambda \in \mathbb{R}$ and I the identity operator.

5.2.2 Definitions

For $n \in \mathbb{N}$, $(x_i)_{i=1}^n$, and $J \subseteq \{1,\dots,n\}$, $A \in \mathbb{R}^{|J|\times|J|}$ diagonal matrix with positive diagonal, denote $\widetilde{\ell}_J$ in eq. (5.3) by showing the dependence from both J and A as

$$\widetilde{\ell}_{J,A}(i,\lambda) = (\lambda n)^{-1}(\widehat{K}_{i,i} - \widehat{K}_{J,i}^\top(\widehat{K}_{J,J} + \lambda n A)^{-1}\widehat{K}_{J,i}). \tag{5.11}$$

Moreover let $J = \{j_1,\dots,j_M\}$, define $\widehat{C}_{J,A}$ as

$$\widehat{C}_{J,A} = \frac{1}{|J|}\sum_{i=1}^{|J|} A_{ii}^{-1} k_{x_{j_i}} \otimes k_{x_{j_i}},$$

and define $\widehat{C}_n$ as

$$\widehat{C}_n = \frac{1}{n}\sum_{i=1}^{n} k_{x_i} \otimes k_{x_i}.$$

We now define the *out-of-sample leverage scores*, that are an extension of $\widetilde{\ell}_{J,A}$ to any point x in the space $\mathcal{X}$.

Definition 8 (out-of-sample leverage scores)**.** *Let $J = \{j_1, \dots, j_M\} \subseteq \{1, \dots, n\}$, with $M \in \mathbb{N}$ and $A \in \mathbb{R}^{M\times M}$ be a positive diagonal matrix. Then for any $x \in \mathcal{X}$ and $\lambda > 0$ we define*

$$\widehat{\ell}_{J,A}(x,\lambda) = \frac{1}{n}\|(\widehat{C}_{J,A} + \lambda I)^{-1/2} k_x\|^2_{\mathcal{H}}.$$

Moreover define $\widehat{\ell}_{\emptyset,[]}(x,\lambda) = (\lambda n)^{-1} k(x,x)$.

In particular we denote by

$$\widehat{\ell}(x,\lambda) = \widehat{\ell}_{[n],I}(x,\lambda),$$

the out-of-sample version of the exact leverage scores $\ell(i,\lambda)$. Indeed note that $\widehat{\ell}(x_i,\lambda) = \ell(i,\lambda)$ for $i \in [n]$ and $\lambda > 0$ as proven by the next proposition that shows, more generally, the relation between $\widehat{\ell}_{J,A}$ and $\widetilde{\ell}_{J,A}$.

Proposition 9. *Let $n \in \mathbb{N}$, $(x_i)_{i=1}^n \subseteq X$. For any $\lambda > 0, J \subseteq \{1, \dots, n\}, A \in \mathbb{R}^{|J|\times|J|}$ with A positive diagonal, we that that for any $x \in \mathcal{X}$, $\widehat{\ell}_{J,A}(x,\lambda)$ in Def. 8 and $\widetilde{\ell}_{J,A}(x,\lambda)$ in Def. 5.3, satisfy*

$$\widehat{\ell}_{J,\frac{n}{|J|}A}(x_i,\lambda) = \widetilde{\ell}_{J,A}(i,\lambda),$$

when $|J| > 0$, and $\widehat{\ell}_{\emptyset,[]}(x_i,\lambda) = \widetilde{\ell}_{\emptyset,[]}(i,\lambda)$, when $|J| = 0$, for any $i \in [n], \lambda > 0$.

Proof. Let $J = \{j_1, \dots, j_{|J|}\}$. We will first show that $\widehat{\ell}_{J,A}(x,\lambda)$ is characterized by,

$$\widehat{\ell}_{J,A}(x,\lambda) = \frac{1}{\lambda n}k(x,x) - \frac{1}{\lambda n} v_J(x)^\top (K_J + \lambda |J| A)^{-1} v_J(x),$$

with $K_J \in \mathbb{R}^{M\times M}$ with $(K_J)_{lm} = k(x_{j_l}, x_{j_m})$ and $v_J(x) = (k(x,x_{j_1}), \dots, k(x,x_{j_M}))$. Denote with $Z_J : \mathcal{H} \to \mathbb{R}^{|J|}$, the linear operator defined by $Z_J = (k_{x_{j_1}}, \dots, k_{x_{j_{|J|}}})^\top$, that is $(Z_J f)_k = \langle k_{x_{j_k}}, f\rangle_{\mathcal{H}}$, for $f \in \mathcal{H}$ and $k \in \{1, \dots |J|\}$. Then, by denoting with $B = |J|A$ we have

$$Z_J^* B^{-1} Z_J = \frac{1}{|J|}\sum_{i=1}^{|J|} A_{ii}^{-1} k_{x_{j_i}} \otimes k_{x_{j_i}} = \widehat{C}_{J,A}.$$

Now note that, since $(Q+\lambda I)^{-1} = \lambda^{-1}(I - Q(Q+\lambda I)^{-1})$ for any positive linear operator and $\lambda > 0$, we have

$$\begin{aligned}\widehat{\ell}_{J,A}(x,\lambda) &= \frac{1}{n}\left\langle k_x, (\widehat{C}_{J,A}+\lambda I)^{-1}k_x\right\rangle_{\mathcal{H}} = \frac{1}{\lambda n}\left\langle k_x, (I - \widehat{C}_{J,A}(\widehat{C}_{J,A}+\lambda I)^{-1})k_x\right\rangle_{\mathcal{H}} \\ &= \frac{k(x,x)}{\lambda n} - \frac{1}{\lambda n}\left\langle k_x, Z_J^* B^{-1/2}(B^{-1/2}Z_J Z_J^* B^{-1/2} + \lambda I)^{-1} B^{-1/2} Z_J k_x\right\rangle_{\mathcal{H}},\end{aligned}$$

where in the last step we use the fact that $R^*R(R^*R+\lambda I)^{-1} = R^*(RR^*+\lambda I)^{-1}R$, for any bounded linear operator R and $\lambda > 0$. In particular we used it with $R = B^{-1/2}Z_J$. Now note that $Z_J Z_J^* \in \mathbb{R}^{|J|\times|J|}$ and in particular $Z_J Z_J^* = K_J$, moreover $Z_J k_x = v(x)$, so

$$\begin{aligned}\widehat{\ell}_{J,A}(x,\lambda) &= \frac{k(x,x)}{\lambda n} - \frac{1}{\lambda n} v(x)^\top B^{-1/2}(B^{-1/2}K_J B^{-1/2} + \lambda I)^{-1} B^{-1/2} v(x) \\ &= \frac{k(x,x)}{\lambda n} - \frac{1}{\lambda n} v(x)^\top (K_J + \lambda B)^{-1} v(x) \\ &= \frac{k(x,x)}{\lambda n} - \frac{1}{\lambda n} v(x)^\top (K_J + \lambda |J| A)^{-1} v(x),\end{aligned}$$

where in the second step we used the fact that $B^{-1/2}(B^{-1/2}QB^{-1/2}+\lambda I)^{-1}B^{-1/2} = (Q+\lambda B)^{-1}$, for any invertible B any positive operator Q and $\lambda > 0$.

Finally note that

$$\widehat{\ell}_{J,\frac{n}{|J|}A}(x_i,\lambda) = \frac{k(x,x)}{\lambda n} - \frac{1}{\lambda n} v(x)^\top (K_J + \lambda n A)^{-1} v(x) = \widetilde{\ell}_{J,A}(i,\lambda).$$

□

5.2.3 Preliminary Results

Recall that an operator A is said to be positive if $\langle v, Av\rangle \geq 0\ \forall v$. Denote with $G_\lambda(A,B)$ the quantity

$$G_\lambda(A,B) = \|(A+\lambda I)^{-1/2}(A-B)(A+\lambda I)^{-1/2}\|,$$

for A, B positive bounded linear operators and for $\lambda > 0$.

Proposition 10. *Let A, B be positive bounded linear operators and $\lambda > 0$, then*

$$\|I - (A+\lambda I)^{-1/2}(B+\lambda I)(A+\lambda I)^{-1/2}\| = G_\lambda(A,B) \leq \frac{G_\lambda(B,A)}{1 - G_\lambda(B,A)},$$

where the last inequality holds if $G_\lambda(B,A) < 1$.

Proof. For the sake of compactness denote with A_λ the operator $A+\lambda I$ and with B_λ the operator $B+\lambda I$. First of all note that $I = A_\lambda^{-1/2} A_\lambda A_\lambda^{-1/2}$, so

$$\begin{aligned} I - A_\lambda^{-1/2} B_\lambda A_\lambda^{-1/2} &= A_\lambda^{-1/2} A_\lambda A_\lambda^{-1/2} - A_\lambda^{-1/2} B_\lambda A_\lambda^{-1/2} \\ &= A_\lambda^{-1/2}(A_\lambda - B_\lambda) A_\lambda^{-1/2} = A_\lambda^{-1/2}(A-B) A_\lambda^{-1/2} \\ &= A_\lambda^{-1/2} B_\lambda^{1/2} \;\; B_\lambda^{-1/2}(A-B) B_\lambda^{-1/2} \;\; B_\lambda^{1/2} A_\lambda^{-1/2}, \end{aligned}$$

where in the last step we multiplied and divided by $B_\lambda^{1/2}$. Then

$$\left\| I - A_\lambda^{-1/2} B_\lambda A_\lambda^{-1/2} \right\| \leq \|A_\lambda^{-1/2} B_\lambda^{1/2}\|^2 \|B_\lambda^{-1/2}(A-B) B_\lambda^{-1/2}\|,$$

moreover, by Prop. 7 of [RCR15] (see also Prop. 8 of [RR17]), if $G_\lambda(B, A) < 1$, we have

$$\|A_\lambda^{-1/2} B_\lambda^{1/2}\|^2 \leq (1 - G_\lambda(B, A))^{-1}.$$

□

Proposition 11. *Let A, B, C be bounded positive linear operators on a Hilbert space. Let $\lambda > 0$. Then, the following holds*

$$G_\lambda(A, C) \leq G_\lambda(A, B) + (1 + G_\lambda(A, B)) G_\lambda(B, C).$$

Proof. In the following we denote with A_λ the operator $A + \lambda I$ and the same for B, C. Then

$$\|A_\lambda^{-1/2}(A-C)A_\lambda^{-1/2}\| \leq \|A_\lambda^{-1/2}(A-B)A_\lambda^{-1/2}\| + \|A_\lambda^{-1/2}(B-C)A_\lambda^{-1/2}\|.$$

Now note that, by dividing and multiplying for $B_\lambda^{1/2}$, we have

$$\begin{aligned} \|A_\lambda^{-1/2}(B-C)A_\lambda^{-1/2}\| &= \|A_\lambda^{-1/2} B_\lambda^{1/2} \; B_\lambda^{-1/2}(B-C) B_\lambda^{-1/2} B_\lambda^{1/2} A_\lambda^{-1/2}\| \\ &\leq \|A_\lambda^{-1/2} B_\lambda^{1/2}\|^2 \|B_\lambda^{-1/2}(B-C)B_\lambda^{-1/2}\| = \|A_\lambda^{-1/2} B_\lambda^{1/2}\|^2 G_\lambda(B, C). \end{aligned}$$

Finally note that, since $\|Z\|^2 = \|Z^* Z\|$ for any bounded linear operator Z, we have

$$\|A_\lambda^{-1/2} B_\lambda^{1/2}\|^2 = \|A_\lambda^{-1/2} B_\lambda A_\lambda^{-1/2}\| = \|I + (I - A_\lambda^{-1/2} B_\lambda A_\lambda^{-1/2})\| \leq 1 + \|I - A_\lambda^{-1/2} B_\lambda A_\lambda^{-1/2}\|.$$

Moreover, by Prop. 10, we have that

$$\|I - A_\lambda^{-1/2} B_\lambda A_\lambda^{-1/2}\| = G_\lambda(A, B).$$

□

Proposition 12. *Let B be a bounded linear operator, then*

$$1 - \|I - BB^*\| \leq \sigma_{\min}(B)^2 \leq \sigma_{\max}(B)^2 \leq 1 + \|I - BB^*\|.$$

Proof. Now we recall that, denoting by $\preceq$ the Lowner partial order, for a positive bounded operator A such that $aI \preceq A \preceq bI$ for $0 \leq a \leq b$, we have $(1-b)I \preceq I - A \preceq (1-a)I \preceq (1+b)I$ and so, since $BB^* = I - (I - BB^*)$, we have

$$(1 - \|I - BB^*\|)I \preceq \sigma_{\min}(B)^2 I \preceq BB^* \preceq \sigma_{\max}(B)^2 I \preceq 1 + (1 + \|I - BB^*\|)I,$$

from we have the desired result. □

Let $\|\cdot\|_{HS}$ denote the Hilbert-Schmidt norm.

We recall and adapt to our needs a result from Prop. 8 of [RCR15].

Proposition 13. *Let $\lambda > 0$ and $v_1, \ldots, v_n$ with $n \geq 1$, be identically distributed random vectors on separable Hilbert space $\mathcal{H}$, such that there exists $\kappa^2 > 0$ for which $\|v\|_{\mathcal{H}} \leq \kappa^2$ almost surely. Denote by Q the Hermitian operator $Q = \frac{1}{n}\sum_{i=1}^n \mathbb{E}[v_i \otimes v_i]$. Let $Q_n = \frac{1}{n}\sum_{i=1}^n v_i \otimes v_i$. Then for any $\delta \in (0, 1]$, the following holds*

$$\left\|(Q + \lambda I)^{-1/2}(Q - Q_n)(Q + \lambda I)^{-1/2}\right\| \leq \frac{4\kappa^2\beta}{3\lambda n} + \sqrt{\frac{2\kappa^2\beta}{\lambda n}}$$

with probability $1 - \delta$ and $\beta = \log \frac{4\operatorname{Tr}(Q(Q+\lambda I)^{-1})}{\|Q(Q+\lambda I)^{-1}\|\delta} \leq \frac{8\kappa^2(1+\operatorname{Tr}(Q_\lambda^{-1}Q))}{\|Q\|\delta}$.

Proof. Let $Q_\lambda = Q + \lambda I$. Here we apply non-commutative Bernstein inequality like [Tro12] (with the extension to separable Hilbert spaces as in[RCR15], Prop. 12) on the random variables $Z_i = M - Q_\lambda^{-1/2}v_i \otimes Q_\lambda^{-1/2}v_i$ with $M_i = Q_\lambda^{-1/2}(\mathbb{E}[v_i \otimes v_i])Q_\lambda^{-1/2}$ for $1 \leq i \leq n$. Note that the expectation of Z_i is 0. The random vectors are bounded by

$$\begin{aligned}\|Q_\lambda^{-1/2}v_i \otimes Q_\lambda^{-1/2}v_i - M_i\| &= \|\mathbb{E}_{v_i'}[Q_\lambda^{-1/2}v_i' \otimes Q_\lambda^{-1/2}v_i' - Q_\lambda^{-1/2}v_i \otimes Q_\lambda^{-1/2}v_i]\|_{\mathcal{H}} \\ &\leq 2\|\kappa^2\|\|(Q+\lambda)^{-1/2}\|^2 \leq \frac{2\kappa^2}{\lambda},\end{aligned}$$

and the second orded moment is

$$\begin{aligned}\mathbb{E}(Z_i)^2 = \mathbb{E}\ \ &\left\langle v_i, Q_\lambda^{-1}v_i\right\rangle\ Q_\lambda^{-1/2}v_i \otimes Q_\lambda^{-1/2}v_i \quad - \quad Q_\lambda^{-2}Q^2 \\ &\leq \frac{\kappa^2}{\lambda}\mathbb{E}[Q_\lambda^{-1/2}v_1 \otimes Q_\lambda^{-1/2}v_1] = \frac{\kappa^2}{\lambda}Q(Q+\lambda I)^{-1} =: S.\end{aligned}$$

Now we can apply the Bernstein inequality with *intrinsic dimension* in [Tro12] (or Prop. 12 in [RCR15]). Now some considerations on β. It is $\beta = \log \frac{4\operatorname{Tr} S}{\|S\|\delta} = \frac{4\operatorname{Tr} Q_\lambda^{-1}Q}{\left\|Q_\lambda^{-1}Q\right\|\delta}$, now we need a

lower bound for $\left\|Q_\lambda^{-1}Q\right\| = \frac{\sigma_1}{\sigma_1+\lambda}$ where $\sigma_1 = \|Q\|$ is the biggest eigenvalue of Q, now, when $0 < \lambda \leq \sigma_1$ we have $\beta \leq \frac{8\,\mathrm{Tr}\,Q}{\lambda\delta}$.

When $\lambda \geq \sigma_1$, note that $\mathrm{Tr}(Q(Q+\lambda I)^{-1}) \leq \lambda^{-1}\,\mathrm{Tr}(Q) \leq \kappa^2/\lambda$, then

$$\frac{\mathrm{Tr}(Q(Q+\lambda I)^{-1})}{\left\|Q_\lambda^{-1}Q\right\|} \leq \frac{\kappa^2}{\lambda\frac{\sigma_1}{\sigma_1+\lambda}} = \frac{\kappa^2}{\lambda} + \frac{\kappa^2}{\sigma_1} \leq \frac{2\kappa^2}{\sigma_1}.$$

So finally $\beta \leq \frac{8(\kappa^2/\|Q\|+\mathrm{Tr}(Q_\lambda^{-1}Q))}{\delta}$ □

5.2.4 Analytic Decomposition

In this section we control the out-of-sample leverage scores $\widehat{\ell}_{J,A}(x,\lambda)$ for a fixed λ, a generic set of indexes J and weights A, with respect to the out-of-sample version of the exact leverage scores $\widehat{\ell}(x,\lambda)$ (Theorem 13). Moreover, we introduce two technical Lemmas used to prove Theorem 13 and further theorems in the next sections. The first one (Lemma 23) relates for a fixed λ two out-of-sample leverage scores $\widehat{\ell}_{J,A}(x,\lambda)$, $\widehat{\ell}_{J',A'}(x,\lambda)$ of two generic pairs of indexes and weights J, A and J', A'. The second one (Lemma 24) relates, for a fixed J and A, the out-of-sample leverage scores $\widehat{\ell}_{J,A}(x,\lambda)$, $\widehat{\ell}_{J,A}(x,\lambda')$ for two different values λ, λ'.

Lemma 23. *Let $\lambda > 0$, $J, J' \subseteq \{1,\dots,n\}$, with $|J|, |J'| \geq 1$ and $A \in \mathbb{R}^{|J|\times|J|}$, $A' \in \mathbb{R}^{|J'|\times|J'|}$ positive diagonal matrices, then*

$$\frac{1-2\nu}{1-\nu}\widehat{\ell}_{J',A'}(x,\lambda) \leq \widehat{\ell}_{J,A}(x,\lambda) \leq \frac{1}{1-\nu}\widehat{\ell}_{J',A'}(x,\lambda), \quad \forall x \in \mathcal{X},$$

with $\nu = G_\lambda(\widehat{C}_{J',A'}, \widehat{C}_{J,A})$.

Proof. By denoting with B the operator

$$B = (\widehat{C}_{J,A} + \lambda I)^{-1/2}(\widehat{C}_{J',A'} + \lambda I)^{1/2},$$

and according to the characterization of $\widehat{\ell}_{J,A}(x,\lambda)$ via Prop. 9, we have

$$\widehat{\ell}_{J,A}(x,\lambda) = n^{-1}\left\|(\widehat{C}_{J,A} + \lambda I)^{-1/2}k_x\right\|_{\mathcal{H}}^2 = n^{-1}\left\|B\,(\widehat{C}_{J',A'} + \lambda I)^{-1/2}k_x\right\|_{\mathcal{H}}^2.$$

So, by recalling the fact that, by definition of Lowner partial order $\preceq$, we have $a\|v\|^2 \leq \|Av\|^2 \leq b\|v\|^2$, for any vector v and bounded linear operator such that $aI \preceq A^*A \preceq bI$ with $0 \leq a \leq b$, and the fact that $\sigma(A^*A) = \sigma(AA^*) = \sigma(A)^2$, we have

$$\sigma_{\min}(B)^2\left\|(\widehat{C}_{J',A'} + \lambda I)^{-1/2}k_x\right\|_{\mathcal{H}}^2 \leq \left\|B(\widehat{C}_{J',A'} + \lambda I)^{-1/2}k_x\right\|_{\mathcal{H}}^2 \leq \sigma_{\max}(B)^2\left\|(\widehat{C}_{J',A'} + \lambda I)^{-1/2}k_x\right\|_{\mathcal{H}}^2.$$

That, by Prop. 9, is equivalent to

$$\sigma_{\min}(B)^2\widehat{\ell}_{J',A'}(x,\lambda) \leq \widehat{\ell}_{J,A}(x,\lambda) \leq \sigma_{\max}(B)^2\widehat{\ell}_{J',A'}(x,\lambda).$$

By Prop. 12 we have $1 - \|I - BB^*\| \leq \sigma_{\min}(B)^2 \leq \sigma_{\max}(B)^2 \leq 1 + \|I - BB^*\|$. Finally, by Prop. 10, we have

$$\|I - BB^*\| \leq \frac{\nu}{1-\nu}.$$

□

Lemma 24. *Let $0 < \lambda \leq \lambda'$, and $J \subseteq \{1,\dots,n\}$ and $A \in \mathbb{R}^{|J|\times|J|}$, then*

$$\widehat{\ell}_{J,A}(x,\lambda') \leq \widehat{\ell}_{J,A}(x,\lambda) \leq \frac{\lambda'}{\lambda}\widehat{\ell}_{J,A}(x,\lambda'), \quad \forall x \in \mathcal{X}.$$

Proof. If $|J| = 0$ we have that $\widehat{\ell}_{\emptyset,[]}(x,\lambda) = \frac{k(x,x)}{\lambda n}$ and the desired result is easily verified. If $|J| \geq 1$, let $B = (C_{J,A} + \lambda I)^{-1/2}(C_{J,A} + \lambda' I)^{1/2}$. By recalling the fact that, by definition of Lowner partial order $\preceq$, we have $a\|v\|^2 \leq \|Av\|^2 \leq b\|v\|^2$, for any vector v and bounded linear operator such that $aI \preceq A^*A \preceq bI$ with $0 \leq a \leq b$, and the fact that $\sigma(A^*A) = \sigma(AA^*) = \sigma(A)^2$, we have

$$\sigma_{\min}(B)^2 \left\|(\widehat{C}_{J,A} + \lambda' I)^{-1/2}k_x\right\|_{\mathcal{H}}^2 \leq \left\|B(\widehat{C}_{J,A} + \lambda' I)^{-1/2}k_x\right\|_{\mathcal{H}}^2 \leq \sigma_{\max}(B)^2 \left\|(\widehat{C}_{J,A} + \lambda' I)^{-1/2}k_x\right\|_{\mathcal{H}}^2.$$

That, by Prop. 9, is equivalent to

$$\sigma_{\min}(B)^2\widehat{\ell}_{J,A}(x,\lambda') \leq \widehat{\ell}_{J,A}(x,\lambda) \leq \sigma_{\max}(B)^2\widehat{\ell}_{J,A}(x,\lambda').$$

Now note that

$$\sigma_{\min}(B)^2 \geq \inf_{\sigma\geq 0}\frac{\sigma+\lambda'}{\sigma+\lambda} = 1, \quad \sigma_{\max}(B)^2 \geq \sup_{\sigma\geq 0}\frac{\sigma+\lambda'}{\sigma+\lambda} = \frac{\lambda'}{\lambda}.$$

□

Theorem 13. *Let $\lambda > 0$, $J \subseteq \{1,\dots,n\}$, with $|J| \geq 1$ and $A \in \mathbb{R}^{|J|\times|J|}$ positive diagonal. Then the following hold for any $x \in \mathcal{X}$,*

$$\frac{1-2\nu_{J,A}}{1-\nu_{J,A}}\widehat{\ell}(x,\lambda) \leq \widehat{\ell}_{J,A}(x,\lambda) \leq \frac{1}{1-\nu_{J,A}}\widehat{\ell}(x,\lambda),$$

where $\nu_{J,A} = G_\lambda(\widehat{C}_n, \widehat{C}_{J,A})$. Morever note that for any $|U| \subseteq \{1,\dots,n\}$, we have

$$\nu_{J,A} \leq \eta_U + (1+\eta_U)\beta_{J,A,U},$$

with $\beta_{J,A,U} = G_\lambda(\widehat{C}_{U,I}, \widehat{C}_{J,A})$ and $\eta_U = G_\lambda(\widehat{C}_n, \widehat{C}_{U,I})$.

Proof. By applying Lemma 23, with their $J' = \{1, \dots, n\}$, $A' = I$, and recalling that $\widehat{\ell}(x, \lambda) = \widehat{\ell}_{\{1,\dots,n\},I}$, we have for all $x \in \mathcal{X}$

$$\frac{1 - 2\nu_{J,A}}{1 - \nu_{J,A}} \widehat{\ell}(x, \lambda) \leq \widehat{\ell}_{J,A}(x, \lambda) \leq \frac{1}{1 - \nu_{J,A}} \widehat{\ell}(x, \lambda).$$

To conclude the proof we bound $\nu_{J,A}$ in terms of $\beta_{J,A,U}$ and η_U, via Prop. 11. □

5.2.5 Proof for BLESS (Alg. 3)

This section presents three technical lemmas used to prove Theorem 14 that provides the guarantees for BLESS (Algorithm 3).

Lemma 25. *Let $n \in \mathbb{N}$, $(x_i)_{i=1}^n \subseteq \mathcal{X}$. Let $U \subseteq \{1, \dots n\}$, with $|U| \geq 1$. Let $(p_k)_{k=1}^{|U|} \subset \mathbb{R}$ be a non-negative sequence summing to 1. Let $M \in \mathbb{N}$ and $J = \{j_1, \dots, j_M\}$ with j_i sampled i.i.d. from $\{1, \dots, |U|\}$ with probability $(p_k)_{k=1}^{|U|}$ and $A = |U| diag(p_{j_1}, \dots, p_{j_M})$. Let $\tau \in (0, 1]$, and $s := \sup_{k \in \{1,\dots,|U|\}} \frac{1}{|U| p_k} \|(\widehat{C}_{U,I} + \lambda I)^{-1/2} k_{x_{u_k}}\|_{\mathcal{H}}^2$. When*

$$M \geq 2s \log \frac{4n}{\tau},$$

then the following holds with probability at least $1 - \tau$

$$\|(\widehat{C}_{U,I} + \lambda I)^{-1/2} (\widehat{C}_{J,A} - \widehat{C}_{U,I}) (\widehat{C}_{U,I} + \lambda I)^{-1/2}\| \leq \sqrt{\frac{4s \log \frac{4n}{\tau}}{M}}.$$

Proof. Denote with ζ_i the random variable

$$\zeta_i = \frac{1}{|U| p_k} (\widehat{C}_{U,I} + \lambda I)^{-1/2} (k_{x_{j_i}} \otimes k_{x_{j_i}}) (\widehat{C}_{U,I} + \lambda I)^{-1/2},$$

for $i \in \{1, \dots, M\}$. In particular note that $\zeta_1, \dots, \zeta_M$ are i.i.d. since $j_1, \dots, j_M$ are. Moreover note the following two facts

$$\begin{aligned}
\|\zeta_i\| &= \sup_{k \in \{1,\dots,|U|\}} \frac{1}{|U| p_k} \|(\widehat{C}_{U,I} + \lambda I)^{-1/2} k_{x_{u_k}}\|_{\mathcal{H}}^2 = s, \\
\mathbb{E}[\zeta_i] &= \sum_{k=1}^{|U|} p_k \frac{1}{|U| p_k} (\widehat{C}_{U,I} + \lambda I)^{-1/2} (k_{x_k} \otimes k_{x_k}) (\widehat{C}_{U,I} + \lambda I)^{-1/2} \\
&= (\widehat{C}_{U,I} + \lambda I)^{-1/2} \widehat{C}_{U,I} (\widehat{C}_{U,I} + \lambda I)^{-1/2} =: W,
\end{aligned}$$

where for the second identity we used the fact that $d/l_k = 1/(p_k|U|)$. Since by definition of $\widehat{C}_{J,A}$ we have

$$\begin{aligned}\frac{1}{M}\sum_{i=1}^{M}\zeta_i &= (\widehat{C}_{U,I}+\lambda I)^{-1/2}\left(\frac{1}{|J|}\sum_{i=1}^{M}\frac{1}{A_{ii}}k_{x_{j_i}}\otimes k_{x_{j_i}}\right)(\widehat{C}_{U,I}+\lambda I)^{-1/2}\\ &= (\widehat{C}_{U,I}+\lambda I)^{-1/2}\widehat{C}_{J,A}(\widehat{C}_{U,I}+\lambda I)^{-1/2},\end{aligned}$$

then, by applying non-commutative Bernstein inequality (Prop. 13 is a version specific for our problem), we have

$$\|(\widehat{C}_{U,I}+\lambda I)^{-1/2}(\widehat{C}_{J,A}-\widehat{C}_{U,I})(\widehat{C}_{U,I}+\lambda I)^{-1/2}\| = \big\|\frac{1}{M}\sum_{i=1}^{M}(\zeta_i-\mathbb{E}[\zeta_i])\big\| \leq \frac{2s\eta}{3M}+\sqrt{\frac{2s\|W\|\eta}{M}},$$

with probability at least $1-\tau$, and $\eta := \log\frac{4\operatorname{Tr}(W)}{\tau\|W\|}$. In particular, by noting that $\|W\|\leq 1$ by definition, when $M \geq 2s\eta$, then

$$\frac{2s\eta}{3M}+\sqrt{\frac{2s\|W\|\eta}{M}} \leq \frac{2s\eta}{3M}+\sqrt{\frac{2s\eta}{M}} \leq \frac{1}{3}\sqrt{\frac{2s\eta}{M}}+\sqrt{\frac{2s\eta}{M}} \leq \sqrt{\frac{4s\eta}{M}}.$$

To conclude note that $\frac{\operatorname{Tr}(W)}{\|W\|} \leq \operatorname{rank}(W) \leq |U| \leq n$, so $\eta \leq \log\frac{4n}{\tau}$. □

Lemma 26. *Let $n, R \in \mathbb{N}$, $(x_i)_{i=1}^n \subseteq \mathcal{X}$. Let $U = \{u_1,\dots,u_R\}$ with u_i i.i.d. with uniform probability on $\{1,\dots,n\}$. Let $\tau \in (0,1]$ and let $\lambda > 0$. When*

$$R \geq \frac{2n\kappa^2}{\lambda n+\kappa^2}\log\frac{4n}{\tau},$$

then the following holds with probability $1-\tau$

$$\|(\widehat{C}_n+\lambda I)^{-1/2}(\widehat{C}_{U,I}-\widehat{C}_n)(\widehat{C}_n+\lambda I)^{-1/2}\| \leq \sqrt{\frac{4n\kappa^2\log\frac{4n}{\tau}}{(\lambda n+\kappa^2)R}}.$$

Proof. Denote by ζ_i the random variable $\zeta_i = (\widehat{C}_n+\lambda I)^{-1/2}(k_{x_{u_i}}\otimes k_{x_{u_i}})(\widehat{C}_n+\lambda I)^{-1/2}$, for $i\in\{1,\dots,R\}$. Note that ζ_i are i.i.d. since u_i are. Moreover note that

$$\begin{aligned}\|\zeta_i\| &= \sup_{i\in\{1,\dots,n\}}\|(\widehat{C}_n+\lambda I)^{-1/2}k_{x_i}\|^2 \leq \sup_{i\in\{1,\dots,n\}}\|(\frac{1}{n}k_{x_i}\otimes k_{x_i}+\lambda I)^{-1/2}k_{x_i}\|^2\\ &\leq \frac{n\kappa^2}{\lambda n+\kappa^2} =: v.\end{aligned}$$

Moreover note that

$$\mathbb{E}[\zeta_i] = \frac{1}{n}\sum_{i=1}^{n}(\widehat{C}_n+\lambda I)^{-1/2}(k_{x_i}\otimes k_{x_i})(\widehat{C}_n+\lambda I)^{-1/2} = (\widehat{C}_n+\lambda I)^{-1/2}\widehat{C}_n(\widehat{C}_n+\lambda I)^{-1/2} =: W.$$

So we have, by non-commutative Bernstein inequality (Prop. 13 is a version specific for our problem),

$$\|(\widehat{C}_n + \lambda I)^{-1/2}(\widehat{C}_{U,I} - \widehat{C}_n)(\widehat{C}_n + \lambda I)^{-1/2}\| = \big\|\frac{1}{M}\sum_{i=1}^{M}(\zeta_i - \mathbb{E}[\zeta_i])\big\| \leq \frac{2v\eta}{3R} + \sqrt{\frac{2v\|W\|\eta}{R}},$$

with probability at least $1-\tau$, and $\eta := \log \frac{4\operatorname{Tr}(W)}{\tau\|W\|}$. In particular, by noting that $\|W\| \leq 1$ by definition, when $R \geq \frac{2n\kappa^2\eta}{(\lambda n+\kappa^2)R}$, analogously to the end of the proof of Lemma 25, we have $\frac{2v\eta}{3R} + \sqrt{\frac{2v\|W\|\eta}{R}} \leq \sqrt{\frac{4n\kappa^2\eta}{(\lambda n+\kappa^2)R}}$. To conclude note that $\frac{\operatorname{Tr}(W)}{\|W\|} \leq \operatorname{rank}(W) \leq n$, so $\eta \leq \log \frac{4n}{\tau}$. □

Lemma 27. *Let $n, R \in \mathbb{N}$, $(x_i)_{i=1}^n \subseteq \mathcal{X}$. Let $U = \{u_1, \dots, u_R\}$ with u_i i.i.d. with uniform probability on $\{1, \dots, n\}$. Let $\tau \in (0,1]$ and let $\lambda > 0$. When*

$$R \geq \frac{16n\kappa^2}{\lambda n + \kappa^2}\log\frac{4n}{\tau},$$

then the following holds with probability $1-\tau$

$$\frac{n}{R}\sum_{i=1}^{R}\widehat{\ell}(x_{u_i}, \lambda) < \max\left(5, \frac{6}{5}\widehat{\mathcal{N}}(\lambda)\right).$$

Proof. First of all denote with z_i the random variable $z_i = \frac{n}{R}\widehat{\ell}(x_{u_i}, \lambda)$ and note that $(z_i)_{i=1}^R$ are $i.i.d.$ since $(u_i)_{i=1}^R$ are. Moreover, by the characterization of $\widehat{\ell}(x, \lambda)$ via Prop. 9, we have

$$\begin{aligned}|z_i| \leq \sup_{k\in\{1,\dots,n\}} \|(\widehat{C}_n + \lambda I)^{-1/2}k_{x_k}\|^2 &\leq \sup_{k\in\{1,\dots,n\}} \|(k_{x_k}\otimes k_{x_k}/n + \lambda I)^{-1/2}k_{x_k}\|^2 \\ &\leq \frac{\kappa^2}{R(\kappa^2/n+\lambda)} =: v,\end{aligned}$$

moreover we have

$$\begin{aligned}\mathbb{E}[z_i] &= \mathbb{E}[\operatorname{Tr}((\widehat{C}_n + \lambda I)^{-1}(k_{x_{u_i}}\otimes k_{x_{u_i}}))] = \operatorname{Tr}((\widehat{C}_n + \lambda I)^{-1}\mathbb{E}[k_{x_{u_i}}\otimes k_{x_{u_i}}]) \\ &= \operatorname{Tr}\left((\widehat{C}_n + \lambda I)^{-1}\sum_{k=1}^{n}\frac{1}{n}k_{x_k}\otimes k_{x_k}\right) = \operatorname{Tr}\left((\widehat{C}_n + \lambda I)^{-1}\widehat{C}_n\right) = \widehat{\mathcal{N}}(\lambda).\end{aligned}$$

So by applying Bernstein inequality, the following holds with probability at least $1-\tau$

$$\left|\frac{n}{R}\sum_{i=1}^{R}\widehat{\ell}(x_{u_i}, \lambda) - \widehat{\mathcal{N}}(\lambda)\right| = \left|\frac{1}{R}\sum_{i=1}^{R}(z_i - \mathbb{E}[z_i])\right| \leq \frac{2v\log\frac{2}{\tau}}{3R} + \sqrt{\frac{2v\widehat{\mathcal{N}}(\lambda)\log\frac{2}{\tau}}{3R}}.$$

So we have

$$\frac{n}{R}\sum_{i=1}^{R}\widehat{\ell}(x_{u_i},\lambda) \le \widehat{\mathcal{N}}(\lambda) + \left|\frac{n}{R}\sum_{i=1}^{R}\widehat{\ell}(x_{u_i},\lambda) - \widehat{\mathcal{N}}(\lambda)\right| \le \widehat{\mathcal{N}}(\lambda) + \frac{2v\log\frac{2}{\tau}}{3R} + \sqrt{\frac{2v\widehat{\mathcal{N}}(\lambda)\log\frac{2}{\tau}}{R}}.$$

Now, if $\widehat{\mathcal{N}}(\lambda) \le 4$, since $R \ge 16v\log\frac{2}{\tau}$, we have that

$$\widehat{\mathcal{N}}(\lambda) + \frac{2v\log\frac{2}{\tau}}{3R} + \sqrt{\frac{2v\widehat{\mathcal{N}}(\lambda)\log\frac{2}{\tau}}{R}} \le 4 + \frac{1}{24} + \sqrt{\frac{1}{2}} < 5.$$

If $\widehat{\mathcal{N}}(\lambda) > 4$, since $R \ge 16v\log\frac{2}{\tau}$, we have

$$\widehat{\mathcal{N}}(\lambda) + \frac{2v\log\frac{2}{\tau}}{3R} + \sqrt{\frac{2v\widehat{\mathcal{N}}(\lambda)\log\frac{2}{\tau}}{3R}} \le \left(1 + \frac{1}{24\widehat{\mathcal{N}}(\lambda)} + \sqrt{\frac{1}{8\widehat{\mathcal{N}}(\lambda)}}\right)\widehat{\mathcal{N}}(\lambda) < \frac{6}{5}\widehat{\mathcal{N}}(\lambda).$$

□

Theorem 14. *Let $n \in \mathbb{N}$, $(x_i)_{i=1}^n \subseteq X$. Let $\delta \in (0,1]$, $t, q > 1$, $\lambda > 0$ and $H, d_h, \lambda_h, J_h, A_h, U_h$ as in Alg. 3. Let $\overline{A}_h = \frac{n}{|J_h|}A_h$ and $\nu_h = G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h,\overline{A}_h})$, $\beta_h = G_{\lambda_h}(\widehat{C}_{U_h,I}, \widehat{C}_{J_h,\overline{A}_h})$, $\eta_h = G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{U_h,I})$. When*

$$\lambda_0 = \frac{\kappa^2}{\min(t,1)}, \quad q_1 \ge \frac{5\kappa^2 q_2}{q(1+t)}, \quad q_2 \ge 12q\frac{(2t+1)^2}{t^2}(1+t)\log\frac{12Hn}{\delta},$$

then the following holds with probability $1-\delta$: for any $h \in \{0,\dots,H\}$

$$\begin{aligned} &a) \quad \frac{1}{T}\widehat{\ell}(x,\lambda_h) \le \widehat{\ell}_{J_h,\overline{A}_h}(x) \le \min(T,2)\widehat{\ell}(x,\lambda_h), \quad \forall x \in \mathcal{X}, \\ &b) \quad d_h \le 3q\,\widehat{\mathcal{N}}(\lambda_h) \vee 10q, \ \textit{and} \ |J_h| \le q_2(3q\widehat{\mathcal{N}}(\lambda_h) \vee 10q). \\ &c) \quad \beta_h \le \frac{7}{11c_T}, \ \eta_h \le \frac{3}{11c_T}, \ \nu_h \le \frac{1}{c_T}. \end{aligned} \tag{5.12}$$

where $T = 1+t$ and $c_T = 2 + 1/(T-1)$.

Proof. Let H, c_T, q and $\lambda_h, U_h, J_h, A_h, d_h, P_h = (p_{h,k})_{k=1}^{R_h}$, for $h \in \{0,\dots,H\}$ as defined in Alg. 3. Let $\overline{A}_h = \frac{n}{|J_h|}A_h$ and $\tau = \delta/(3H)$. Now we are going to define some events and we prove a recurrence relation that they satisfy. Finally we unroll the recurrence relation and bound the resulting events in probability.

Definitions of the events. Now we are going to define some events that will be useful to prove the theorem. Denote with E_h the event such that the conditions in Eq. (5.12)-(a) hold for J_h, A_h, U_h. Denote with F_h the event such that

$$\frac{n}{R_h}\sum_{u\in U_h}\widehat{\ell}(x_u,\lambda_{h-1}) \le \frac{6}{5}\widehat{\mathcal{N}}(\lambda).$$

Denote with $B_{1,h}$ the event such that β_h, satisfies

$$\beta_h \leq \sqrt{\frac{4s_h \log \frac{4n}{\tau}}{M_h}}, \quad \text{with} \quad s_h := \sup_{k\in\{1,\dots,R_h\}} \frac{1}{R_h p_{h,k}} \|(\widehat{C}_{U_h,I} + \lambda_h I)^{-1/2} k_{x_{u_k}}\|^2. \tag{5.13}$$

Denote with $B_{2,h}$ the event such that η_h, satisfies

$$\eta_h \leq \sqrt{\frac{4\kappa^2 n \log \frac{\kappa^2}{\lambda_h \tau}}{(\lambda_h n + \kappa^2) R_h}}.$$

First bound for s_h. Note that, by definition of $p_{h,k}$, that is, by Prop. 9

$$p_{h,k} = n\widetilde{\ell}_{J_{h-1},A_{h-1}}(x_{u_k}, \lambda_h)/(d_h R_h) = n\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_{u_k}, \lambda_h)/(d_h R_h),$$

so

$$s_h = \sup_{k\in\{1,\dots,R_h\}} \frac{d_h \|(\widehat{C}_{U_h,I} + \lambda_h I)^{-1/2} k_{x_{u_k}}\|^2}{n\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_{u_k}, \lambda_h)} = \sup_{u\in U_h} \frac{d_h \widehat{\ell}_{U_h,I}(x_u, \lambda_h)}{\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_u, \lambda_h)},$$

where the last step consists in apply the definition of $\widehat{\ell}_{U_h,I}$. By applying Lemma 23 and 24 to $\widehat{\ell}_{U_h,I}(x, \lambda_h)$, we have

$$\widehat{\ell}_{U_h,I}(x, \lambda_h) \leq \frac{1}{1-\eta_h} \widehat{\ell}(x, \lambda_h) \leq \frac{\lambda_{h-1}}{\lambda_h(1-\eta_h)} \widehat{\ell}(x, \lambda_{h-1})$$

and analogously by applying Lemma 24 to $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x, \lambda_h)$, we have

$$\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x, \lambda_h) \geq \widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x, \lambda_{h-1}).$$

So, by extending the $\sup$ of s_h to the whole $\mathcal{X}$, we have

$$s_h \leq d_h \sup_{x\in\mathcal{X}} \frac{\widehat{\ell}_{U_h,I}(x, \lambda_h)}{\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x, \lambda_h)} \leq \frac{\lambda_{h-1} d_h}{\lambda_h(1-\eta_h)} \sup_{x\in\mathcal{X}} \frac{\widehat{\ell}(x, \lambda_{h-1})}{\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x, \lambda_{h-1})}.$$

Now we are ready to prove the recurrence relation, for $h \in \{1, \dots H\}$,

$$E_h \supseteq B_{1,h} \cap B_{2,h} \cap E_{h-1} \cap F_h.$$

Analysis of E_0. Note that, since $\|\widehat{C}_n\| \leq \kappa^2$, then $\frac{1}{\kappa^2+\lambda} I \preceq (\widehat{C}_n + \lambda I)^{-1} \preceq \frac{1}{\lambda}$, so for any $x \in \mathcal{X}$ the following holds

$$\frac{k(x,x)}{(\kappa^2+\lambda)n} \leq \widehat{\ell}(x, \lambda) \leq \frac{k(x,x)}{\lambda n}.$$

Since $\lambda_0 = \frac{\kappa^2}{\min(2,T)-1}$ and $\widehat{\ell}_{\emptyset,[]}(x,\lambda_0) = \frac{k(x,x)}{\lambda_0 n}$, we have

$$\frac{1}{T}\widehat{\ell}(x,\lambda_0) \le \frac{1}{T}\frac{k(x,x)}{\lambda n} \le \ell_{\emptyset,[]}(x,\lambda_0) = \frac{k(x,x)}{\lambda_0 n} = \frac{\min(2,T)k(x,x)}{(\kappa^2+\lambda_0)n} \le \min(2,T)\widehat{\ell}(x,\lambda_0).$$

Setting conventionally $d_0, \nu_0, \eta_0, \beta_0 = 0$ (they are not used by the algorithm or the proof), we have that E_0 holds everywhere and so, with probability 1.

Analysis of $E_{h-1} \cap B_{1,h} \cap B_{2,h}$. First note that under E_{h-1}, the following holds $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x,\lambda_{h-1}) \ge \frac{1}{T}\widehat{\ell}(x,\lambda_{h-1})$ and so

$$s_h \le \frac{\lambda_{h-1}d_h}{\lambda_h(1-\eta_h)} \sup_{x\in\mathcal{X}} \frac{\widehat{\ell}(x,\lambda_{h-1})}{\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x,\lambda_{h-1})} \le \frac{\lambda_{h-1}d_h}{\lambda_h(1-\eta_h)} \sup_{x\in\mathcal{X}} \frac{\widehat{\ell}(x,\lambda_{h-1})}{\frac{1}{T}\widehat{\ell}(x,\lambda_{h-1})} \le \frac{T\lambda_{h-1}d_h}{\lambda_h(1-\eta_h)}.$$

Now note that under $B_{2,h}$, by applying the definition of R_h in Alg. 3, by the condition on q_1, we have

$$\eta_h \le \sqrt{\frac{4\kappa^2 n \log\frac{\kappa^2}{\lambda_h\tau}}{(\lambda_h n+\kappa^2)R_h}} \le \sqrt{\frac{4\kappa^2 n \log\frac{\kappa^2}{\lambda_h\tau}}{\min\{\lambda_h n,\kappa^2\}R_h}} \le \sqrt{\frac{4\log\frac{\kappa^2}{\lambda_h\tau}}{q_1}} \le 3/(11c_T) \le 3/22.$$

So under $B_{1,h} \cap B_{2,h} \cap E_{h-1}$ and the fact that $q = \frac{\lambda_{h-1}}{\lambda_h}$, we have $s_h \le \frac{T\lambda_{h-1}d_h}{\lambda_h(1-\eta_h)} \le (8/7)qTd_h$ and so, since $M_h = q_2 d_h$, by the condition on q_2, we have

$$\beta_h \le \sqrt{\frac{4s_h\log\frac{4n}{\tau}}{M_h}} \le \sqrt{\frac{(32/7)qTd_h\log\frac{4n}{\tau}}{M_h}} = \sqrt{\frac{(32/7)qT\log\frac{4n}{\tau}}{q_2}} < \frac{7}{11c_T},$$

where in the last step we used the definition of M_h in Alg. 3. Then, since under $B_{1,h}\cap B_{2,h}\cap E_{h-1}$ we have that $\beta_h \le 7/(11c_T)$, $\eta_h \le 3/(11c_T) \le 3/22$, then, by applying Proposition 11 to ν_h w.r.t. η_h, β_h, we have

$$\nu_h \le \eta_h + (1+\eta_h)\beta_h \le \left(\frac{3}{11} + \left(1+\frac{3}{22}\right)\frac{7}{11}\right)\frac{1}{c_T} < \frac{1}{c_T}.$$

Then $\frac{1}{T} \le \frac{1-2\nu_h}{1-\nu_h}$ and $\frac{1}{1-\nu_h} \le \min(T,2)$, so by applying Theorem 13, we have

$$\frac{1}{T}\widehat{\ell}(x,\lambda_h) \le \widehat{\ell}_{J_h,\overline{A}_h}(x,\lambda_h) \le \min(T,2)\widehat{\ell}(x,\lambda_h).$$

Analysis of $E_{h-1} \cap F_h$. First note that under E_{h-1} the following holds $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x,\lambda_{h-1}) \le \min(T,2)\widehat{\ell}(x,\lambda_{h-1})$, so, by applying Lemma 24 to $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x,\lambda_h)$, we have

$$d_h = \frac{n}{R_h}\sum_{u\in U_h}\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_u,\lambda_h) \le \frac{\lambda_{h-1}n}{\lambda_h R_h}\sum_{u\in U_h}\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_u,\lambda_{h-1}) \le \frac{2\lambda_{h-1}n}{\lambda_h R_h}\sum_{u\in U_h}\widehat{\ell}(x_u,\lambda_{h-1}).$$

Moreover under F_h, we have $\frac{n}{R_h}\sum_{u\in U_h}\widehat{\ell}(x_u,\lambda_{h-1}) \leq \max(5, \frac{6}{5}\widehat{\mathcal{N}}(\lambda_{h-1}))$, so, under $E_{h-1}\cap F_h$, we have

$$d_h \leq 2q\max(5,\ (6/5)\widehat{\mathcal{N}}(\lambda_{h-1})) \leq \max(10q, 3q\widehat{\mathcal{N}}(\lambda_h)).$$

This implies that

$$|J_h| = M_h = q_2 d_h \leq q_2\max(10q, 3q\widehat{\mathcal{N}}(\lambda_h))$$

Unrolling the recurrence relation. The two results above imply $E_h \supseteq B_{1,h}\cap B_{2,h}\cap E_{h-1}\cap F_h$. Now we unroll the recurrence relation, obtaining

$$E_h \supseteq E_0 \cap (\cap_{j=1}^h F_j) \cap (\cap_{j=1}^h B_{1,j}) \cap (\cap_{j=1}^h B_{2,j}),$$

so by taking their intersections, we have

$$\cap_{h=0}^H E_h \supseteq E_0 \cap (\cap_{j=1}^H F_j) \cap (\cap_{j=1}^H B_{1,j}) \cap (\cap_{j=1}^H B_{2,j}). \tag{5.14}$$

Bounding $B_{1,h}, B_{2,h}, F_h$ in high probability. Let $h \in [H]$. The probability of the event $B_{1,h}$ can be written as $\mathbb{P}(B_{1,h}) = \int \mathbb{P}(B_{1,h}|U_h,P_h)d\mathbb{P}(U_h,P_h)$. Now note that $\mathbb{P}(B_{1,h}|U_h,P_h)$ is controlled by Lemma 25, that proves that for any U_h, P_h, the probability of $\mathbb{P}(B_{1,h}|U_h,P_h)$ is at least $1-\tau$. Then

$$\mathbb{P}(B_{1,h}) = \int \mathbb{P}(B_{1,h}|U_h,P_h)d\mathbb{P}(U_h,P_h) \geq \inf_{U_h}\mathbb{P}(B_{1,h}|U_h,P_h) \geq 1-\tau.$$

To see that $\mathbb{P}(B_{1,h}|U_h,P_h)$ is controlled by Lemma 25, note that, since $|U_h|$ is exactly R_h, by definition of $\overline{A}_h$ and A_h

$$\overline{A}_h = \frac{n}{|J_h|}A_h = |U_h|\ \mathrm{diag}(p_{j_1},\dots,p_{j_{|J_h|}}),$$

that is exactly the condition on the weights required by Lemma 25 which controls exactly Equation (5.13). Finally $B_{2,h}, F_h$ are directly controlled respectively by Lemmas 26 and 27 and so hold with probability at least $1-\tau$ each. Finally note that E_0 holds with probability 1. So by taking the intersection bound according to Equation (5.14), we have that $\cap_{h=0}^H E_h$ holds at least with probability $1-3H\tau$. □

5.2.6 Proof for BLESS-R (Alg. 4)

Similarly to the previous section, this section presents two technical lemmas used to prove Theorem 15 that provides the guarantees for BLESS-R (Algorithm 4).

Lemma 28. *Let $\lambda > 0$, $n \in \mathbb{N}$, $\delta \in (0,1]$. Let $(x_i)_{i=1}^n \subseteq \mathcal{X}$. Let $b \in (0,1]$ and $p_1,\dots,p_n \in (0,b]$. Let $u_1,\dots u_n$ sampled independently and uniformly on $[0,1]$. Let v_j be independent $Bernoulli(p_j/b)$ random variables, with $j \in [n]$. Denote by z_j the random variable $z_j =$*

$1_{u_j \leq b} v_j$. *Finally, let the random set* J *containing* j *iff* $z_j = 1$. *Let* $A = \frac{n}{|J|}(p_{j_1}, \ldots, p_{j_{|J|}})$, *where* $j_1, \ldots, j_{|J|}$ *are the sorting of* J. *Then the following holds with probability at least* $1 - \delta$

$$G_\lambda(\widehat{C}_n, \widehat{C}_{J,A}) \leq \frac{2s\eta}{3n} + \sqrt{\frac{2s\eta}{n}}, \quad \textit{with} \quad s = \sup_{i \in [n]} \frac{1}{p_i} \|(\widehat{C}_n + \lambda I)^{-1/2} k_{x_i}\|_{\mathcal{H}}^2,$$

with $s = \log \frac{4n}{\delta}$.

Proof. Let ζ_i be defined as

$$\zeta_i = \frac{z_i}{p_i} \frac{1}{n} (\widehat{C}_n + \lambda I)^{-1/2} (k_{x_i} \otimes k_{x_i}) (\widehat{C}_n + \lambda I)^{-1/2},$$

for $i \in [n]$, where z_i are the Bernoulli random variables computed by Algorithm 4. First note that

$$\begin{aligned}
(\widehat{C}_n + \lambda I)^{-1/2} \widehat{C}_{J,A} (\widehat{C}_n + \lambda I)^{-1/2} &= \frac{1}{|J|} \sum_{j \in J} \frac{|J|}{n p_j} (\widehat{C}_n + \lambda I)^{-1/2} (k_{x_i} \otimes k_{x_i}) (\widehat{C}_n + \lambda I)^{-1/2} \\
&= \frac{1}{n} \sum_{j \in J} \frac{1}{p_j} (\widehat{C}_n + \lambda I)^{-1/2} (k_{x_i} \otimes k_{x_i}) (\widehat{C}_n + \lambda I)^{-1/2} \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{z_i}{p_j} (\widehat{C}_n + \lambda I)^{-1/2} (k_{x_i} \otimes k_{x_i}) (\widehat{C}_n + \lambda I)^{-1/2} \\
&= \sum_{i=1}^{n} \zeta_i.
\end{aligned}$$

In particular we study the expectation and the variance of ζ_i to bound $G_\lambda(\widehat{C}_n, \widehat{C}_{J,A})$. By noting that the expectation of z_i is $\mathbb{E}[z_i] = \mathbb{E}[1_{u_i \geq b} v_i] = \mathbb{E}[1_{u_i \geq b}] \mathbb{E}[v_i] = b \times \frac{p_i}{b} = p_i$, for any $i \in [n]$, then

$$\begin{aligned}
\mathbb{E} \sum_{i=1}^{n} \zeta_i &= \sum_{i=1}^{n} \frac{\mathbb{E}[z_i]}{p_i} \frac{1}{n} (\widehat{C}_n + \lambda I)^{-1/2} (k_{x_i} \otimes k_{x_i}) (\widehat{C}_n + \lambda I)^{-1/2} \\
&= \sum_{i=1}^{n} \frac{1}{n} (\widehat{C}_n + \lambda I)^{-1/2} (k_{x_i} \otimes k_{x_i}) (\widehat{C}_n + \lambda I)^{-1/2} \\
&= (\widehat{C}_n + \lambda I)^{-1/2} \widehat{C}_n (\widehat{C}_n + \lambda I)^{-1/2} =: W,
\end{aligned}$$

Now we will bound almost everywhere $\|\zeta_i\|$ as

$$\|\zeta_i\| \leq \sup_{i \in [n]} \frac{z_i}{p_i} \frac{1}{n} \|(\widehat{C}_n + \lambda I)^{-1/2} k_{x_i}\|_{\mathcal{H}}^2 \leq \frac{1}{n} \sup_{i \in [n]} \frac{1}{p_i} \|(\widehat{C}_n + \lambda I)^{-1/2} k_{x_i}\|_{\mathcal{H}}^2.$$

We are ready to apply non-commutative Bernstein inequality (Prop. 13 is specific version for this setting), obtaining, with probability at least $1-\delta$

$$G_\lambda(\widehat{C}_n, \widehat{C}_{J,A}) = \|\frac{1}{n}\sum_{i=1}^{n}(\zeta_i - \mathbb{E}[\zeta_i])\| \leq \frac{2s\eta}{3n} + \sqrt{\frac{2s\eta}{n}},$$

with $\eta = \log \frac{4\operatorname{Tr}(W)}{\|W\|\delta}$. Finally note that since $\operatorname{Tr}(W)/\|W\| \leq \operatorname{rank}(W) \leq n$, we have $\eta \leq \log \frac{4n}{\delta}$. □

Lemma 29. *Let $\lambda > 0$, $n \in \mathbb{N}$, $\delta \in (0,1]$. Let $(x_i)_{i=1}^n \subseteq \mathcal{X}$. Let $b \in (0,1]$ and $p_1, \dots, p_n \in (0,b]$. Let $u_1, \dots u_n$ sampled independently and uniformly on $[0,1]$. Let v_j be independent $Bernoulli(p_j/b)$ random variables, with $j \in [n]$. Denote by z_j the random variable $z_j = 1_{u_j \leq b} v_j$. Finally, let the random set J containing j iff $z_j = 1$. Then the following holds with probability at least $1-\delta$*

$$|J| \leq \sum_{i\in[n]} p_i + (1 + \sqrt{\sum_{i\in[n]} p_i}) \log \frac{3}{\delta}.$$

Proof. By definition of J, note that

$$|J| = \sum_{i\in[n]} z_i.$$

We are going to concentrate the sum of random variables via Bernstein. Any z_i is bounded, by construction, by 1. Moreover

$$\mathbb{E}[z_i] = \mathbb{E}[1_{u_i \geq b} v_i] = \mathbb{E}[1_{u_i \geq b}]\mathbb{E}[v_i] = b \times \frac{p_i}{b} = p_i.$$

Analogously $\mathbb{E}[z_i^2] - \mathbb{E}[z_i]^2 = p_i - p_i^2 \leq p_i$. By applying Bernstein inequality, we have

$$|\sum_{i\in[n]}(z_i - p_i)| \leq \log \frac{2}{\delta} + \sqrt{\log \frac{2}{\delta} \sum_{i\in[n]} p_i},$$

with probability $1-\delta$. Then with the same probability,

$$|J| \leq \sum_{i\in[n]} p_i + (1 + \sqrt{\sum_{i\in[n]} p_i}) \log \frac{3}{\delta}.$$

□

Theorem 15. *Let $n \in \mathbb{N}$, $(x_i)_{i=1}^n \subseteq X$. Let $\delta \in (0,1]$, $t, q > 1$, $\lambda > 0$ and $H, d_h, \lambda_h, J_h, A_h$ as in Alg. 4. Let $\overline{A}_h = \frac{n}{|J_h|} A_h$ and $\nu_h = G_\lambda(\widehat{C}_n, \widehat{C}_{J_h, \overline{A}_h})$. When*

$$\lambda_0 = \frac{\kappa^2}{\min(t,1)}, \quad q_1 \geq 2Tq(1+2/t)\log\frac{4n}{\delta}$$

then, the following holds with probability $1-\delta$*: for any* $h \in \{0, \dots, H\}$

$$
\begin{aligned}
&a) \qquad \frac{1}{T}\widehat{\ell}(x,\lambda_h) \;\le\; \widehat{\ell}_{J_h,\overline{A}_h}(x) \;\le\; \min(T,2)\widehat{\ell}(x,\lambda_h), \quad \forall x \in \mathcal{X}, \\
&b) \qquad |J_h| \;\le\; 3q_1 \min(T,2)\left(5 \vee \widehat{\mathcal{N}}(\lambda_h)\right)\log\frac{6H}{\delta}, \\
&c) \qquad \nu_h \le \frac{1}{c_T}.
\end{aligned}
\tag{5.15}
$$

where $T = 1 + t$ *and* $c_T = 2 + 1/(T-1)$.

Proof. Let H, c_T, q and $\lambda_h, J_h, A_h, (p_{h,i})_{i=1}^n$ for $h \in \{0, \dots, H\}$ as defined in Alg. 4 and define $\tau = \delta/(2H)$. Now we are going to define some events and we prove a recurrence relation that they satisfy. Finally we unroll the recurrence relation and bound the resulting events in probability.

Definitions of the events. Now we are going to define some events that will be useful to prove the theorem. Denote with E_h the event such that the conditions in Eq. (5.15)-(a) hold for $J_h, \overline{A}_h$. Denote with Z_h the event such that

$$
|J_h| \le \sum_{i\in[n]} p_{h,i} + (1 + (\sum_{i\in[n]} p_{h,i})^{1/2})\log\frac{3}{\tau}.
$$

Denote with V_h the event such that $\nu_h := G_{\lambda_h}(\widehat{C}_{U,I}, \widehat{C}_{J_h,A_h})$, satisfies

$$
\nu_h \le s_h \log\frac{8\kappa^2}{\lambda_h\tau} + \sqrt{2s_h\log\frac{8\kappa^2}{\lambda_h\tau}}, \quad \text{with} \quad s_h = \sup_{i\in[n]}\frac{1}{np_{h,i}}\|(\widehat{C}_n + \lambda_h I)^{-1/2}k_{x_i}\|^2_{\mathcal{H}}. \tag{5.16}
$$

Analysis of s_h**.** Note that, by definition of $p_{h,i}$, for Algorithm 4, and of $\widehat{\ell}$, we have so

$$
s_h = \sup_{i\in[n]}\frac{1}{np_{h,i}}\|(\widehat{C}_n + \lambda_h I)^{-1/2}k_{x_i}\|^2_{\mathcal{H}} = \sup_{i\in[n]}\frac{\widehat{\ell}(x_i,\lambda_i)}{q_1\widetilde{\ell}_{J_h,A_h}(x_i)} = \sup_{i\in[n]}\frac{\widehat{\ell}(x_i,\lambda_i)}{q_1\widehat{\ell}_{J_h,\overline{A}_h}(x_i)}.
$$

with $\overline{A}_h = \frac{n}{|J_h|}A_h$, where the last step is due to the equivalence between $\widetilde{\ell}$ and $\widehat{\ell}$ in Proposition 9.

Now we are ready to prove the recurrence relation, for $h \in \{1, \dots H\}$,

$$
E_h \supseteq V_h \cap Z_h \cap E_{h-1}.
$$

Analysis of E_0**.** Note that, since $\|\widehat{C}_n\| \le \kappa^2$, then $\frac{1}{\kappa^2+\lambda}I \preceq (\widehat{C}_n + \lambda I)^{-1} \preceq \frac{1}{\lambda}$, so for any $x \in \mathcal{X}$ the following holds

$$
\frac{k(x,x)}{(\kappa^2+\lambda)n} \le \widehat{\ell}(x,\lambda) \le \frac{k(x,x)}{\lambda n}.
$$

Since $\lambda_0 = \frac{\kappa^2}{\min(2,T)-1}$ and $\widehat{\ell}_{\emptyset,[]}(x,\lambda_0) = \frac{k(x,x)}{\lambda_0 n}$, we have

$$\frac{1}{T}\widehat{\ell}(x,\lambda_0) \le \frac{1}{T}\frac{k(x,x)}{\lambda n} \le \ell_{\emptyset,[]}(x,\lambda_0) = \frac{k(x,x)}{\lambda_0 n} = \frac{\min(2,T)k(x,x)}{(\kappa^2+\lambda_0)n} \le \min(2,T)\widehat{\ell}(x,\lambda_0).$$

Setting conventionally $d_0, \nu_0, \eta_0, \beta_0 = 0$ (they are not used by the algorithm or the proof), we have that E_0 holds everywhere and so, with probability 1.

Analysis of $E_{h-1} \cap V_h$. Note that under E_{h-1}, we have $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x,\lambda_{h-1}) \ge \frac{1}{T}\widehat{\ell}(x,\lambda_{h-1})$, so

$$\begin{aligned} s_h &= \sup_{i\in[n]} \frac{\widehat{\ell}(x_i,\lambda_h)}{q_1 \widehat{\ell}_{J_h,\overline{A}_h}(x_i,\lambda_{h-1})} \le T \sup_{i\in[n]} \frac{\widehat{\ell}(x_i,\lambda_h)}{q_1 \widehat{\ell}(x_i,\lambda_{h-1})} \\ &\le \frac{T\lambda_{h-1}}{\lambda_h} \sup_{i\in[n]} \frac{\widehat{\ell}(x_i,\lambda_{h-1})}{q_1 \widehat{\ell}(x_i,\lambda_{h-1})} = \frac{T\lambda_h}{q_1\lambda_{h-1}} = \frac{Tq}{q_1}, \end{aligned}$$

where we used the fact that $\widehat{\ell}(x_i,\lambda_h) \le \frac{\lambda_{h-1}}{\lambda_h}\widehat{\ell}(x_i,\lambda_{h-1})$, via Lemma 24. In particular since we are in V_h, this means that, since $q_1 \ge 2Tq(1+2/t)\log\frac{4n}{\delta}$, we have

$$\nu_h \le \frac{Tq}{q_1}\log\frac{8\kappa^2}{\lambda_h\tau} + \sqrt{2\frac{Tq}{q_1}\log\frac{8\kappa^2}{\lambda_h\tau}} \le (4+2t^{-1})^{-2} + \sqrt{2/(4+2t^{-1})^2} \tag{5.17}$$

$$\le (1/8+\sqrt{1/8})(2+t^{-1})^{-1} \le \frac{1}{2c_T}. \tag{5.18}$$

Then $\frac{1}{T} \le \frac{1-2\nu_h}{1-\nu_h}$ and $\frac{1}{1-\nu_h} \le \min(T,2)$, so by applying Theorem 13, we have

$$\frac{1}{T}\widehat{\ell}(x,\lambda_h) \le \widehat{\ell}_{J_h,\overline{A}_h}(x,\lambda_h) \le \min(T,2)\widehat{\ell}(x,\lambda_h).$$

Analysis of $E_{h-1} \cap Z_h$. First consider $\sum_{i\in[n]} p_{h,i}$. By the fact that $\widetilde{\ell}_{J_{h-1},A_{h-1}} = \widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}$, by Proposition 9, we have

$$\begin{aligned} \sum_{i\in[n]} p_{h,i} &= q_1 \sum_{i\in[n]} \widetilde{\ell}_{J_{h-1},A_{h-1}}(x_i,\lambda_h) = q_1 \sum_{i\in[n]} \widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_i,\lambda_h) \\ &\le q_1 \frac{\lambda_{h-1}}{\lambda_h} \sum_{i\in[n]} \widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_i,\lambda_{h-1}), \le q_1 \min(T,2)\frac{\lambda_{h-1}}{\lambda_h} \sum_{i\in[n]} \widehat{\ell}(x_i,\lambda_{h-1}), \\ &\le q_1 \min(T,2)\frac{\lambda_{h-1}}{\lambda_h} \sum_{i\in[n]} \widehat{\ell}(x_i,\lambda_h) = q_1 \min(T,2)\widehat{\mathcal{N}}(\lambda_h), \end{aligned}$$

where we applied in order (1) Lemma 24, to bound $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_i,\lambda_h)$ in terms of $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_i,\lambda_{h-1})$, (2) the fact that we are in the event E_{h-1} and so $\widehat{\ell}_{J_{h-1},\overline{A}_{h-1}}(x_i,\lambda_{h-1}) \le \min(T,2)\widehat{\ell}(x_i,\lambda_{h-1})$, then (3) again Lemma 24 to bound $\widehat{\ell}(x_i,\lambda_{h-1})$ w.r.t. $\widehat{\ell}(x_i,\lambda_h)$, and (4) finally the definition of $\widehat{\mathcal{N}}(\lambda_h)$.

Now if $\widehat{\mathcal{N}}(\lambda_h) \leq 10$, we have that

$$\sum_{i\in[n]} p_{h,i} + (1 + (\sum_{i\in[n]} p_{h,i})^{1/2}) \log\frac{3}{\tau} \leq 15 q_1 \min(T, 2) \log\frac{3}{\tau}.$$

If $\widehat{\mathcal{N}}(\lambda_h) > 10$, we have that

$$\sum_{i\in[n]} p_{h,i} + (1 + (\sum_{i\in[n]} p_{h,i})^{1/2}) \log\frac{3}{\tau} \leq 3\widehat{\mathcal{N}}(\lambda_h) q_1 \min(T, 2) \log\frac{3}{\tau}.$$

So under $E_{h-1} \cap Z_h$, we have that

$$|J| \leq 3 q_1 \min(T, 2) \left(5 \vee \widehat{\mathcal{N}}(\lambda_h)\right) \log\frac{3}{\tau}.$$

Unrolling the recurrence relation. The two results above imply $E_h \supseteq V_h \cap Z_h \cap E_{h-1}$. Now we unroll the recurrence relation, obtaining

$$E_h \supseteq E_0 \cap (\cap_{j=1}^h Z_j) \cap (\cap_{j=1}^h V_j),$$

so by taking their intersections, we have

$$\cap_{h=0}^H E_h \supseteq E_0 \cap (\cap_{j=1}^H Z_j) \cap (\cap_{j=1}^H V_j). \tag{5.19}$$

Bounding V_h, Z_h in high probability Let $h \in [H]$. Denote by $P_h = (p_{h,j})_{j\in[n]}$. The probability of the event Z_h can be written as $\mathbb{P}(Z_h) = \int \mathbb{P}(Z_h|P_h) d\mathbb{P}(P_h)$. Now note that $\mathbb{P}(Z_h|P_h)$ is controlled by Lemma 29, that proves that the probability of $\mathbb{P}(Z_h|P_h)$ is at least $1-\tau$. Then

$$\mathbb{P}(Z_h) = \int \mathbb{P}(Z_h|P_h) d\mathbb{P}(P_h) \geq \inf_{P_h} \mathbb{P}(Z_h|P_h) \geq 1 - \tau.$$

The probability event V_h is lower bounded by $1-\tau$, via the same reasoning, using Lemma 28. Finally note that E_0 holds with probability 1. So by taking the intersection bound according to Equation (5.19), we have that $\cap_{h=0}^H E_h$ holds at least with probability $1 - 3H\tau$. □

5.2.7 Proof of Theorem 12

Here we state the proof of Theorem 12, presented in Section 5.1.6.

Proof. The proof of this theorem splits in the proof for Algorithm 3 that corresponds to Theorem 14 and the proof for Algorithm 4, that corresponds to Theorem 15. In particular, the

result about leverage scores is expressed in terms of out-of-sample-leverage-scores $\widehat{\ell}_{J_h,A_h}$ (Definition 8). The desired result, about $\widetilde{\ell}_{J_h,A_h}$, is obtained via Proposition 9.

Note that the two theorems provides stronger guarantees than the ones required by this theorem. We will use only points (a) and (b) of their statements. Moreover they prove the result for the out-of-sample-leverage-scores (Definition 8) and here we specify the result only for $x = x_i$, with $i \in [n]$. □

5.3 Efficient Supervised Learning with Leverage Scores

In this section, we discuss the impact of BLESS in a supervised learning. Unlike most previous results on leverage scores sampling in this context [AM15a, CLV17a, MM17], we consider the setting of statistical learning we presented in Chapter 2. The notation used is the one defined in Section 4.6.1.

5.3.1 Learning with FALKON-BLESS

The algorithm we propose, called FALKON-BLESS, combines BLESS with FALKON (see Chapter 4) As we discuss in the following, the combination with BLESS leads to further improvements.

We now quickly recall FALKON and its algorithmic ideas. First, sampling is used to select a subset $\{\widetilde{x}_1, \dots, \widetilde{x}_M\} \subseteq \{x_1, \dots, x_n\}$ of the input data uniformly at random, and to define an approximate solution

$$\widehat{f}_{\lambda,M}(x) = \sum_{j=1}^{M} k(\widetilde{x}_j, x)c_j, \qquad c = (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda \widehat{K}_{MM})^{-1} \widehat{K}_{nM}^\top y, \tag{5.20}$$

where $c = (c_1, \dots, c_M)$, $\widehat{K}_{nM} \in \mathbb{R}^{n\times M}$, has entries $(\widehat{K}_{nM})_{ij} = k(x_i, \widetilde{x}_j)$ and $\widehat{K}_{MM} \in \mathbb{R}^{M\times M}$ has entries $(\widehat{K}_{MM})_{jj'} = k(\widetilde{x}_j, \widetilde{x}_{j'})$, with $i \in [n], j, j' \in [M]$. FALKON proposes to compute a solution of the linear system 5.20 via a preconditioned iterative solver. The preconditioner is the core of the algorithm and is defined by a matrix B such that

$$BB^\top = \Big(\frac{n}{M}\widehat{K}_{MM}^2 + \lambda \widehat{K}_{MM}\Big)^{-1}. \tag{5.21}$$

The overall algorithm has complexity $\mathcal{O}(nMt)$ in time and $\mathcal{O}(M^2)$ in space, where t is the number of conjugate gradient iterations performed.
In this Chapter, we analyze a variation of FALKON where the points $\{\widetilde{x}_1, \dots, \widetilde{x}_M\}$ are selected via leverage score sampling using BLESS, see Algorithm 3 or Algorithm 4, so that $M = M_h$

and $\widetilde{x}_m = x_{j_m}$, for $J_h = \{j_1, \ldots, j_{M_h}\}$ and $m \in [M_h]$. Further, the preconditioner in (5.21) is replaced by

$$B_h B_h^\top = \left(\frac{n}{M} \widehat{K}_{J_h,J_h} A_h^{-1} \widehat{K}_{J_h,J_h} + \lambda_h \widehat{K}_{J_h,J_h} \right)^{-1}. \tag{5.22}$$

This solution can lead to huge computational improvements. Indeed, the total cost of FALKON-BLESS is the sum of computing BLESS and FALKON, corresponding to

$$\mathcal{O}\left(n|J_h|t + (1/\lambda)|J_h|^2 \log n + |J_h|^3\right) \qquad \mathcal{O}(|J_h|^2), \tag{5.23}$$

in time and space respectively, where $|J_h|$ is the size of the set J_H returned by BLESS.

5.3.2 Statistical Properties of FALKON-BLESS

In this section, we state and discuss our second main result, providing an excess risk bound for FALKON-BLESS. Here the population version of the effective dimension (Definition 1 of Chapter 2) plays a key role. Let $\rho_{\mathcal{X}}$ be the marginal measure of ρ on $\mathcal{X}$, let $C : \mathcal{H} \to \mathcal{H}$ be the linear operator defined as follows and $\mathcal{N}(\lambda)$ be the population version of $\widehat{\mathcal{N}}(\lambda)$ as defined in Definition 1 of Chapter 2,

$$\mathcal{N}(\lambda) = \mathrm{Tr}(C(C + \lambda I)^{-1}), \quad \text{with} \quad (Cf)(x') = \int_{\mathcal{X}} k(x', x) f(x) d\rho_{\mathcal{X}}(x),$$

for any $f \in \mathcal{H}, x \in \mathcal{X}$. It is possible to show that $\mathcal{N}(\lambda)$ is the limit of $\widehat{\mathcal{N}}(\lambda)$ as n goes to infinity, see Lemma 30 below taken from [RCR15]. If we assume throughout that,

$$k(x, x') \leq \kappa^2, \quad \forall x, x' \in \mathcal{X}, \tag{5.24}$$

then the operator C is symmetric, positive definite and trace class, and the behavior of $\mathcal{N}(\lambda)$ can be characterized in terms of the properties of the eigenvalues $(\sigma_j)_{j \in \mathbb{N}}$ of C. Indeed as for $\widehat{\mathcal{N}}(\lambda)$, we have that $\mathcal{N}(\lambda) \leq \kappa^2/\lambda$, moreover if $\sigma_j = \mathcal{O}(j^{-\alpha})$, for $\alpha \geq 1$, we have $\mathcal{N}(\lambda) = \mathcal{O}(\lambda^{-1/\alpha})$. Then for larger α, $\mathcal{N}$ is smaller than $1/\lambda$ and faster learning rates are possible, as shown below. We next discuss the properties of the FALKON-BLESS solution denoted by $\widehat{f}_{\lambda,n,t}$.

Theorem 16. *Let $n \in \mathbb{N}$, $\lambda > 0$ and $\delta \in (0, 1]$. Assume that $y \in [-\frac{a}{2}, \frac{a}{2}]$, almost surely, $a > 0$, and denote by $f_{\mathcal{H}}$ a minimizer of the expected risk (see equation* (2.4)*). There exists $n_0 \in \mathbb{N}$, such that for any $n \geq n_0$, if $t \geq \log n$, $\lambda \geq \frac{9\kappa^2}{n} \log \frac{n}{\delta}$, then the following holds with probability at least $1 - \delta$:*

$$\mathcal{R}(\widehat{f}_{\lambda,n,t}) \leq \frac{4a}{n} \; + \; 32 \|f_{\mathcal{H}}\|_{\mathcal{H}}^2 \left(\frac{a^2 \log^2 \frac{2}{\delta}}{n^2 \lambda} + \frac{a\, \widehat{\mathcal{N}}(\lambda) \; \log \frac{2}{\delta}}{n} + \lambda \right).$$

In particular, when $\mathcal{N}(\lambda) = \mathcal{O}(\lambda^{-1/\alpha})$, for $\alpha \geq 1$, by selecting $\lambda_* = n^{-\alpha/(\alpha+1)}$, we have

$$\mathcal{R}(\widehat{f}_{\lambda_*,n,t}) \leq cn^{-\frac{\alpha}{\alpha+1}},$$

where c is given explicitly in the proof.

We comment on the above result discussing the statistical and computational implications.

Statistics. The above theorem provides statistical guarantees in terms of finite sample bounds on the excess risk of FALKON-BLESS, A first bound depends of the number of examples n, the regularization parameter λ and the population effective dimension $\mathcal{N}(\lambda)$. The second bound is derived optimizing λ, and is the same as the one achieved by exact kernel ridge regression which is known to be optimal [CDV07, SHS$^+$09, LRRC18]. Note that improvements under further assumptions are possible and are derived in Section 5.4, see Thm. 19. Here, we comment on the computational properties of FALKON-BLESS and compare it to previous solutions.

Computations. To discuss computational implications, we recall a result from [RCR15] showing that the population version of the effective dimension $\mathcal{N}(\lambda)$ and the effective dimension $\widehat{\mathcal{N}}(\lambda)$ associated to the empirical kernel matrix converge up to constants.

Lemma 30. *Let $\lambda > 0$ and $\delta \in (0, 1]$. When $\lambda \geq \frac{9\kappa^2}{n} \log \frac{n}{\delta}$, then with probability at least $1 - \delta$,*

$$(1/3)\mathcal{N}(\lambda) \leq \widehat{\mathcal{N}}(\lambda) \leq 3\mathcal{N}(\lambda).$$

Recalling the complexity of FALKON-BLESS (5.23), using Thm 16 and Lemma 30, we derive a cost

$$\mathcal{O}\left(n\mathcal{N}(\lambda)\log n + \frac{1}{\lambda}\mathcal{N}(\lambda)^2 \log n + \mathcal{N}(\lambda)^3\right)$$

in time and $\mathcal{O}(\mathcal{N}(\lambda)^2)$ in space, for all n, λ satisfying the assumptions in Theorem 16. These expressions can be further simplified. Indeed, it is easy to see that for all $\lambda > 0$,

$$\mathcal{N}(\lambda) \leq \kappa^2/\lambda, \tag{5.25}$$

so that $\mathcal{N}(\lambda)^3 \leq \frac{\kappa^2}{\lambda}\mathcal{N}(\lambda)^2$. Moreover, if we consider the optimal choice $\lambda_* = \mathcal{O}(n^{-\frac{\alpha}{\alpha+1}})$ given in Theorem 16, and take $\mathcal{N}(\lambda) = \mathcal{O}(\lambda^{-1/\alpha})$, we have $\frac{1}{\lambda_*}\mathcal{N}(\lambda_*) \leq \mathcal{O}(n)$, and therefore $\frac{1}{\lambda}\mathcal{N}(\lambda)^2 \leq \mathcal{O}(n\mathcal{N}(\lambda))$. In summary, for the parameter choices leading to optimal learning rates, FALKON-BLESS has complexity $\widetilde{\mathcal{O}}(n\mathcal{N}(\lambda_*))$, in time and $\widetilde{\mathcal{O}}(\mathcal{N}(\lambda_*)^2)$ in space, ignoring log terms. We can compare this to previous results. In [RCR17] uniform sampling is considered leading to $M \leq \mathcal{O}(1/\lambda)$ and achieving a complexity of $\widetilde{\mathcal{O}}(n/\lambda)$ which is always larger than the one achieved by FALKON in view of (5.25). Approximate leverage scores sampling is also considered in [RCR17] requiring $\widetilde{\mathcal{O}}(n\widehat{\mathcal{N}}(\lambda)^2)$ time and reducing the time complexity of FALKON to $\widetilde{\mathcal{O}}(n\widehat{\mathcal{N}}(\lambda_*))$. Clearly in this case the complexity of leverage scores sampling dominates, and our results provide BLESS as a fix.

5.4 Theoretical Analysis for FALKON-BLESS

In the next section the FALKON algorithm is recalled with some minor changes in the notation with respect to the definition given in Section 4.5.1 of the previous Chapter. The changes in notation are required to better describe the link with the BLESS algorithm. When not explicitly redefined the notation follows the definitions of Section 4.6.

Then it is proved in Theorem 17 that the excess risk of FALKON-BLESS is bounded by the one of Nyström-KRR. In Theorem 18 the learning rates for Nyström-KRR with BLESS are provided. In Theorem 19 a more general version of Theorem 16 is provided, taking into account more refined regularity conditions on the learning problem. Finally the proof of Theorem 16 is derived as a corollary.

5.4.1 Definition of the Algorithm

Definition 9 (Generalized Preconditioner)**.** *Given $\lambda > 0$, $(\widetilde{x}_j)_{j=1}^M \subseteq X$, $M \in \mathbb{N}$ and $A \in \mathbb{R}^{M\times M}$ positive diagonal matrix, we say that B is a* generalized preconditioner*, if*

$$B = \frac{1}{\sqrt{n}} A^{-1/2} Q T^{-1} R^{-1},$$

where $Q \in \mathbb{R}^{M\times q}$ partial isometry with $Q^\top Q = I$ and $q \leq M$, where $T, R \in \mathbb{R}^{q\times q}$ are invertible triangular, and Q, T, R satisfy

$$A^{-1/2} \widehat{K}_{MM} A^{-1/2} = Q T^\top T Q^\top, \quad R^\top R = \frac{1}{M} T T^\top + \lambda I,$$

with $\widehat{K}_{MM} \in \mathbb{R}^{M\times M}$ defined as $(\widehat{K}_{MM})_{ij} = K(\widetilde{x}_i, \widetilde{x}_j)$.

Definition 10 (Generalized FALKON Algorithm)**.** *Let $\lambda > 0$ and $t, n, M \in \mathbb{N}$. Let $(x_i, y_i)_{i=1}^n \subseteq X \times Y$ be the dataset. Given $J \subseteq [n]$ let $\widetilde{X}_J = \cup_{j\in J} x_j$ be the selected Nyström centers and denote by $\{\widetilde{x}_1, \dots, \widetilde{x}_{|J|}\}$ the points in $\widetilde{X}_J$. Let $A \in \mathbb{R}^{|J|\times|J|}$ be a positive diagonal matrix of weights and K the kernel function. Let B, q be as in Definition 9 based on $\widetilde{X}_M$ and A. The* Generalized FALKON estimator *is defined as follows*

$$\widehat{f}_{\lambda,J,A,t} = \sum_{i=1}^{|J|} \alpha_i K(x, \widetilde{x}_i), \quad \textit{with} \quad \alpha = B\beta_t,$$

where $\beta_t \in \mathbb{R}^q$ denotes the vector resulting from t iterations of the conjugate gradient algorithm applied to the following linear system

$$W\beta = b, \quad W = B^\top (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM}) B, \quad b = B^\top \widehat{K}_{nM}^\top y,$$

with $\widehat{K}_{nM} \in \mathbb{R}^{n\times M}$, $(\widehat{K}_{nM})_{ij} = K(x_i, \widetilde{x}_j)$, *and* $\widehat{K}_{MM} \in \mathbb{R}^{M\times M}$, $(\widehat{K}_{MM})_{ij} = K(\widetilde{x}_i, \widetilde{x}_j)$, *and with* $y = (y_1, \dots, y_n) \in \mathbb{R}^n$.

Definition 11 (Standard Nyström Kernel Ridge Regression)**.** *With the same notation as above, the standard Nyström Kernel Ridge Regression estimator is defined as*

$$\widetilde{f}_{\lambda,J} = \sum_{i=1}^{|J|} \alpha_i K(x, \widetilde{x}_i), \quad \textit{with} \quad \alpha = (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM})^\dagger y.$$

5.4.2 Main Results

Here, Theorem 17 proves the excess risk of FALKON-BLESS is bounded by the one of Nyström-KRR. In Theorem 18 the learning rates for Nyström-KRR are provided. In Theorem 19 a more general version of Theorem 16 is provided, taking into account more refined regularity conditions on the learning problem. Finally the proof of Theorem 16 is derived as a corollary.

Let $Z_n = (x_i, y_i)_{i=1}^n$ be a dataset and $J \subseteq \{1, \dots, n\}$ and $A \in \mathbb{R}^{|J|\times|J|}$ positive diagonal matrix. In the rest of this section we denote by $\widehat{f}_{\lambda,J,A,t}$ the FALKON estimator as in Definition 10 trained on Z_n and based on the Nyström centers $\widetilde{X}_M = \cup_{j\in J}\{x_j\}$ and weights A with regularization λ and number of iterations t. Moreover we denote by $\widehat{f}_{\lambda,J}$ the standard Nyström estimator trained on Z_n and based on the Nyström centers $\widetilde{X}_M$.

The following theorem is obtained by combining Lemma 2, 3 and Thm. 1 of [RCR17], with our Proposition 10.

Theorem 17. *Let* $\lambda > 0$, $n \geq 3$, $\delta \in (0, 1]$, $t_{\max} \in \mathbb{N}$. *Let* $Z_n = (x_i, y_i)_{i=1}^n$ *be an i.i.d. dataset. Let* H *and* $(\lambda_h)_{h=0}^H, (M_h)_{h=0}^H, (J_h)_{h=0}^H, (A_h)_{h=0}^H$ *be outputs of Algorithm 3 runned with parameter* $T = 2$.

The following holds with probability $1 - 2\delta$*: for each* $h \in \{0, \dots, H\}$ *such that* $0 < \lambda_h \leq \|C\|$,

$$\mathcal{R}(\widehat{f}_{\lambda_h,J_h,A_h,t})^{1/2} \leq \mathcal{R}(\widetilde{f}_{\lambda_h,J_h})^{1/2} \; + \; 4\widehat{v}\, e^{-t} \sqrt{1 + \frac{9\kappa^2}{\lambda_h n} \log \frac{nHt_{\max}}{\delta}}, \quad \forall t \in \{0, \dots, t_{\max}\},$$

with $\widehat{v}^2 := \frac{1}{n}\sum_{i=1}^n y_i^2$.

Proof. Let $\tau = \delta/(t_{\max}H)$, let $h \in \{1, \dots, H\}$ and let $\overline{A}_h = \frac{n}{|J_h|}A_h$. By Lemma 12 and Lemma 13 in the previous Chapter we have that, when $G_\lambda(\widehat{C}_{J_h,\overline{A}_h}, \widehat{C}_n) < 1$ then the condition number of W_h, that is the preconditioned matrix in Definition 10 with $\lambda = \lambda_h$, is controlled by

$$\text{cond}(W_h) \leq \frac{1 + G_{\lambda_h}(\widehat{C}_{J_h,\overline{A}_h}, \widehat{C}_n)}{1 - G_{\lambda_h}(\widehat{C}_{J_h,\overline{A}_h}, \widehat{C}_n)}.$$

Now, by Proposition 10, we have

$$G_{\lambda_h}(\widehat{C}_{J_h,\overline{A}_h}, \widehat{C}_n) \leq \frac{G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h,\overline{A}_h})}{1 - G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h,\overline{A}_h})}.$$

Combining the two results above, if $G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h,\overline{A}_h}) \leq 1/3$ then

$$\text{cond}(W_h) \leq \frac{1}{1 - 2\, G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h,\overline{A}_h})} \leq 3.$$

Now denote by $E_{h,t}$ the event such that

$$\mathcal{R}(\widehat{f}_{\lambda_h,J_h,A_h,t})^{1/2} \;\leq\; \mathcal{R}(\widetilde{f}_{\lambda_h,J_h})^{1/2} \;+\; 4\widehat{v}\, e^{-t} \sqrt{1 + \frac{9\kappa^2}{\lambda_h n} \log \frac{n}{\tau}}.$$

Since $\text{cond}(W_h) \leq 3$, we have that $\log \frac{\sqrt{\text{cond}(W_h)}+1}{\sqrt{\text{cond}(W_h)}+1} \geq 1$ and so we can apply Theorem 4 in the previous Chapter with parameter $\nu = 1$, obtaining that each $E_{h,t}$, with $t \in \{0, \ldots, t_{\max}\}$ hold with probability $1 - \tau$. So by taking the intersection bound, we know that $E_h := \cap_{t=0}^{t_{\max}} E_{h,t}$ holds with probability $1 - t_{\max}\tau$.

Finally denote by F_H the event: $G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h,\overline{A}_h}) \leq 1/3$ for any $h \in \{0, \ldots, H\}$. Note that Theorem 14 states that, by running Algorithm 3 with $T = 2$, the event F_H holds with probability at least $1 - \delta$.

The desired result correspond to the event $\cap_{h=1}^{H} E_h \cap F_H$ which, by taking the intersection bound, holds with probability at least $1 - \delta - t_{\max} H\tau$. □

5.4.3 Result for Nyström-KRR and BLESS

Referring to the notation introduced in Section 4.6.1, and using the Assumptions 3 and 6 (Section 2.5.2) that will be satisfied by the conditions on Theorem 16, we now prove learning rates for Nyström-KRR with BLESS and then prove the main results of this work.

Theorem 18 (Generalization properties of Nyström-RR using BLESS)**.** *Let $\delta \in (0, 1]$ and $\lambda > 0, n \in \mathbb{N}$. Under Assumption 3, 6, let the Nyström estimator as in Definition 11 and assume that $(J_h)_{h=1}^H, (A_h)_{h=1}^H, (\lambda_h)_{h=1}^H$ is obtained via Algorithm 3 or 4. When $\frac{9\kappa^2}{n} \log \frac{n}{\delta} \leq \lambda \leq \|C\|$, then the following holds with probability $1 - 4\delta$*

$$\mathcal{R}(\widetilde{f}_{\lambda_h,J_h}) \leq 8\|g\|_{\mathcal{H}} \left(\frac{b \log \frac{2}{\delta}}{n\sqrt{\lambda_h}} + \sqrt{\frac{\sigma^2 \widehat{\mathcal{N}}(\lambda_h) \log \frac{2}{\delta}}{n}} + \lambda_h^{1/2+v} \right).$$

Proof. The proof consists in following the decomposition in Thm. 1 of [RCR15], valid under Assumption 6 and using our set J_h to determine the Nyström centers. First note that under Assumption 6, there exists a function $f_{\mathcal{H}} \in \mathcal{H}$, such that $\mathcal{E}(f_{\mathcal{H}}) = \inf_{f \in \mathcal{H}} \mathcal{E}(f)$ (see [CDV07] and also [SHS+09, LRRC18]). According to Thm. 2 of [RCR15], under Assumption 6, we have that

$$\mathcal{R}(\widetilde{f}_{\lambda_h, J_h})^{1/2} \leq q(\underbrace{\mathcal{S}(\lambda_h, n)}_{\text{Sample error}} + \underbrace{\mathcal{C}(M_h)^{1/2+v}}_{\text{Computational error}} + \underbrace{\lambda_h^{1/2+v}}_{\text{Approximation error}}),$$

where $\mathcal{S}(\lambda, n) = \left\| (C + \lambda I)^{-1/2} (\widehat{S}_n^* \widehat{y} - \widehat{C}_n f_{\mathcal{H}}) \right\|$ and $\mathcal{C}(M_h) = \left\| (I - P_{M_h})(C + \lambda I)^{1/2} \right\|^2$ with $P_{M_h} = \widehat{C}_{J_h, I} \widehat{C}^{\dagger}_{J_h, I}$. Moreover $q = \|g\|_{\mathcal{H}} (\beta^2 \vee (1 + \theta\beta))$, $\beta = \left\| (\widehat{C}_n + \lambda I)^{-1/2} (C + \lambda I)^{1/2} \right\|$, $\theta = \left\| (\widehat{C}_n + \lambda I)^{1/2} (C + \lambda I)^{-1/2} \right\|$.

The term $\mathcal{S}(\lambda_h, n)$ is controlled under Assumption 3 by Lemma 4 of the same paper, obtaining

$$\mathcal{S}(\lambda, n) \leq \frac{b \log \frac{2}{\delta}}{n \sqrt{\lambda_h}} + \sqrt{\frac{\sigma^2 \widehat{\mathcal{N}}(\lambda_h) \log \frac{2}{\delta}}{n}},$$

with probability at least $1 - \delta$. The term β is controlled by Lemma 5 of the same paper,

$$\beta \leq 2,$$

with probability $1 - \delta$ under the condition on λ. Moreover

$$\theta^2 = \|(C + \lambda I)^{-1/2} \widehat{C}_n (C + \lambda I)^{-1/2}\| \leq 1 + \|(C + \lambda I)^{-1/2} (\widehat{C}_n - C)(C + \lambda I)^{-1/2}\|,$$

where the last term is bounded by $1/2$ with probability $1 - \delta$ under the same condition on λ, via Proposition 8 and the following Remark 1 of the same paper.

Now we study the term $\mathcal{C}(M_h)$ that is the one depending on the result of BLESS. First note that, since $\mathrm{diag}(A_h) > 0$, then

$$P_{M_h} = \widehat{C}_{J_h, I} \widehat{C}^{\dagger}_{J_h, I} = \widehat{C}_{J_h, \overline{A}_h} \widehat{C}^{\dagger}_{J_h, \overline{A}_h}.$$

By applying Proposition 3 and Proposition 7 of the same paper, the following holds

$$\mathcal{C}(M_h) \leq \frac{\lambda_h}{1 - G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h, \overline{A}_h})}, \leq 2\lambda_h,$$

with probability at least $1 - \delta$, where we applied Theorem 14-(c) and Theorem 15-(c), which control exactly $G_{\lambda_h}(\widehat{C}_n, \widehat{C}_{J_h, \overline{A}_h})$ and prove it to be smaller than $1/2$ in high probability.

Finally by taking the intersection bound of the events above, we have

$$\mathcal{R}(\widetilde{f}_{\lambda_h, J_h})^{1/2} \leq 4\|g\|_{\mathcal{H}} \left(\frac{b \log \frac{2}{\delta}}{n \sqrt{\lambda_h}} + \sqrt{\frac{\sigma^2 \widehat{\mathcal{N}}(\lambda_h) \log \frac{2}{\delta}}{n}} + 2\lambda_h^{1/2+v} \right),$$

with probability $1 - 4\delta$. □

Theorem 19 (Generalization properties of learning with FALKON-BLESS)**.** *Let $\delta \in (0,1]$ and $\lambda > 0, n \geq 3$, $t_{\max} \in \mathbb{N}$. Let $Z_n = (x_i, y_i)_{i=1}^n$ be an i.i.d. dataset. Let H and M_H, J_H, A_H be outputs of Algorithm 3 runned with parameter $T = 2$. Let $y \in [-a/2, a/2]$ almost surely, with $a > 0$. Under Assumption 6, Let $\lambda > 0$, $n \geq 3$, $\delta \in (0,1]$, when $\frac{9\kappa^2}{n} \log \frac{n}{\delta} \leq \lambda \leq \|C\|$, then the following holds with probability $1 - 6\delta$*

$$\mathcal{R}(\widehat{f}_{\lambda,J_H,A_H,t}) \leq 4a\, e^{-t} + 32\|g\|_{\mathcal{H}}^2 \left(\frac{a^2 \log^2 \frac{2}{\delta}}{n^2 \lambda} + \frac{a\widehat{\mathcal{N}}(\lambda) \log \frac{2}{\delta}}{n} + 2\lambda^{1+2r} \right), \quad \forall t \in \{0, \dots, t_{\max}\},$$

Proof. The result is obtained by combining Theorem 17, with Theorem 18 and noting that when $y \in [-a/2, a/2]$ almost surely, then it satisfies Assumption 3 with $b, \sigma \leq a$. □

5.4.4 Proof of Theorem 16

Proof. The result is a corollary of Theorem 19, where we assumed only the existence of $f_{\mathcal{H}}$. This correspond to assume Assumption 6, with $r = 1/2$ and $g = f_{\mathcal{H}}$ (see [CDV07]). □

5.5 Experiments

We now present some experimental results. We first show that the leverage score obtained by the BLESS and BLESS-R are accurate, and then we show which effect they have in a supervised learning problem with the FALKON-BLESS algorithm.

5.5.1 Leverage Scores Accuracy

We first study the accuracy of the leverage scores generated by BLESS and BLESS-R, comparing SQUEAK [CLV17a] and Recursive-RLS (RRLS) [MM17]. We begin by uniformly sampling a subsets of $n = 7 \times 10^4$ points from the SUSY dataset [BSW14], and computing the exact leverage scores $\ell(i, \lambda)$ using a Gaussian Kernel with $\sigma = 4$ and $\lambda = 10^{-5}$, which is at the limit of our computational feasibility. We then run each algorithm to compute the approximate leverage scores $\widetilde{\ell}_{J_H}(i, \lambda)$, and we measure the accuracy of each method using the ratio $\widetilde{\ell}_{J_H}(i, \lambda)/\ell(i, \lambda)$ (R-ACC). The final results are presented in Figure 5.1. On the left side for each algorithm we report runtime, mean R-ACC, and the 5^{th} and 95^{th} quantile, each averaged over the 10 repetitions. On the right side a box-plot of the R-ACC. As shown in Figure 5.1 BLESS and BLESS-R achieve the same optimal accuracy of SQUEAK with just a fraction of time. Note that despite our best efforts, we could not obtain high-accuracy results for RRLS (maybe a wrong constant in the original implementation). However note that RRLS is computationally demanding compared

	Time	R-ACC	5^{th}/ 95^{th} quant
BLESS	**17**	1.06	0.57 / 2.03
BLESS-R	**17**	1.06	0.73 / 1.50
SQUEAK	52	1.06	0.70 / 1.48
Uniform	-	1.09	0.22 / 3.75
RRLS	235	1.59	1.00 / 2.70

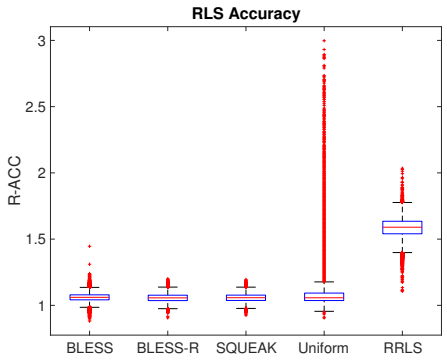

Fig. 5.1: Leverage scores relative accuracy for $\lambda = 10^{-5}$, $n = 70\,000$, $M = 10\,000$, 10 repetitions.

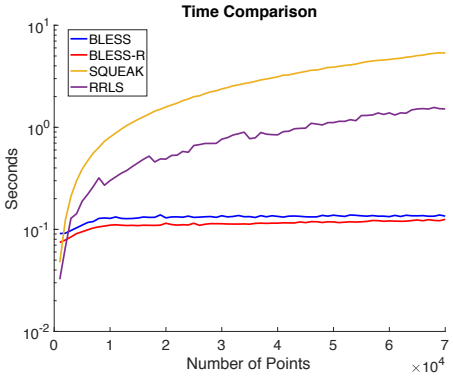

Fig. 5.2: Runtimes with $\lambda = 10^{-3}$ and n increasing

Fig. 5.3: C-err at 5 iterations for varying λ_{falkon}

to BLESS, being orders of magnitude slower, as expected from the theory. Finally, although uniform sampling is the fastest approach, it suffers from much larger variance and can over or under-estimate leverage scores by an order of magnitude more than the other methods, making it more fragile for downstream applications.
In Fig. 5.2 we plot the runtime cost of the compared algorithms as the number of points grows from $n = 1000$ to 70000, this time for $\lambda = 10^{-3}$. We see that while previous algorithms' runtime grows near-linearly with n, BLESS and BLESS-R run in a constant $1/\lambda$ runtime, as predicted by the theory.

5.5.2 BLESS for Supervised Learning

We study the performance of FALKON-BLESS and compare it with the original FALKON [RCR17] where an equal number of Nyström centres are sampled uniformly at random (FALKON-UNI). We take from [RCR17] the two biggest datasets and their best hyper-parameters for the FALKON algorithm.

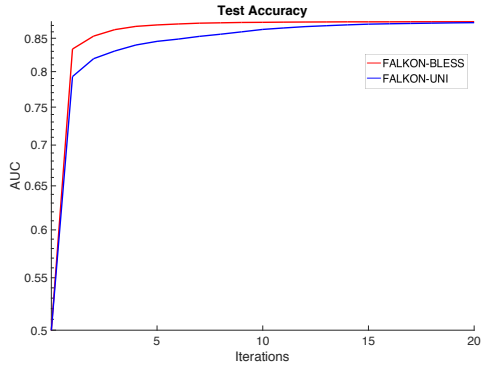

Fig. 5.4: AUC per iteration of the SUSY dataset

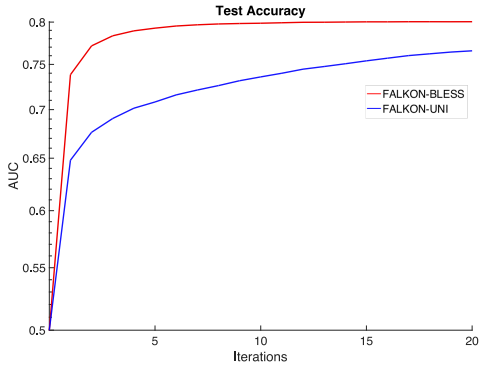

Fig. 5.5: AUC per iteration of the HIGGS dataset

We noticed that it is possible to achieve the same accuracy of FALKON-UNI, by using λ_{bless} for BLESS and λ_{falkon} for FALKON with $\lambda_{bless} \gg \lambda_{falkon}$, in order to lower the $\widehat{\mathcal{N}}$ and keep the number of Nyström centres low. For the SUSY dataset we use a Gaussian Kernel with $\sigma = 4, \lambda_{falkon} = 10^{-6}, \lambda_{bless} = 10^{-4}$ obtaining $M_H \simeq 10^4$ Nyström centres. For the HIGGS dataset we use a Gaussian Kernel with $\sigma = 22, \lambda_{falkon} = 10^{-8}, \lambda_{bless} = 10^{-6}$, obtaining $M_H \simeq 3 \times 10^4$ Nyström centres. We then sample a comparable number of centers uniformly for FALKON-UNI. Looking at the plot of their AUC at each iteration (Fig.5.4, 5.5) we observe that FALKON-BLESS converges much faster than FALKON-UNI. For the SUSY dataset (Figure 5.4) 5 iterations of FALKON-BLESS (160 seconds) achieve the same accuracy of 20 iterations of FALKON-UNI (610 seconds). Since running BLESS takes just 12 secs. this corresponds to a $\sim 4\times$ speedup. For the HIGGS dataset 10 iterations of FALKON-BLESS (with BLESS requiring 1.5 minutes, for a total of 1.4 hours) achieve better accuracy of 20 iterations of FALKON-UNI (2.7 hours). Additionally we observed that FALKON-BLESS is more stable than FALKON-UNI w.r.t. λ_{falkon}, σ. In Figure 5.3 the classification error after 5 iterations of FALKON-BLESS and FALKON-UNI over the SUSY dataset ($\lambda_{bless} = 10^{-4}$). We notice that FALKON-BLESS has a wider optimal region (95% of the best error) for the regulariazion parameter ($[1.3 \times 10^{-3}, 4.8 \times 10^{-8}]$) w.r.t. FALKON-UNI ($[1.3 \times 10^{-3}, 3.8 \times 10^{-6}]$).

Chapter 6

Kernelized Bandit Optimization

In the previous chapters, we considered the statistical learning setting. In that setting, given input/output pairs, the goal is to learn the unknown function that determines the relation between input and output over all possible inputs. In details the goal is to learn an unknown function f from a set of provided noisy samples $\{x_i, y_i\}_{i=1}^n$ generated from this function as $y_i = f(x_i) + \varepsilon$, where ε is a form of noise. The learned function is good if for any new input point x_{new}, the corresponding output y_{new} can be predicted.

In this chapter, we consider instead a setting known as *optimization under bandit feedback* or *bandit optimization*, where the main goal is not to learn a good estimator of the function f on the entire domain but to maximize the function f. Differently from the statistical learning setting a set of points is not provided from the beginning. Instead, they need to be collected according to an optimal strategy, considering that sampling one input/output pair comes with a cost proportional to the distance of the output from the actual maximum of the function.

6.1 Bandit Optimization

In bandit optimization we assume there exists an input space $\mathcal{A}$ also known as set of arms. For the sake of simplicity, we are going to assume that $\mathcal{A} = \{x_i\}_{i=1}^A$ is a fixed finite set of A points in $\mathbb{R}^d$. This assumption can be relaxed and we are going to discuss how it can be relaxed later in Section 7.2.

We define with $f : \mathcal{A} \to \mathbb{R}$ a reward function that we wish to maximize over the set of arms $\mathcal{A}$. This reward function is unknown and noisy, and differently from the statistical learning setting, we are not provided with input/output pairs to approximate the reward function. What we can do instead is to collect input/output pairs over $T \in \mathbb{N}_+$ iterations. The number of iterations T can potentially be infinite but, for the sake of simplicity, we consider it finite in the following. For

each iteration $t \in [T]$, after choosing an arm $x_t \in \mathcal{A}$, we can evaluate the reward function at the chosen point to collect the corresponding output $y_t = f(x_t) + \eta_t$ (also called reward), where η_t is a zero-mean noise. However we further assume that for each evaluation of the reward function we are going to suffer a cost proportional to the distance between the output of the chosen input $y_t = f(x_t) + \eta_t$ and the output of the best arm $y_\star = f(x_\star) + \eta_t$. The best arm $x_\star$ is the input that in expectation returns the highest reward. Without this assumption on the cost of sampling, the problem could be solved trivially with techniques described in the previous chapters. Indeed one could query as many input/output pairs as desired recovering the statistical learning setting. To measure the suffered cost over T iterations we define the following quantity known as cumulative regret

$$R_T = \sum_{t=1}^{T} f(x_\star) - f(x_t), \tag{6.1}$$

where x_t with $t \in [T]$ is the input chosen at time t and $x_\star = \operatorname{argmax}_{x_i \in \mathcal{A}} f(x_i)$.

The goal in this setting is to choose the best possible sequence of points x_t in order to minimize the cumulative regret. In particular, the objective of a so-called *no-regret* algorithm is to have R_T/T going to zero as fast as possible when T grows. In bandit optimization, the learning process can be described as a sequential game between a learner and an environment. In this game setting evaluating the reward function f with respect to an arm x_i is often referred to as *pulling the arm* x_i. Before the game starts the learner chooses an optimization strategy and a prior on the reward function f. Then for each step $t \in [T]$ of the game, the learner

(1) chooses an arm $x_t \in \mathcal{A}$ according to the optimization strategy,

(2) queries the environment in order to receive the corresponding output $y_t = f(x_t) + \eta_t$, where η_t is a zero-mean noise,

(3) updates its model of the problem based on the observed output.

Different optimization strategies and different priors define different bandit optimization algorithms. In the following, we are interested in studying algorithms that use non-linear non-parametric priors over f. In particular, we consider Gaussian Processes (GP) as priors [RW06]. Using a GP is similar to use a KRR estimator (see (2.32) in Section 2.3) as prior of f, adding the option to estimate the uncertainty of the estimator. We define a GP in mode details in Section 6.3. The optimization strategy determines the exploration-exploitation trade-off of the sampling process. On the one hand, at any point in time t, one would want to choose an arm x_t that has produced the biggest reward until time t. On the other hand, one would want to explore other options that may look inferior or unexplored, but that may produce bigger rewards in future iterates. Between the many options like *explore-then-commit* [Rob52] or *Thompson sampling* [Tho33] we consider in the following the *upper confidence bound principle* [LR85].

6.2 Upper Confident Bound

The upper confidence bound (UCB) strategy follows the so-called *optimism in face of uncertainty* principle. This principle states that the learner should choose an arm as if the environment is as nice as plausibly possible. This principle assigns to each arm $x_i \in \mathcal{A}$ a value called the UCB such that in high probability this value is an overestimate of $f(x_i)$, the true unknown reward of arm x_i. The learner then proceeds to pick at each iteration the arm x_t with the highest UCB and uses the observed output y_t to update the UCBs. If the UCBs are properly built, it can be proved that the learning algorithm will converge to choose mainly the optimal arm.

To formalize this strategy, let $\mu_t : \mathcal{A} \to \mathbb{R}$ be the function that given an arm $x_i \in \mathcal{A}$ returns the empirical mean of the reward of the arm x_i, based on the observations collected until time t. Let $\Delta_t : \mathcal{A} \to \mathbb{R}_+$ be a non-negative function used to overestimate the mean. We can express the upper confidence bounds as a function $u_t : \mathcal{A} \to \mathbb{R}$ defined as

$$u_t(x_i) = \mu_t(x_i) + \Delta_t(x_i). \tag{6.2}$$

The exact definitions of the μ_t and Δ_t functions depend on the assumptions that are made on the problem. The mean μ_t depends in particular on the prior over the reward function f. The function Δ_t needs to be properly chosen to allow the learning algorithm to achieve low regret guarantees. Further, these two functions both depend on the observations collected until time t. Every time a new pair $\{x_t, y_t\}$ gets collected their definition changes. The function μ_t will get closer with time to the true mean reward for each arm, and the overestimate given by Δ_t will fade. In particular, the value $\Delta_t(x_i)$ decreases proportionally to the number of times the arm x_i is pulled.

Using the UCB principle, the learning strategy can be re-written as follows.
For each $t \in [T]$:

(1) select an arm $x_t = \operatorname{argmax}_{x_i \in \mathcal{A}} u_t(x_i)$,

(2) observe the corresponding output $y_t = f(x_t) + \eta_t$,

(3) updates μ_t and Δ_t based on the observed y_t.

We will see in details in Section 6.4 how the UCBs u_t are built in the GP-UCB algorithm [SKKS10].

6.3 Gaussian Process

We give in the following the formal definition of a Gaussian process.

A *Gaussian process* $\mathrm{GP}(\mu, k)$ is a generalization of the Gaussian distribution to a space of functions and it is defined by a mean function $\mu : \mathcal{A} \to \mathbb{R}$ and a covariance function $k : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$. We consider zero-mean $\mathrm{GP}(0, k)$ priors and bounded covariance $k(x_i, x_i) \leq \kappa^2$ for all $x_i \in \mathcal{A}$.

An important property of Gaussian processes is that if we combine a prior $f \sim \mathrm{GP}(0, k)$ and assume that the observation noise is zero-mean Gaussian (i.e., $\eta_t \sim \mathcal{N}(0, \xi^2)$), then the posterior distribution of f conditioned on a set of observations $\{(x_s, y_s)\}_{s=1}^{t}$ is also a GP. More precisely, if $X_t = [x_1, \ldots, x_t]^\intercal \in \mathbb{R}^{t \times d}$ is the matrix with all arms selected so far and $y_t = [y_1, \ldots, y_t]^\intercal$ the corresponding observations, then the posterior is still a GP and the mean and variance of the function at a test point x are defined as

$$\mu_t\left(x \mid X_t, y_t\right) = \widehat{k}_t(x)^\intercal (\widehat{K}_t + \lambda I)^{-1} y_t, \tag{6.3}$$

$$\sigma_t^2\left(x \mid X_t\right) = k(x, x) - \widehat{k}_t(x)^\intercal (\widehat{K}_t + \lambda I)^{-1} \widehat{k}_t(x), \tag{6.4}$$

where $\lambda = \xi^2$, $\widehat{K}_t \in \mathbb{R}^{t \times t}$ is the matrix $[\widehat{K}_t]_{i,j} = k(x_i, x_j)$ constructed from all pairs x_i, x_j in X_t, and $\widehat{k}_t(x) = [k(x_1, x), \ldots, k(x_t, x)]^\intercal$. Notice that $\widehat{k}_t(x)$ can be seen as an *embedding* of an arm x represented using by the arms $x_1, \ldots, x_t$ observed so far.

6.4 GP-UCB

GP-UCB is popular no-regret algorithm for optimization under bandit feedback and was introduced by [SKKS10] for Gaussian process optimization.

The GP-UCB algorithm uses a Gaussian process $\mathrm{GP}(0, k)$ as a prior for f. Inspired by the optimism in face of uncertainty principle, at each time step t, GP-UCB uses the posterior GP to compute the mean and variance of an arm x_i and obtain the score

$$u_t(x_i) = \mu_t(x_i) + \beta_t \sigma_t(x_i), \tag{6.5}$$

where we use the short-hand notation $\mu_t(\cdot) = \mu\left(\cdot \mid X_t, y_t\right)$ and $\sigma_t(\cdot) = \sigma\left(\cdot \mid X_t\right)$ to indicate the mean and the variance that depend on the points observed until time t.

Finally, GP-UCB chooses the maximizer $x_{t+1} = \mathrm{argmax}_{x_i \in \mathcal{A}}\, u_t(x_i)$ as the next arm to evaluate. According to the score u_t, an arm x is likely to be selected if it has high mean reward μ_t *or* high variance σ_t, i.e., its estimated reward $\mu_t(x)$ is very uncertain. As a result, selecting the arm x_{t+1} with the largest score trades off between collecting (estimated) large reward (*exploitation*) and improving the accuracy of the posterior (*exploration*).

The parameter β_t balances between these two objectives and must be properly tuned to guarantee low regret. [SKKS10] proposes different approaches for tuning β_t, depending on the assumptions on f and $\mathcal{A}$.

Tuning correctly β_t it is possible to prove (see [SKKS10] for the Bayesian analysis and [CG17] for the frequentist analysis) that the regret of GP-UCB after T iterations can be controlled as

$$R_T \leq \mathcal{O}(\sqrt{T}\gamma_T), \tag{6.6}$$

where γ_T is the quantity known as *maximum information gain* that can be further controlled based on the assumptions of the problem (we define in details this quantity in the next chapter).

While GP-UCB is interpretable, simple to implement and provably achieves low regret, it is computationally expensive. In particular, computing $\sigma_t(x)$ has a complexity at least $\Omega(t^2)$ for the matrix-vector product $(\widehat{K}_{t-1} + \xi^2 I)^{-1}\widehat{k}_{t-1}(x)$. Multiplying this complexity by T iterations and A arms results in an overall $\mathcal{O}(AT^3)$ cost, which does not scale to a large number of iterations T.

Chapter 7

Gaussian Process Optimization with Adaptive Sketching

In this chapter, we present BKB (budgeted kernelized bandit), a new approximate Gaussian process (GP) algorithm for optimization under bandit feedback that achieves near-optimal regret (and hence near-optimal convergence rate) with near-constant per-iteration complexity and no assumption on the input space or covariance of the GP.

We have seen in the previous chapter that GP-UCB [SKKS10] is a a well studied Bayesian approach for the optimization of black-box functions. Despite its effectiveness in simple problems, GP-UCB hardly scale to high-dimensional functions, as its per-iteration time and space cost is at least quadratic in the number of dimensions d and iterations T. Given a set of A alternatives to choose from, the overall runtime $\mathcal{O}(AT^3)$ is prohibitive.

BKB combines GP-UCB with randomized matrix sketching based on leverage score sampling, and we prove that randomly sampling inducing points based on their posterior variance gives an accurate low-rank approximation of the GP, preserving variance estimates and confidence intervals. As a consequence, BKB does not suffer from variance starvation, an important problem faced by many previous sparse GP approximations [WGKJ18]. Moreover, we show that our procedure selects at most $\widetilde{\mathcal{O}}(\widehat{\mathcal{N}})$ points, where $\widehat{\mathcal{N}}$ is the effective dimension of the explored space, which is typically much smaller than both d and t. This greatly reduces the dimensionality of the problem, thus leading to a $\mathcal{O}(TA\widehat{\mathcal{N}}^2)$ runtime and $\mathcal{O}(A\widehat{\mathcal{N}})$ space complexity.

7.1 Budgeted Kernel Bandits

In this section, we introduce the BKB (*budgeted kernel bandit*) algorithm, a novel efficient approximation of GP-UCB, and we provide guarantees for its computational complexity. The

analysis in Section 7.1.3 shows that BKB can be tuned to significantly reduce the complexity of GP-UCB with a negligible impact on the regret. We begin by introducing the first two major contributions: an approximation of the GP-UCB scores supported only by a small subset $\mathcal{I}_t$ of *inducing points*, and a method to *incrementally and adaptively* construct an accurate subset $\mathcal{I}_t$.

7.1.1 The algorithm

The main complexity bottleneck to compute the scores in Equation (6.5) is due to the fact that after t steps, the posterior GP is supported on *all* t previously seen arms. As a consequence, evaluating Equations (6.3) and (6.4) requires computing a t dimensional vector $\widehat{k}_t(x)$ and $t \times t$ matrix $\widehat{K}_t$ respectively. To avoid this dependency we restrict both $\widehat{k}_t$ and $\widehat{K}_t$ to be supported on a *subset* $\mathcal{I}_t$ of m arms. This approach is a case of the sparse Gaussian process approximation [QCRW07], or equivalently, linear bandits constrained to a subspace [KCCB19].

Approximated GP-UCB scores. Consider a subset of arm $\mathcal{I}_t = \{x_i\}_{i=1}^m$ and let $X_{\mathcal{I}_t} \in \mathbb{R}^{m \times d}$ be the matrix with all arms in $\mathcal{I}_t$ as rows. Let $\widehat{K}_{\mathcal{I}_t} \in \mathbb{R}^{m \times m}$ be the matrix constructed by evaluating the covariance k between any two pairs of arms in $\mathcal{I}_t$ and $\widehat{k}_{\mathcal{I}_t}(x) = [k(x_1, x), \dots, k(x_m, x)]^\intercal$. The Nyström embedding $z_t(\cdot)$ associated with subset $\mathcal{I}_t$ is defined as the mapping[1]

$$z_t(\cdot) = \left(\widehat{K}_{\mathcal{I}_t}^{1/2}\right)^+ \widehat{k}_{\mathcal{I}_t}(\cdot) : \mathbb{R}^d \to \mathbb{R}^m,$$

where $(\cdot)^+$ indicates the pseudo-inverse. We denote with $Z_t(X_t) = [z_t(x_1), \dots, z_t(x_t)]^\intercal \in \mathbb{R}^{t \times m}$ the associated matrix of points and we define $V_t = Z_t(X_t)^\intercal Z_t(X_t) + \lambda I$. Then, we approximate the posterior mean, variance, and UCB for the value of the function at x_i as

$$\begin{aligned}
\widetilde{\mu}_t(x_i) &= z_t(x_i)^\intercal V_t^{-1} Z_t(x_i)^\intercal y_t, \\
\widetilde{\sigma}_t^2(x_i) &= \frac{1}{\lambda}\Big(k(x_i, x_i) - z_t(x_i)^\intercal Z_t(X_t)^\intercal Z_t(X_t) V_t^{-1} z_t(x_i)\Big), \\
\widetilde{u}_t(x_i) &= \widetilde{\mu}_t(x_i) + \widetilde{\beta}_t \widetilde{\sigma}_t(x_i),
\end{aligned} \tag{7.1}$$

where $\widetilde{\beta}_t$ is appropriately tuned to achieve small regret in the theoretical analysis of Section 7.1.3. Finally, at each time step t, BKB selects arm $\widetilde{x}_{t+1} = \operatorname{argmax}_{x_i \in \mathcal{A}} \widetilde{u}_t(x_i)$.

Notice that in general, $\widetilde{\mu}_t$ and $\widetilde{\sigma}_t$ do *not* correspond to any GP posterior. In fact, if we were simply replacing the $k(x_i, x_i)$ in the expression of $\widetilde{\sigma}_t^2(x_i)$ by its value in the Nyström embedding, i.e., $z_t(x_i)^\intercal z_t(x_i)$, then we would recover a sparse GP approximation known as the *subset of regressors*. Using $z_t(x_i)^\intercal z_t(x_i)$ is known to cause *variance starvation*, as it can severely underestimate the variance of a test point x_i when it is far from the points in $\mathcal{I}_t$. Our formulation of

[1] Recall that in the exact version, $\widehat{k}_t(x)$ can be seen as an embedding of any arm x into the space induced by all the t arms selected so far, i.e., using all selected points as inducing points.

$\widetilde{\sigma}_t$ is known in Bayesian world as the *deterministic training conditional* (DTC), where it is used as a heuristic to prevent variance starvation. However, DTC does *not* correspond to a GP since it violates consistency [QCRW07]. In this work, we justify this approach rigorously, showing that it is crucial to prove approximation guarantees necessary both for the optimization process and for the construction of the set of inducing points.

Algorithm 5: BKB

Data: Arm set $\mathcal{A}$, $\overline{q}$, $\{\beta_t\}_{t=1}^{T}$
Result: Arm choices $\mathcal{D}_T = \{(\widetilde{x}_t, y_t)\}$

```
Select uniformly at random x_1 and observe y_1;
Initialize I_1 = {x_1};
for t = {1, ..., T − 1} do
    Compute μ̃_t(x_i) and σ̃²_t(x_i) for all x_i ∈ A;
    Select x̃_{t+1} = argmax_{x_i ∈ A} ũ_t(x_i) (Eq. 7.1);
    for i = {1, ..., t + 1} do
        Set p̃_{t+1,i} = q̄ · σ̃²_t(x̃_i);
        Draw q_{t+1,i} ∼ Bernoulli(p̃_{t+1,i});
        If q_{t+1} = 1 include x̃_i in I_{t+1};
    end
end
```

Choosing the inducing points. A critical aspect to effectively keep the complexity of BKB low while still controlling the regret is to carefully choose the inducing points to include in the subset $\mathcal{I}_t$. As the complexity of computing $\widetilde{u}_t$ scales with the size m of $\mathcal{I}_t$, a smaller set gives a faster algorithm. Conversely, the difference between $\widetilde{\mu}_t$ and $\widetilde{\sigma}_t$ and their exact counterparts depends on the accuracy of the embedding z_t, which increases with the size of the set $\mathcal{I}_t$. Moreover, even for a fixed m, the quality of the embedding greatly depends on *which* inducing points are included. For instance, selecting the same arm as inducing point twice, or two co-linear arms, does not improve accuracy as the embedding space does not change. Finally, we need to take into account two important aspects of sequential optimization when choosing $\mathcal{I}_t$. First, we need to focus our approximation more on regions of $\mathcal{A}$ that are relevant to the objective (i.e., high-reward arms). Second, as these regions change over time, we need to keep adapting the composition and size of $\mathcal{I}_t$ accordingly.

To address the first objective, we choose to construct $\mathcal{I}_t$ by randomly subsampling only out of the set of arms $\widetilde{X}_t$ evaluated so far. This set will naturally focus on high-reward arms, as low-reward arms will be selected increasingly less often and will become a small minority of $\widetilde{X}_t$. To address the change in focus over time, arms are selected for inclusion in $\mathcal{I}_t$ with a probability proportional to their posterior variance σ_t at step t, which changes accordingly. We report the selection procedure in Algorithm 5, with the complete BKB algorithm.

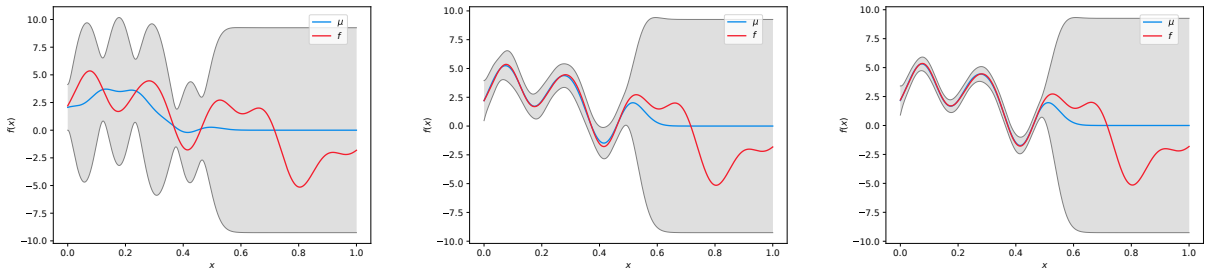

Fig. 7.1: We simulate a GP on $[0, 1] \in \mathbb{R}$ using Gaussian kernel with bandwidth $\sigma^2 = 100$. We draw f from the GP and give to BKB $t \in \{6, 63, 215\}$ evaluations sampled uniformly in $[0, 0.5]$. We plot f and $\widetilde{\mu}_t \pm 3\widetilde{\sigma}_t$.

We initialize $\mathcal{I}_1 = \{\widetilde{x}_1\}$ by selecting an arm uniformly at random. At each step t, after selecting $\widetilde{x}_{t+1}$, we must regenerate $\mathcal{I}_t$ to reflect the changes in $\widetilde{X}_{t+1}$ (i.e., resparsify the GP approximation). Ideally, we would sample each arm in $\widetilde{X}_{t+1}$ proportionally to σ^2_{t+1}, but this would be too computationally expensive. Therefore, we apply two approximations. First we approximate σ^2_{t+1} with σ^2_t. This is equivalent to ignoring the last arm and does not significantly impact the accuracy. We can then replace σ^2_t with $\widetilde{\sigma}^2_t$ which can be computed efficiently, and in practice we simply cache and reuse the $\widetilde{\sigma}^2_t$ already computed when constructing Equation (7.1). Finally, given a parameter $\overline{q} \geq 1$, we set our approximate inclusion probability as $\widetilde{p}_{t+1,i} = \overline{q}\widetilde{\sigma}^2_t(\widetilde{x}_s)$. The $\overline{q}$ parameter is used to increase the inclusion probability in order to boost the overall success probability of the approximation procedure at the expense of a small increase in the size of $\mathcal{I}_{t+1}$. Given $\widetilde{p}_{t+1,i}$, we start from an empty $\mathcal{I}_{t+1}$ and iterate over all $\widetilde{x}_i$ for $i \in [t+1]$ drawing $q_{t+1,i}$ from a Bernoulli distribution with probability $\widetilde{p}_{t+1,i}$. If $q_{t+1,i} = 1$, $\widetilde{x}_i$ is included in $\mathcal{I}_{t+1}$.

Notice that while constructing $\mathcal{I}_t$ based on σ^2_t is a common heuristic for sparse GPs, it has not been yet rigorously justified. In the next section, we show that this posterior variance sampling approach is equivalent to λ-ridge leverage score (RLS) sampling [AM15b]. We leverage the known results from this field to prove both accuracy and efficiency guarantees for our selection procedure.

7.1.2 Complexity analysis

Let $m_t = |\mathcal{I}_t|$ be the size of the set $\mathcal{I}_t$ at step t. At each step, we first compute the embedding $z_t(x_i)$ of all arms in $\mathcal{O}(Am_t^2 + m_t^3)$ time, which corresponds to one inversion of $\widehat{K}^{1/2}_{\mathcal{I}_t}$ and the matrix-vector product specific to each arm. We then rebuild the matrix V_t from scratch using all the arms observed so far. In general, it is sufficient to use counters to record the arms pulled so far, rather than the full list of arms, so that V_t can be constructed in $\mathcal{O}(\min\{t, A\}m_t^2)$ time. Then, the inverse V_t^{-1} is computed in $\mathcal{O}(m_t^3)$ time. We can now efficiently compute $\widetilde{\mu}_t$, $\widetilde{\sigma}_t$, and $\widetilde{u}_t$ for all arms in $\mathcal{O}(Am_t^2)$ time reusing the embeddings and V_t^{-1}. Finally, computing all $q_{t+1,i}$s

and $\mathcal{I}_{t+1}$ takes $\mathcal{O}(\min\{t+1, A\})$ time using the estimated variances $\widetilde{\sigma}_t^2$. As a result, the per-step complexity is of order $\mathcal{O}\big((A + \min\{t, A\})m_T^2\big)$.[2] Space-wise, we only need to store the embedded arms and V_t matrix, which takes at most $\mathcal{O}(Am_T)$ space.

The size of $\mathcal{I}_T$. The size m_t of $\mathcal{I}_t$ can be expressed using the $q_{t,i}$ r.v. as the sum $m_t = \sum_{i=1}^t q_{t,i}$. In order to provide a bound on the total number of inducing points, which directly determines the computational complexity of BKB, we go through three major steps.

The first is to show that w.h.p. m_t is close to the sum $\sum_{i=1}^t \widetilde{p}_{t,i} = \sum_{i=1}^t \overline{q}\widetilde{\sigma}_t^2(\widetilde{x}_i)$, i.e. close to the sum of the probabilities we used to sample each $q_{t,i}$. However, the different $q_{t,i}$ are *not independent* and each $\widetilde{p}_{t,i}$ is itself a r.v. Nonetheless all $q_{t,i}$ are conditionally independent given the previous $t-1$ steps, and this is sufficient to obtain the result.

The second and a more complex step is to guarantee that the random sum $\sum_{i=1}^t \widetilde{\sigma}_t^2(\widetilde{x}_i)$ is close to $\sum_{i=1}^t \sigma_t^2(\widetilde{x}_i)$ and, at a lower level, that each individual estimate $\widetilde{\sigma}_t^2(\cdot)$ is close to $\sigma_t^2(\cdot)$. To achieve this we exploit the connection between ridge leverage scores and posterior variance σ_t^2. In particular, we show that the variance estimator $\widetilde{\sigma}_t^2(\cdot)$ used by BKB is a variation of the RLS estimator of [CLV17b] for RLS sampling. As a consequence, we can transfer the strong accuracy and size guarantees of RLS sampling to our optimization setting (see Section 7.3.2). Note that anchoring the probabilities to the RLS (i.e., the sum of the posterior variances) means that the size of $\mathcal{I}_t$ naturally follows the effective dimension of the arms pulled so far. This strikes an adaptive balance between decreasing each individual probability to avoid $\mathcal{I}_t$ growing too large, while at the same time automatically increasing the effective degrees of freedom of the sparse GP when necessary.

The first two steps lead to $m_t \approx \sum_{i=1}^t \sigma_i^2(\widetilde{x}_i)$, for which we need to derive a more explicit bound. In the GP analyses, this quantity is bounded using the maximal information gain γ_T after T rounds. For this, let $X_{\mathcal{A}} \in \mathbb{R}^{A\times d}$ be the matrix with all arms as rows, $\mathcal{D}$ a subset of these rows, potentially with duplicates, and $\widehat{K}_{\mathcal{D}}$ the associated kernel matrix. Then, [SKKS10] define

$$\gamma_T = \max_{\mathcal{D}\subset\mathcal{A}:|\mathcal{D}|=T} \tfrac{1}{2}\log\det(\widehat{K}_{\mathcal{D}}/\lambda + I), \tag{7.2}$$

and show that $\sum_{i=1}^t \sigma_i^2(\widetilde{x}_i) \leq \gamma_t$, and that γ_T itself can be bounded for specific $\mathcal{A}$ and kernel functions, e.g. $\gamma_T \leq \mathcal{O}(\log(T)^{d+1})$ for Gaussian kernels. Using the equivalence between RLS and posterior variance σ_t^2, we can also relate the posterior variance $\sigma_t^2(\widetilde{x}_i)$ of the evaluated arms to the so-called GP's *effective dimension* $\widehat{\mathcal{N}}$ or degrees of freedom

$$\widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) = \sum_{i=1}^t \sigma_t^2(\widetilde{x}_i) = \mathrm{Tr}(\widehat{K}_T(\widehat{K}_T + \lambda I)^{-1}), \tag{7.3}$$

using the following inequality by [CLV17d],

$$\log\det\left(\widehat{K}_T/\lambda + I\right) \leq \mathrm{Tr}(\widehat{K}_T(\widehat{K}_T + \lambda I)^{-1})\left(1 + \log\left(\tfrac{\|\widehat{K}_T\|}{\lambda} + 1\right)\right). \tag{7.4}$$

[2]Notice that $m_t \leq \min\{t, a\}$ and thus the complexity term $\mathcal{O}(m_t^3)$ is absorbed by the other terms.

We use both RLS and $\widehat{\mathcal{N}}$ to describe BKB's selection.

We now give the main result of this section.

Theorem 20. *For a desired $0 < \varepsilon < 1$, $0 < \delta < 1$, let $\alpha = (1+\varepsilon)/(1-\varepsilon)$. If we run* BKB *with $\overline{q} \geq 6\alpha \log(4T/\delta)/\varepsilon^2$, then with probability $1-\delta$, for all $t \in [T]$ and for all $x \in \mathcal{A}$, we have*

$$\sigma_t^2(x)/\alpha \leq \widetilde{\sigma}_t^2(x) \leq \alpha \sigma_t^2(x)$$

and

$$|\mathcal{I}_t| \leq 3(1+\kappa^2/\lambda)\alpha\overline{q}\widehat{\mathcal{N}}(\lambda, \widetilde{X}_t).$$

Computational complexity. We already showed that BKB's implementation with Nyström embedding requires $\mathcal{O}(T(A + \min\{t, A\})m_T^3)$ time and $\mathcal{O}(Am_T)$ space. Combining this with Theorem 20 and the bound $m_T \leq \widetilde{\mathcal{O}}(\widehat{\mathcal{N}})$, we obtain a $\widetilde{\mathcal{O}}(TA\widehat{\mathcal{N}}^2 + \min\{t, A\})\widehat{\mathcal{N}}^3)$ time complexity. Whenever $\widehat{\mathcal{N}} \ll T$ and $T \ll A$, this is essentially a quadratic $\mathcal{O}(T^2)$ runtime, a large improvement over the quartic $\mathcal{O}(T^4) \leq \mathcal{O}(T^3 A)$ runtime of GP-UCB.

Tuning $\overline{q}$. Note that although $\overline{q}$ must satisfy the condition of Theorem 20 for the result to hold, it is quite robust to uncertainty on the desired horizon T. In particular, the bound holds for *any* $\varepsilon > 0$, and even if we continue updating $\mathcal{I}_T$ after the T-th step, the bound still holds by implicitly increasing the parameter ε. Alternatively, after the T-th iteration the user can suspend the algorithm, increase $\overline{q}$ to suit the new desired horizon, and rerun only the subset selection on the arms selected so far.

Avoiding variance starvation. Another important consequence of Theorem 20 is that BKB's variance estimate is always close to the exact one up to a small constant factor. To the best of our knowledge, it makes BKB the first efficient and general GP algorithm that provably avoids variance starvation, which can be caused by two sources of error. The first source is the degeneracy, i.e. low-rankness of the GP approximation which causes the estimate to grow over-confident when the number of observed points grows and exceeds the degrees of freedom of the GP. BKB *adaptively chooses its degrees of freedom* as the size of $\mathcal{I}_t$ scales with the effective dimension. The second source of error arises when a point is far away from $\mathcal{I}_t$. Our use of a DTC variance estimator avoids under-estimation before we update the subset $\mathcal{I}_t$. Afterward, we can use guarantees on the quality of $\mathcal{I}_t$ to guarantee that we do not over-estimate the variance too much, exploiting a similar approach used to guarantee accuracy in RLS estimation. Both problems, and BKB's accuracy, are highlighted in Figure 7.1 using a benchmark experiment proposed by [WGKJ18].

Incremental dictionary update. At each step t, BKB recomputes the dictionary $\mathcal{I}_{t+1}$ from scratch by sampling each of the arms pulled so far with a suitable probability $\widetilde{p}_{t+1,i}$. A more efficient variant would be to build $\mathcal{I}_{t+1}$ by adding the new point $\mathbf{x}_{t+1}$ with probability $\widetilde{p}_{t+1,t+1}$ and including the points in $\mathcal{I}_t$ with probability $\widetilde{p}_{t+1,i}/\widetilde{p}_{t,i}$. This strategy is used in the streaming

setting to avoid storing all points observed so far and incrementally update the dictionary (see [CLV17b]). Nonetheless, the stream of points, although arbitrary, is assumed to be generated *independently* from the dictionary itself. On the other hand, in our bandit setting, the points $\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \dots$ are actually chosen by the learner depending on the dictionaries built over time, thus building a strong dependency between the stream of points and the dictionary itself. How to analyze such dependency and whether the accuracy of the inducing points is preserved in this case remains as an open question. Finally, notice that despite being more elegant and efficient, such incremental dictionary update would not significantly reduce the asymptotic computational complexity, since maximiming u_t, whose main cost is computing the posterior variance for each arm, would still dominate the overall runtime.

7.1.3 Regret Analysis

We are now ready to present the second main contribution of this chapter, a bound on the regret achieved by BKB. To prove our result we additionally assume that the reward function f has a bounded norm, i.e., $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle < \infty$. We use an upper-bound $\|f\|_{\mathcal{H}} \leq F$ to properly tune $\widetilde{\beta}_t$ to the range of the rewards. If F is not known in advance, standard guess-and-double techniques apply.

Theorem 21. *Assume $\|f\|_{\mathcal{H}} \leq F < \infty$. For any desired $0 < \varepsilon < 1$, $0 < \delta < 1$, $0 < \lambda$, let $\alpha = (1+\varepsilon)/(1-\varepsilon)$ and $\overline{q} \geq 6\alpha \log(4T/\delta)/\varepsilon^2$. If we run* BKB *with*

$$\widetilde{\beta}_t = 2\xi \sqrt{\alpha \log(\kappa^2 t) \left(\sum\nolimits_{s=1}^{t} \widetilde{\sigma}_t^2(\widetilde{x}_s) \right) + \log(1/\delta)} + \left(1 + \tfrac{1}{\sqrt{1-\varepsilon}}\right) \sqrt{\lambda} F,$$

then, with probability of at least $1-\delta$, the regret R_T of BKB *is bounded as*

$$R_T \leq 2(2\alpha)^{3/2} \sqrt{T} \left(\xi \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T) + \sqrt{\lambda F^2 \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T)} + \xi \log(1/\delta) \right).$$

Theorem 21 shows that BKB achieves exactly the same regret as (exact) GP-UCB up to small α constant and $\log(\kappa^2 T)$ multiplicative factor.[3] For instance, setting $\varepsilon = 1/2$ results in a bound only $3\log(T)$ times larger than the one of GP-UCB. At the same time, the choice $\varepsilon = 1/2$ only accounts for a constant factor 12 in the per-step computational complexity, which is still dramatically reduced from $t^2 A$ to $\widehat{\mathcal{N}}^2 A$. Note also that even if we send ε to 0, in the worst case we will include all arms selected so far, i.e. $\mathcal{I}_t = \{\widetilde{X}_t\}$. Therefore, even in this case BKB's runtime does not grow unbounded, but BKB transforms back into exact GP-UCB. Moreover, we show that $\widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \leq \log\det(\widehat{K}_T/\lambda + I)$, as in Proposition 15 in Section 7.3.1, so any bound on $\log\det(\widehat{K}_T/\lambda + I)$ available for GP-UCB applies directly to BKB. This means that

[3] Here we derive a *frequentist* regret bound and thus we compare with the result of [CG17] rather than the original *Bayesian* analysis of [SKKS10].

up to an extra $\log T$ factor, we match GP-UCB's $\widetilde{\mathcal{O}}(\log(T)^{2d})$ rate for the Gaussian kernel, $\widetilde{\mathcal{O}}(T^{\frac{1}{2}\frac{2\nu+3d^2}{2\nu+d^2}})$ rate for the Matérn kernel, and $\widetilde{\mathcal{O}}(d\sqrt{T})$ for the linear kernel. While these bounds are not minimax optimal, they closely follow the lower bounds derived in [SBC17]. On the other hand, in the case of linear kernel (i.e., the linear bandits) we nearly match the lower bound of [DHK08].

Another interesting aspect of BKB is that computing the trade-off parameter $\widetilde{\beta}_t$ can be done efficiently. Previous methods bounded this quantity with a loose (deterministic) upper bound, e.g., $\mathcal{O}(\log(T)^d)$ for Gaussian kernels, to avoid the large cost of computing $\log\det(\widehat{K}_T/\lambda + I)$. In our $\widetilde{\beta}_t$, we bound the $\log\det$ by $\widehat{\mathcal{N}}$, which is then bounded by $\sum_{s=1}^{t} \widetilde{\sigma}_t^2(x_s)$, see Theorem 20, where all $\widetilde{\sigma}_t^2$s are already efficiently computed at each step. While this is up to $\log t$ larger than the exact $\log\det$, it is *data adaptive* and much smaller than the known worst case upper bounds.

It it crucial, that our regret guarantee is achieved without requiring an *increasing accuracy* in our approximation. One would expect that to obtain a sublinear regret the error induced by the approximation should decrease as $1/T$. Instead, in BKB, the constants ε and λ that govern the accuracy level are fixed and thus it is not possible to guarantee that $\widetilde{\mu}_t$ will ever get close to μ_t everywhere. Adaptivity is the key: we can afford the same approximation level at every step because accuracy is actually increased only on a specific part of the arm set. For example, if a suboptimal arm is selected too often due to bad approximation, it will be eventually included in $\mathcal{I}_t$. After the inclusion, the approximation accuracy in the region of the suboptimal arm increases, and it would not be selected anymore. As the set of inducing points is updated *fast enough*, the impact of inaccurate approximations is limited over time, thus preventing large regret to accumulate. Note that this is a significant divergence from existing results. In particular approximation bounds that are uniformly accurate for all $x_i \in \mathcal{A}$, such as those obtained with quadrature FF [MK18], rely on packing arguments. Due to the nature of packing, this usually causes the runtime or regret to scale exponentially with the input dimension d, and requires kernel k to have a specific structure, e.g., to be stationary. Our new analysis avoids both of these problems.

Finally, we point out that the adaptivity of BKB allows drawing an interesting connection between learning and computational complexity. In fact, both the regret and the computation of BKB scale with the log-determinant and effective dimension of $\widehat{K}_T$, which is related to the effective dimension of the sequence of arms selected over time. As a result, if the problem is difficult from a learning point of view (i.e., the regret is large because of large log-determinant), then BKB automatically adapts the set $\mathcal{I}_t$ by including many more inducing points to guarantee the level of accuracy needed to solve the problem. Conversely, if the problem is simple (i.e., small regret), then BKB can greatly reduce the size of $\mathcal{I}_t$ and achieve the derived level of accuracy.

7.1.4 Sketch of the Proof

We build on the GP-UCB analysis of [CG17]. Their analysis relies on a confidence interval formulation of GP-UCB that is more conveniently expressed using an explicit feature-based representation of the GP. For any GP with covariance k, there is a corresponding RKHS $\mathcal{H}$ with k as its kernel function. Furthermore, any kernel function k is associated to a non-linear feature map $\phi(\cdot) : \mathbb{R}^d \to \mathcal{H}$ such that $k(x, x') = \phi(x')^\intercal \phi(x')$. As a result, any reward function $f \in \mathcal{H}$ can be written as $f(x) = \phi(x)^\intercal w_\star$, where $w_\star \in \mathcal{H}$.

Confidence-interval view of GP-UCB. Let $\Phi(X_t) = [\phi(x_1), \dots, \phi(x_t)]^\intercal$ be the matrix X_t after the application of $\phi(\cdot)$ to each row. We can then define the regularized design matrix as $A_t = \Phi(X_t)^\intercal \Phi(X_t) + \lambda I$, and then compute the regularized least-squares estimate as

$$\widehat{w}_t = \operatorname*{argmin}_{w \in \mathcal{H}} \sum\nolimits_{i=1}^{t} (y_i - \phi(x_i)^\intercal w)^2 + \lambda \|w\|_2^2 = A_t^{-1} \Phi(X_t)^\intercal y_t.$$

We define the *confidence interval* C_t as the ellipsoid induced by A_t with center $\widehat{w}_t$ and radius β_t

$$C_t = \{w : \|w - \widehat{w}_t\|_{A_t} \leq \beta_t\}, \qquad \beta_t = \lambda^{1/2} F + R\sqrt{2(\log\det(A_t/\lambda) + \log(1/\delta)}, \tag{7.5}$$

where the radius β_t is such that $w_\star \in C_t$ w.h.p. [CG17]. Finally, using Lagrange multipliers we reformulate the GP-UCB scores as

$$u_t(x_i) = \max_{w \in C_t} \phi(x_i)^\intercal w = \phi(x_{\phi(x_i)^\intercal \widehat{w}_t})^{\mu_t(x_i)} + \beta_t \phi(x_{\sqrt{\phi(x_i)^\intercal A_t^{-1} \phi(x_i)}.})^{\sigma_t(x_i)} \tag{7.6}$$

Approximating the confidence ellipsoid. Consider subset of arm $\mathcal{I}_t = \{x_i\}_{i=1}^m$ chosen by BKB at each step and denote by $X_{\mathcal{I}_t} \in \mathbb{R}^{m \times d}$ the matrix with all arms in $\mathcal{I}_t$ as rows. Let $\widetilde{\mathcal{H}}_t = \mathrm{Im}(\Phi(X_{\mathcal{I}_t}))$ be the smaller m-rank RKHS spanned by $\Phi(X_{\mathcal{I}_t})$; and by P_t the symmetric orthogonal projection operator on $\widetilde{\mathcal{H}}_t$. We then define an *approximate* feature map $\widetilde{\phi}_t(\cdot) = P_t \phi(\cdot) : \mathbb{R}^d \to \widetilde{\mathcal{H}}_t$ and associated approximations of A_t and $\widehat{w}_t$ as

$$\widetilde{A}_t = \widetilde{\Phi}_t(X_t)^\intercal \widetilde{\Phi}_t(X_t) + \lambda I, \tag{7.7}$$

$$\widetilde{w}_t = \operatorname*{argmin}_{w \in \mathcal{H}} \sum_{i=1}^{t} (y_i - \widetilde{\phi}(x_i)^\intercal w)^2 + \lambda \|w\|_2^2 = \widetilde{A}_t^{-1} \widetilde{\Phi}_t(X_t)^\intercal y_t. \tag{7.8}$$

This leads to an approximate confidence ellipsoid $\widetilde{C}_t = \left\{w : \|w - \widetilde{w}_t\|_{\widetilde{A}_t} \leq \widetilde{\beta}_t\right\}$, where we denote with $\|\cdot\|_A = \left\|A^{1/2}(\cdot)\right\|$. A subtle element in these definitions is that while $\widetilde{\Phi}_t(X_t)^\intercal \widetilde{\Phi}_t(X_t)$ and $\widetilde{w}_t$ are now *restricted* to $\widetilde{\mathcal{H}}_t$, the identity operator λI in the regularization of $\widetilde{A}_t$ still *acts over the whole* $\mathcal{H}$, and therefore $\widetilde{A}_t$ does not belong to $\widetilde{\mathcal{H}}_t$ and remains full-rank and invertible. This immediately leads to the usage of $k(x_i, x_i)$ in the definition of $\widetilde{\sigma}$ in Eq. (7.1), instead of the its approximate version using the Nyström embedding.

Bounding the regret. To find an appropriate $\widetilde{\beta}_t$ we follow an approach similar to the one of [AYPS11]. Exploiting the relationship $y_t = \widetilde{\phi}(\widetilde{x}_t)^\intercal w_\star + \eta_t$, we bound

$$\|w_\star - \widetilde{w}_t\|^2_{\widetilde{A}_t} \leq \phi(x_{\lambda^{1/2}\|w_\star\|})^{(a)} + \phi(x_{\|\widetilde{\Phi}_t(X_t)\boldsymbol{\eta}_t\|_{\widetilde{A}_t^{-1}}})^{(b)} + \phi(x_{\|\Phi(X_t)^\intercal\|_{I-P_t}\cdot\|w_\star\|})^{(c)}.$$

Both (a) and (b) are present in GP-UCB and OFUL's analysis. The first term (a) is due to the bias introduced in the least-square estimator $\widetilde{w}_t$ by the regularization λ. Then, term (b) is due to the noise in the reward observations. Note that the same term (b) appears in GP-UCB's analysis as $\|\Phi(X_t)\boldsymbol{\eta}_t\|_{A_t^{-1}}$ and it is bounded by $\log\det(A_t/\lambda)$ using self-normalizing concentration inequalities [CG17]. However, our $\|\widetilde{\Phi}_t(X_t)\eta_t\|_{\widetilde{A}_t^{-1}}$ is a more complex object, since the projection P_t contained in $\widetilde{\Phi}_t(X_t) = P_t\Phi(X_t)$ depends on the whole process up to time time t, and therefore $\widetilde{\Phi}_t(X_t)$ also depends on the whole process, losing its martingale structure. To avoid this, we use Sylvester's identity and the projection operator P_t to bound

$$\log\det(\widetilde{A}_t/\lambda) = \log\det\left(\tfrac{\Phi(X_t)P_t\Phi(X_t)^\intercal}{\lambda} + I\right) \leq \log\det\left(\tfrac{\Phi(X_t)\Phi(X_t)^\intercal}{\lambda} + I\right) = \log\det(A_t/\lambda).$$

In other words, restricting the problem to $\widetilde{\mathcal{H}}_t$ acts as a regularization and reduces the variance of the martingale. Unfortunately, $\log\det(A_t/\lambda)$ is too expensive to compute, so we first bound it with $\widehat{\mathcal{N}}(\lambda, \widetilde{X}_t)\log(\kappa^2 t)$, and then we bound $\widehat{\mathcal{N}}(\lambda, \widetilde{X}_t) \leq \alpha\sum_{s=1}^t \widetilde{\sigma}_t^2(x_s)$, Theorem 20, which can be computed efficiently. Finally, a new bias term (c) appears. Combining Theorem 20 with the results of [CR18] for projection P_t obtained using RLSs sampling, we show that

$$I - P \preceq \lambda A_t^{-1}/(1-\varepsilon).$$

The combination of (a), (b), and (c) leads to the definition of $\widetilde{\beta}_t$ and the final regret bound as $R_T \leq \sqrt{\widetilde{\beta}_T}\sqrt{\sum_{t=1}^T \phi(x_t)^\intercal \widetilde{A}_t^{-1}\phi(x_t)}$. To conclude the proof, we bound $\sum_{t=1}^T \phi(x_t)^\intercal \widetilde{A}_t^{-1}\phi(x_t)$ with the following corollary of Theorem 20.

Corollary 7. *Under the same conditions as Theorem 21, for all $t \in T$, we have $A_t/\alpha \preceq \widetilde{A}_t \preceq \alpha A_t$.*

Remarks. The novel bound $\|\Phi(X_t)^\intercal\|_{I-P_t} \leq \frac{\lambda}{1-\varepsilon}\|\Phi(X_t)^\intercal\|_{A_t^{-1}}$ has a crucial role in controlling the bias due to the projection P_t. Note that the second term measures the error with the same metric A_t^{-1} used by the variance martingale. In other words, the bias introduced by BKB's approximation can be seen as a *self-normalizing* bias. It is larger along directions that have been sampled less frequently, and smaller along directions correlated with arms selected often (e.g., the optimal arm).

Our analysis bears some similarity with the one recently and independently developed by [KCCB19]. Nonetheless, our proof improves their result along two dimensions. First, we consider the more

general (and challenging) GP optimization setting. Second, *we do not fix* the rank of our approximation in advance. While their analysis also exploits a self-normalized bias argument, this applies only to the k largest components. If the problem has an effective dimension larger than k, their radius and regret becomes essentially linear. In BKB we use our adaptive sampling scheme to include all necessary directions and to achieve the same regret rate as exact GP-UCB.

7.2 Discussion

As the prior work in Bayesian optimization is vast, we do not compare to alternative GP acquisition functions, such as GP-EI or GP-PI, and only focus on approximation techniques with theoretical guarantees. Similarly, we exclude scalable variational inference based methods, even when their approximate posterior is provably accurate such as pF-DTC [HCKB19], since they only provide guarantees for GP regression and not for the more difficult optimization setting. We also do not discuss SUPKERNELUCB [VKM+13], which has a tighter analysis than GP-UCB, since the algorithm does not work well in practice.

Infinite arm sets. Looking at the proof of Theorem 20, the guarantees on $\widetilde{u}_t$ hold for any $\mathcal{H}$, and in Theorem 21, we only require that the maximum $\widetilde{x}_{t+1} = \operatorname{argmax}_{x\in\mathcal{A}} \max_{w\in\widetilde{C}_t} \phi(x)^\intercal w$ is returned. Therefore, the accuracy and regret guarantees also hold also for an infinite set of arms $\mathcal{A}$. However, the search over $\mathcal{A}$ can be difficult. In the general case, maximization of a GP posterior is an NP-hard problem, with algorithms that often scale exponentially with the input dimension d and are not practical. We treated the easier case of finite sets, where enumeration is sufficient. Note that this automatically introduces an $\Omega(A)$ runtime dependency, which could be removed if the user provides an efficient method to solve the maximization problem on a specific infinite set $\mathcal{A}$. As an example, [MK18] prove that a GP posterior approximated using QFF can be optimized efficiently in low dimensions and we expect similar results hold for BKB and low *effective* dimension. Finally, note that recomputing a new set $\mathcal{I}_t$ still requires $\min\{A, t\}\widehat{\mathcal{N}}^2$ at each step. As discussed at the end of Section 7.1, this is a bottleneck in BKB due to the non-incremental dictionary sampling and independent from the arm selection. How to address it remains an open question.

Linear bandit with matrix sketching. Our analysis is related to the ones of CBRAP [YLK17] and SOFUL [KCCB19]. CBRAP uses Gaussian projections to embed all arms in a lower dimensional space for efficiency. Unfortunately their approach must either use an embedded space at least $\Omega(T)$ large, which in most cases would be even slower than exact OFUL, or it incurs linear regret w.h.p. Another approach for Euclidean spaces based on matrix approximation is SOFUL, introduced by [KCCB19]. It uses Frequent Direction [GLPW16], a method similar to incremental PCA, to embed the arms into $\mathbb{R}^m$, where m is *fixed* in advance. To compare, we distinguish between SOFUL-UCB and SOFUL-TS, a variant based on Thompson sampling. SOFUL-UCB achieves a $\widetilde{\mathcal{O}}(TAm^2)$ runtime and $\widetilde{\mathcal{O}}((1+\varepsilon_m)^{3/2}(d+m)\sqrt{T})$ regret, where ε_m

is the sum of the $d - m$ smallest eigenvalues of A_T. However, notice that if the tail do not decrease quickly, this algorithm also suffers linear regret and no adaptive way to tune m is known. On the same task BKB achieves a $\widetilde{\mathcal{O}}(d\sqrt{T})$ regret, since it adaptively chooses the size of the embedding. Computationally, directly instantiating BKB to use a linear kernel would achieve a $\widetilde{\mathcal{O}}(TAm_t^2)$ runtime[4], matching [KCCB19]'s. Compared to SOFUL-TS, BKB achieves better regret, but is potentially slower. Since Thompson sampling does not need to compute all confidence intervals, but solves a simpler optimization problem, SOFUL-TS requires only $\widetilde{\mathcal{O}}(TAm)$ time against BKB's $\widetilde{\mathcal{O}}(TAm_t^2)$. It is unknown if a variant of BKB can match this complexity.

Approximate GP with RFF. Traditionally, RFF approaches have been popular to transform GP optimization in a finite-dimensional problem and allow for scalability. Unfortunately GP-UCB with traditional RFF is not low-regret, as RFF are well known to suffer from variance starvation [WGKJ18] and unfeasibly large RFF embeddings would be necessary to prevent it. Recently, [MK18] proposed an alternative approach based on QFF, a specialized approach to random features for stationary kernels. They achieve the same regret rate as GP-UCB and BKB, with a near-optimal $\mathcal{O}(TA\log(T)^{d+1})$ runtime. Moreover they present an additional variations based on Thompson sampling whose posterior can be exactly maximized in polynomial time if the input data is low dimensional or the covariance k additive, while it is still an open question how to efficiently maximize BKB's UCB $\widetilde{u}_t$ for infinite $\mathcal{A}$. However QFF based approaches apply to stationary kernel only, and require to ε-cover $\mathcal{A}$, hence they cannot escape an exponential dependency on the dimensionality d. Conversely BKB can be applied to any kernel function, and while not specifically designed for this task it also achieve a close $\widetilde{\mathcal{O}}(TA\log(T)^{3(d+1)})$ runtime. Moreover, in practice the size of $\mathcal{I}_T$ is less than exponential in d.

7.2.1 Relaxing Assumptions

In our derivations, we make several assumptions. While some are necessary, others can be relaxed.

Assumptions on the noise. Throughout the chapter, we assume that the noise η_t is i.i.d. Gaussian. Since [CG17]'s results hold for any ξ-sub-Gussian noise that is measurable based with respect to the prior observations, this assumption can be easily relaxed.

Assumptions on the arms. So far we considered a set of arms that is (a) in $\mathbb{R}^d$, (b) fixed for all t, and (c) finite. Relaxing (a) is easy, since we do not make any assumption beyond boundedness on the kernel function k and there are many bounded kernel function for non-Euclidean spaces, e.g., strings or graphs. Relaxing (b) is trivial, we just need to embed the changing arm sets as they are provided, and store and re-embed previously selected arms as necessary. The per-step time complexity will now depend on the size of the set of arms available at each step. Relaxing (c) is straightforward from a theoretical perspective, but has varying computational

[4]Note that for both algorithms the bottleneck is maximizing the UCB.

consequences. In particular, looking at the proof of Theorem 20, the guarantees on $\widetilde{u}_t$ hold for all $\mathcal{H}$ and in Theorem 21, we only require that the maximum $\widetilde{x}_{t+1} = \operatorname{argmax}_{x\in\mathcal{A}} \max_{w\in\widetilde{C}_t} \phi(x)^\intercal w$ is returned. Therefore, at least from the regret point of view, everything holds also for infinite $\mathcal{A}$. However, while the inner maximization over $\widetilde{C}_t$ can be solved in closed form for a fixed x, the same cannot be said of the search over $\mathcal{A}$. If the designer can provide an efficient method to solve the maximization problem on an infinite $\mathcal{A}$, e.g., linear bandit optimization over compact subsets or $\mathbb{R}^d$, then all BKB guarantees apply.

7.3 Details of the Proofs

We present in this section the proofs of the main results of this chapter.

7.3.1 Properties of the Posterior Variance

For simplicity and completeness we provide known statements regarding the posterior variance $\sigma_t^2(\cdot)$. While most of these hold for generic RLS, we will adapt them to our notation.

Proposition 14 ([CLV17b])**.** *For the posterior variance, we have that*

$$\frac{1}{\kappa^2/\lambda+1}\sigma_{t-1}^2(\widetilde{x}_t) \leq \frac{1}{\sigma_{t-1}^2(\widetilde{x}_t)+1}\sigma_{t-1}^2(\widetilde{x}_t) \leq \sigma_t^2(\widetilde{x}_t) \leq \sigma_{t-1}^2(\widetilde{x}_t).$$

Proof. The leftmost inequality follows from $\kappa^2/\lambda \geq \sigma_0^2(x)$ and $\sigma_a^2(x) \geq \sigma_b^2(x), \forall a \leq b$, the others are by [CLV17b]. □

Proposition 15 ([HKAK06, CLV17d])**.** *The effective dimension $\widehat{\mathcal{N}}(\lambda, \widetilde{X}_T)$ is upper bounded as*

$$\begin{aligned}
\widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) = \operatorname{Tr}(\widehat{K}_T(\widehat{K}_T+\lambda I)^{-1}) &= \sum\nolimits_{t=1}^{T} \sigma_T^2(\widetilde{x}_t) \\
&\overset{(1)}{\leq} \sum\nolimits_{t=1}^{T} \sigma_t^2(\widetilde{x}_t) \\
&\overset{(2)}{\leq} \log\det\left(\widehat{K}_T/\lambda + I\right) \\
&\overset{(3)}{\leq} \operatorname{Tr}(\widehat{K}_T(\widehat{K}_T+\lambda I)^{-1})\left(1+\log\left(\tfrac{\|\widehat{K}_T\|}{\lambda}+1\right)\right).
\end{aligned}$$

Proof. Inequality (1) is due to Proposition 14, inequality (2) is due to [HKAK06], and inequality (3) is due to [CLV17d]. □

7.3.2 Proof of Theorem 20

Let B_t be the unfavorable event where the guarantees of Theorem 20 do not hold. Our goal is to prove that B_t happens at most with probability δ uniformly for all $t \in [T]$.

7.3.2.1 Notation

In the following we refer to $\Phi(\widetilde{X}_t)$ as Φ_t, $\widetilde{\Phi}(\widetilde{X}_t)$ as $\widetilde{\Phi}_t$ and $\phi(\widetilde{x}_t)$ as ϕ_t. When the subscript is clear from the context, we omit it. Since we leverage several results of [CLV17d], we start with some additional notation.

First we extend our notation for the subset $\mathcal{I}_t$ to include a possible reweighing of the inducing points. We denote with $\mathcal{I}_t = \{(\phi_j, s_j)\}_{j=1}^{m_t}$, a *weighted* subset, i.e., a weighted *dictionary*, of columns from Φ_t, with positive weights $s_j > 0$ that must be appropriately chosen. Now, denote with $i_j \in [t]$, the index of the sample ϕ_j as a column in Φ_t. Using a standard approach [AM15b], we choose $s_j = 1/\sqrt{\widetilde{p}_{t,i_j}}$, where $\widetilde{p}_{t,i} = \overline{q}\widetilde{\sigma}_{t-1}^2(\widetilde{x}_i)$ is the probability[5] used by Algorithm 5 when sampling ϕ_{i_j} from Φ_t.

Let $S_t \in \mathbb{R}^{t \times t}$ be the diagonal matrix with $q_{t,i}/\sqrt{\widetilde{p}_{t,i}}$ on the diagonal, where $q_{t,i}$ are the $\{0,1\}$-valued random variables selected by Algorithm 5. Then, we can see that

$$\sum_{j=1}^{m_t} \frac{1}{\widetilde{p}_{t,i_j}} \phi_{i_j} \phi_{i_j}^\mathsf{T} = \sum_{i=1}^{t} \frac{q_{t,i}}{\widetilde{p}_{t,i}} \phi_i \phi_i^\mathsf{T} = \Phi_t S_t S_t^\mathsf{T} \Phi_t^\mathsf{T}. \tag{7.9}$$

[CLV17b] define $\mathcal{I}_t$ to be an ε-accurate dictionary of Φ_t if it satisfies

$$(1-\varepsilon)\Phi_t\Phi_t^\mathsf{T} - \varepsilon\lambda I \preceq \Phi_t S_t S_t^\mathsf{T} \Phi_t^\mathsf{T} \preceq (1+\varepsilon)\Phi_t\Phi_t^\mathsf{T} + \varepsilon\lambda I. \tag{7.10}$$

We can also now fully define the projection operator at time t (see Section 7.1.4 for more details) as

$$P_t = \Phi_t S_t (S_t^\mathsf{T} \Phi_t^\mathsf{T} \Phi_t S_t)^+ S_t^\mathsf{T} \Phi_t^\mathsf{T},$$

which is the projection matrix spanned by the dictionary.

7.3.2.2 Event Decomposition

We decompose Theorem 20 into an accuracy part, i.e., $\mathcal{I}_t$ must induce accurate $\widetilde{\sigma}_t$, and an efficiency part, i.e., $m_t \leq \widehat{\mathcal{N}}(t)$. We also the accuracy of $\widetilde{\sigma}_t$ to the definition of ε-accuracy.

[5] Note that $\widetilde{p}_{t,i}$ might be larger than 1, but with a small abuse of notation and without the loss of generality we still refer to it as a probability.

Lemma 31. *Let $\alpha = \frac{1+\varepsilon}{1-\varepsilon}$. If $\mathcal{I}_t$ is ε-accurate w.r.t. Φ_t, then*

$$A_t/\alpha \preceq \widetilde{A}_t \preceq \alpha A_t \quad \textit{and} \quad \sigma_t^2(x)/\alpha \leq \min\left\{\widetilde{\sigma}_t^2(x), 1\right\} \leq \alpha\sigma_t^2(x) \ \textit{for all } x \in \mathcal{A}.$$

Proof. Inverting the bound in Equation (7.10) and using the fact that $P_t\Phi_t S_t = \Phi_t S_t$, we get

$$\begin{aligned} P_t\Phi_t\Phi_t^\intercal P_t &\preceq \frac{1}{1-\varepsilon}(P_t\Phi_t S_t S_t^\intercal \Phi_t^\intercal P_t + \varepsilon\lambda P_t) \preceq \frac{1}{1-\varepsilon}(\Phi_t S_t S_t^\intercal \Phi_t^\intercal + \varepsilon\lambda P_t) \\ &\preceq \frac{1}{1-\varepsilon}((1+\varepsilon)\Phi_t\Phi_t^\intercal + \varepsilon\lambda I + \varepsilon\lambda P_t) \preceq \frac{1+\varepsilon}{1-\varepsilon}\left(\Phi_t\Phi_t^\intercal + \frac{2\varepsilon}{1+\varepsilon}\lambda I\right). \end{aligned}$$

Repeating the same process for the other side, we obtain

$$\frac{1-\varepsilon}{1+\varepsilon}\left(\Phi_t\Phi_t^\intercal - \frac{2\varepsilon}{1-\varepsilon}\lambda I\right) \preceq P_t\Phi_t\Phi_t^\intercal P_t \preceq \frac{1+\varepsilon}{1-\varepsilon}\left(\Phi_t\Phi_t^\intercal + \frac{2\varepsilon}{1+\varepsilon}\lambda I\right).$$

Applying the above to $\widetilde{A}_t$, we get

$$\widetilde{A}_t = P_t\Phi_t\Phi_t^\intercal P_t + \lambda I \succeq \frac{1-\varepsilon}{1+\varepsilon}\left(\Phi_t\Phi_t^\intercal - \frac{2\varepsilon}{1-\varepsilon}\lambda I\right) + \lambda I = \frac{1-\varepsilon}{1+\varepsilon}\left(\Phi_t\Phi_t^\intercal + \lambda I\right) = \frac{1-\varepsilon}{1+\varepsilon}A_t,$$

which can again be applied on the other side to obtain our result. To prove the accuracy of the approximate posterior variance $\widetilde{\sigma}_t^2(x_i)$ we simply apply the definition to get

$$\frac{1-\varepsilon}{1+\varepsilon}\phi(x_{\phi_i^\intercal A_t \phi_i})^{\sigma_t^2(x_i)} \preceq \phi(x_{\phi_i^\intercal \widetilde{A}_t \phi_i})^{\widetilde{\sigma}_t^2(x_i)} \preceq \frac{1+\varepsilon}{1-\varepsilon}\phi(x_{\phi_i^\intercal A_t \phi_i})^{\sigma_t^2(x_i)}.$$

□

Using Lemma 31, we decompose our unfavorable event $B_t = A_t \cup E_t$, where A_t is the event where $\mathcal{I}_t$ is not ε-accurate w.r.t. Φ_t and E_t is the event where m_t is much larger than $\widehat{\mathcal{N}}(\lambda, \widetilde{X}_t)$. We now further decompose the event A_t as

$$\begin{aligned} A_t &= (A_t \cap A_{t-1}) \cup (A_t \cap A_{t-1}^\complement) \\ &\subseteq A_{t-1} \cup (A_t \cap A_{t-1}^\complement) = A_0 \cup \left(\bigcup_{s=1}^{t}(A_s \cap A_{s-1}^\complement)\right) = \bigcup_{s=1}^{t}(A_s \cap A_{s-1}^\complement), \end{aligned}$$

where A_0 is the empty event since Φ_0 is empty and it is well approximated by the empty $\mathcal{I}_0$. Moreover, we simplify a part of the expression by noting

$$B_t = A_t \cup E_t = A_t \cup (E_t \cap A_{t-1}^\complement) \cup (E_t \cap A_{t-1}) \subseteq A_t \cup A_{t-1} \cup (E_t \cap A_{t-1}^\complement),$$

which will help us when bounding the event E_t, where we will directly act as if A_t does not hold. Putting it all together, we get

$$\begin{aligned}
\bigcup_{t=1}^{T} B_t &= \bigcup_{t=1}^{T}(A_t \cup E_t) \subseteq \bigcup_{t=1}^{T}\Big(A_t \cup A_{t-1} \cup (E_t \cap A_{t-1}^{\complement})\Big) \\
&= \left(\bigcup_{t=1}^{T} A_t\right) \cup \left(\bigcup_{t=1}^{T}(E_t \cap A_{t-1}^{\complement})\right) = \left(\bigcup_{t=1}^{T} A_t\right) \cup \left(\bigcup_{t=1}^{T}(E_t \cap A_{t-1}^{\complement})\right) \\
&\subseteq \left(\bigcup_{t=1}^{T}\left(\bigcup_{s=1}^{t}(A_s \cap A_{s-1}^{\complement})\right)\right) \cup \left(\bigcup_{t=1}^{T}(E_t \cap A_{t-1}^{\complement})\right) \\
&= \left(\bigcup_{t=1}^{T}(A_t \cap A_{t-1}^{\complement})\right) \cup \left(\bigcup_{t=1}^{T}(E_t \cap A_{t-1}^{\complement})\right).
\end{aligned}$$

7.3.2.3 Bounding $\Pr(A_t \cap A_{t-1}^{\complement})$

We now bound the probability of event $A_t \cap A_{t-1}^{\complement}$. In our first step, we formally define A_t using Equation (7.10). In particular, we rewrite the ε-accuracy condition as

$$\begin{aligned}
&(1-\varepsilon)\Phi_t\Phi_t^{\intercal} - \varepsilon\lambda I \preceq \Phi_t S_t S_t^{\intercal}\Phi_t^{\intercal} \preceq (1+\varepsilon)\Phi_t\Phi_t^{\intercal} + \varepsilon\lambda I \\
&\iff -\varepsilon(\Phi_t\Phi_t^{\intercal} + \lambda I) \preceq \Phi_t S_t S_t^{\intercal}\Phi_t^{\intercal} - \Phi_t\Phi_t^{\intercal} \preceq \varepsilon(\Phi_t\Phi_t^{\intercal} + \lambda I) \\
&\iff -\varepsilon I \preceq (\Phi_t\Phi_t^{\intercal} + \lambda I)^{-1/2}(\Phi_t S_t S_t^{\intercal}\Phi_t^{\intercal} - \Phi_t\Phi_t^{\intercal})(\Phi_t\Phi_t^{\intercal} + \lambda I)^{-1/2} \preceq \varepsilon I \\
&\iff \|(\Phi_t\Phi_t^{\intercal} + \lambda I)^{-1/2}(\Phi_t S_t S_t^{\intercal}\Phi_t^{\intercal} - \Phi_t\Phi_t^{\intercal})(\Phi_t\Phi_t^{\intercal} + \lambda I)^{-1/2}\| \leq \varepsilon,
\end{aligned}$$

where $\|\cdot\|$ is the spectral norm. We now focus on the last reformulation and frame it as a random matrix concentration question in RKHS $\mathcal{H}$. Let $\psi_{t,i} = (\Phi_t\Phi_t^{\intercal} + \lambda I)^{-\frac{1}{2}}\phi_i$ and $P_t = \Phi_t(\Phi_t^{\intercal}\Phi_t + \lambda I)^{-\frac{1}{2}} = [\psi_{t,1}, \ldots, \psi_{t,t}]^{\intercal}$, and define the operator $G_{t,i} = \left(\frac{q_{t,i}}{\widetilde{p}_{t,i}} - 1\right)\psi_{t,i}\psi_{t,i}^{\intercal}$. Then we rewrite ε-accuracy as

$$\left\|(\Phi_t\Phi_t^{\intercal}+\lambda I)^{-\frac{1}{2}}\Phi_t(S_tS_t^{\intercal}-I)\Phi_t^{\intercal}(\Phi_t\Phi_t^{\intercal}+\lambda I)^{-\frac{1}{2}}\right\| = \left\|\sum_{i=1}^{t}\left(\frac{q_{t,i}}{\widetilde{p}_{t,i}} - 1\right)\psi_{t,i}\psi_{t,i}^{\intercal}\right\| = \left\|\sum_{i=1}^{t} G_{t,i}\right\| \leq \varepsilon,$$

and the event A_t as the event where $\left\|\sum_{i=1}^{t} G_{t,i}\right\| \geq \varepsilon$, Note that this reformulation exploits the fact that $q_{t,i} = 0$ encodes the column that are not selected in $\mathcal{I}_t$ (see Equation (7.9)). To study this random object, we begin by defining the filtration $\mathcal{F}_t = \{q_{s,i}, \eta_s\}_{s=1}^{t}$ at time t containing all the randomness coming from the construction of the various $\mathcal{I}_s$ and the noise on the function η_t. In particular, note that the $\{0,1\}$-valued r.v. $q_{t,i}$ used by Algorithm 5 are not necessarily Bernoulli r.v.s, since the probability $\widetilde{p}_{t,i}$ used to select 0 or 1 is itself random. However, they become

well defined Bernoulli when conditioned on $\mathcal{F}_{t-1}$. Let $\mathbb{I}\{\cdot\}$ indicates the indicator function of an event. We have that

$$\begin{aligned}
\Pr(A_t \cap A_{t-1}^{\complement}) &= \Pr\left(\left\|\sum_{i=1}^{t} G_{t,i}\right\| \geq \varepsilon \cap \left\|\sum_{i=1}^{t} G_{t-1,i}\right\| \leq \varepsilon\right) \\
&= \mathop{\mathbb{E}}_{\mathcal{F}_t}\left[\mathbb{I}\left\{\left\|\sum_{i=1}^{t} G_{t,i}\right\| \geq \varepsilon \cap \left\|\sum_{i=1}^{t} G_{t-1,i}\right\| \leq \varepsilon\right\}\right] \\
&= \mathop{\mathbb{E}}_{\mathcal{F}_{t-1}}\left[\mathop{\mathbb{E}}_{\eta_t,\{q_{t,i}\}}\left[\mathbb{I}\left\{\left\|\sum_{i=1}^{t} G_{t,i}\right\| \geq \varepsilon \cap \left\|\sum_{i=1}^{t} G_{t-1,i}\right\| \leq \varepsilon\right\} \;\middle|\; \mathcal{F}_{t-1}\right]\right] \\
&= \mathop{\mathbb{E}}_{\mathcal{F}_{t-1}}\left[\mathop{\mathbb{E}}_{\{q_{t,i}\}}\left[\mathbb{I}\left\{\left\|\sum_{i=1}^{t} G_{t,i}\right\| \geq \varepsilon \cap \left\|\sum_{i=1}^{t} G_{t-1,i}\right\| \leq \varepsilon\right\} \;\middle|\; \mathcal{F}_{t-1}\right]\right],
\end{aligned}$$

where the last passage is due to the fact that $G_{t,i}$ is independent from η_t. Next, notice that conditioned on $\mathcal{F}_{t-1}$, the event $A_{t-1}^{\complement}$ becomes deterministic, and we can restrict our expectations to the outcomes where $\left\|\sum_{i=1}^{t} G_{t-1,i}\right\| \leq \varepsilon$,

$$\Pr(A_t \cap A_{t-1}^{\complement}) = \mathop{\mathbb{E}}_{\mathcal{F}_{t-1}:\left\|\sum_{i=1}^{t} G_{t-1,i}\right\| \leq \varepsilon}\left[\mathop{\mathbb{E}}_{\{q_{t,i}\}}\left[\mathbb{I}\left\{\left\|\sum_{i=1}^{t} G_{t,i}\right\| \geq \varepsilon\right\} \;\middle|\; \mathcal{F}_{t-1}\right]\right].$$

Moreover, conditioned on $\mathcal{F}_{t-1}$ all the $q_{t,i}s$ become independent r.v., and we are able to use the following result of [Tro15].

Proposition 16. *Let $G_1, \ldots, G_n$ be a sequence of independent self-adjoint random operators such that $\mathbb{E}[G_i] = 0$ and $\|G_i\| \leq R$ a.s. Denote $\sigma^2 = \left\|\sum_{i=1}^{t} \mathbb{E}[G_i^2]\right\|$. Then, for any $\varepsilon \geq 0$,*

$$\Pr\left(\left\|\sum_{i=1}^{t} G_i\right\| \geq \varepsilon\right) \leq 4t \exp\left(\frac{\varepsilon^2/2}{\sigma^2 + R\varepsilon/3}\right).$$

We begin by computing the mean of $G_{t,i}$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{q_{t,i}}[G_{t,i} \mid \mathcal{F}_{t-1}] &= \mathop{\mathbb{E}}_{q_{t,i}}\left[\left(\frac{q_{t,i}}{\widetilde{p}_{t,i}} - 1\right)\psi_{t,i}\psi_{t,i}^{\intercal} \;\middle|\; \mathcal{F}_{t-1}\right] \\
&= \left(\frac{\mathbb{E}_{q_{t,i}}[q_{t,i} \mid \mathcal{F}_{t-1}]}{\widetilde{p}_{t,i}} - 1\right)\psi_{t,i}\psi_{t,i}^{\intercal} = \left(\frac{\widetilde{p}_{t,i}}{\widetilde{p}_{t,i}} - 1\right)\psi_{t,i}\psi_{t,i}^{\intercal} = \mathbf{0},
\end{aligned}$$

where we use the fact that $\widetilde{p}_{t,i}$ is fixed conditioned on $\mathcal{F}_{t-1}$ and it is the (conditional) expectation of $q_{t,i}$. Since G is zero-mean, we can use Proposition 16. First, we find R and for that, we upper bound

$$\|G_{t,i}\| = \left\|\left(\frac{q_{t,i}}{\widetilde{p}_{t,i}} - 1\right)\psi_{t,i}\psi_{t,i}^{\intercal}\right\| \leq \left|\left(\frac{q_{t,i}}{\widetilde{p}_{t,i}} - 1\right)\right| \|\psi_{t,i}\psi_{t,i}^{\intercal}\| \leq \frac{1}{\widetilde{p}_{t,i}}\|\psi_{t,i}\psi_{t,i}^{\intercal}\|.$$

Note that due to the definition of $\psi_{t,i}$,

$$\|\psi_{t,i}\psi_{t,i}^\intercal\| = \psi_{t,i}^\intercal\psi_{t,i} = \phi_i^\intercal(\Phi_t\Phi_t^\intercal + \lambda I)^{-1}\phi_i = \sigma_t^2(\widetilde{x}_i).$$

Moreover, we are only considering outcomes of $\mathcal{F}_{t-1}$ where $\left\|\sum_{i=1}^t G_{t-1,i}\right\| \le \varepsilon$, which implies that $\mathcal{I}_{t-1}$ is ε-accurate, and by Lemma 31 we have that $\widetilde{\sigma}_{t-1}(\widetilde{x}_i) \ge \sigma_{t-1}(\widetilde{x}_i)/\alpha$. Finally, due to Proposition 14, we have $\sigma_{t-1}(\widetilde{x}_i) \ge \sigma_t(\widetilde{x}_i)$. Putting this all together we can bound

$$\frac{1}{\widetilde{p}_{t,i}}\|\psi_{t,i}\psi_{t,i}^\intercal\| = \frac{1}{\overline{q}\widetilde{\sigma}_{t-1}(\widetilde{x}_i)}\sigma_t(\widetilde{x}_i) \le \frac{\alpha}{\overline{q}} = R.$$

For the variance term, we expand

$$\begin{aligned}
\sum_{i=1}^t \mathop{\mathbb{E}}_{q_{t,i}}\left[G_{t,i}^2 \mid \mathcal{F}_{t-1}\right] &= \sum_{i=1}^t \mathop{\mathbb{E}}_{q_{t,i}}\left[\left(\frac{q_{t,i}}{\widetilde{p}_{t,i}} - 1\right)^2 \,\middle|\, \mathcal{F}_{t-1}\right]\psi_{t,i}\psi_{t,i}^\intercal\psi_{t,i}\psi_{t,i}^\intercal \\
&= \sum_{i=1}^t \left(\mathop{\mathbb{E}}_{q_{t,i}}\left[\frac{q_{t,i}^2}{\widetilde{p}_{t,i}^2} \,\middle|\, \mathcal{F}_{t-1}\right] - \mathop{\mathbb{E}}_{q_{t,i}}\left[2\frac{q_{t,i}}{\widetilde{p}_{t,i}} \,\middle|\, \mathcal{F}_{t-1}\right] + 1\right)\psi_{t,i}\psi_{t,i}^\intercal\psi_{t,i}\psi_{t,i}^\intercal \\
&= \sum_{i=1}^t \left(\mathop{\mathbb{E}}_{q_{t,i}}\left[\frac{q_{t,i}}{\widetilde{p}_{t,i}^2} \,\middle|\, \mathcal{F}_{t-1}\right] - 1\right)\psi_{t,i}\psi_{t,i}^\intercal\psi_{t,i}\psi_{t,i}^\intercal = \sum_{i=1}^t \left(\mathop{\mathbb{E}}_{q_{t,i}}\left[\frac{q_{t,i}}{\widetilde{p}_{t,i}^2} \,\middle|\, \mathcal{F}_{t-1}\right] - 1\right)\psi_{t,i}\psi_{t,i}^\intercal\psi_{t,i}\psi_{t,i}^\intercal \\
&= \sum_{i=1}^t \left(\frac{1}{\widetilde{p}_{t,i}} - 1\right)\psi_{t,i}\psi_{t,i}^\intercal\psi_{t,i}\psi_{t,i}^\intercal \preceq \sum_{i=1}^t \frac{1}{\widetilde{p}_{t,i}}\|\psi_{t,i}\psi_{t,i}^\intercal\|\psi_{t,i}\psi_{t,i}^\intercal \preceq \sum_{i=1}^t R\psi_{t,i}\psi_{t,i}^\intercal,
\end{aligned}$$

where we used the fact that $q_{t,i}^2 = q_{t,i}$ and $\mathbb{E}_{q_{t,i}}[q_{t,i}|\mathcal{F}_{t-1}] = \widetilde{p}_{t,i}$. We can now bound this quantity as

$$\left\|\sum_{i=1}^t \mathop{\mathbb{E}}_{q_{t,i}}\left[G_{t,i}^2 \mid \mathcal{F}_{t-1}\right]\right\| \le \left\|\sum_{i=1}^t R\psi_{t,i}\psi_{t,i}^\intercal\right\| = R\left\|\sum_{i=1}^t \psi_{t,i}\psi_{t,i}^\intercal\right\| = R\|P_t^\intercal P_t\| \le R = \sigma^2.$$

Therefore, we have $\sigma^2 = R$ and $R = 1/\overline{q}$. Now, applying Proposition 16 and a union bound we conclude the proof.

7.3.2.4 Bounding $\Pr(E_t \cap A_{t-1}^{\complement})$

We will use the following concentration for independent Bernoulli random variables.

Proposition 17 ([CLV17b], App. D.4)**.** *Let $\{q_s\}_{s=1}^t$ be independent Bernoulli random variables, each with success probability p_s, and let $d = \sum_{s=1}^t p_s \ge 1$ be their sum. Then,*[6]

$$\mathbb{P}\left(\sum_{s=1}^t q_s \ge 3d\right) \le \exp\{-3d(3d - (\log(3d) + 1))\} \le \exp\{-2d\}.$$

[6]This is a simple variant of the Chernoff bound where the Bernoulli random variables are not identically distributed.

We now rigorously define event E_t as the event where

$$\sum_{i=1}^{t} q_{t,i} \geq 3\alpha(1+\kappa^2/\lambda)\log(t/\delta)\sum_{i=1}^{t}\sigma_t^2(\widetilde{x}_i) = 3\alpha(1+\kappa^2/\lambda)\widehat{\mathcal{N}}(\lambda,\widetilde{X}_t)\log(t/\delta).$$

Once again, we use conditioning and in particular,

$$\Pr(E_t \cap A_t^{\complement}) = \mathop{\mathbb{E}}_{\mathcal{F}_{t-1}:\left\|\sum_{i=1}^{t} G_{t-1,i}\right\|\leq\varepsilon}\left[\mathop{\mathbb{E}}_{\{q_{t,i}\}}\left[\mathbb{I}\left\{\sum_{i=1}^{t} q_{t,i} \geq 3\alpha(1+\kappa^2/\lambda)\log(t/\delta)\sum_{i=1}^{t}\sigma_t^2(\widetilde{x}_i)\right\}\middle|\mathcal{F}_{t-1}\right]\right].$$

Conditioned on $\mathcal{F}_{t-1}$ the r.v. $q_{t,i}$ becomes independent Bernoulli with probability $\widetilde{p}_{t,i} = \overline{q}\widetilde{\sigma}_{t-1}(\widetilde{x}_i)$. Since we restrict the outcomes to $A_{t-1}^{\complement}$, we can exploit Lemma 31 and the guarantees of ε-accuracy to bound $\widetilde{p}_{t,i} \leq \alpha\sigma_{t-1}^2(\widetilde{x}_i)$. Then, we use Proposition 14 to bound $\sigma_{t-1}^2(\widetilde{x}_i) \leq (1+\kappa^2/\lambda)\sigma_t^2(\widetilde{x}_i)$. Therefore, $q_{t,i}$ are conditionally independent Bernoulli with probability at most $\overline{q}(1+\kappa^2/\lambda)\sigma_t^2(\widetilde{x}_i)$. Applying a simple stochastic dominance argument and Proposition 17 gets the needed statement.

7.3.3 Proof of Theorem 21

Following [AYPS11], we divide the proof in two parts, first bounding the approximate confidence ellipsoid, and then bounding the regret.

7.3.3.1 Bounding the Confidence Ellipsoid

We begin by proving an intermediate result regarding the confidence ellipsoid.

Theorem 22. *Under the same assumptions as Theorem 21 with probability at least $1-\delta$ and for all $t \geq 0$, $w_\star$ lies in the set*

$$\widetilde{C}_t = \left\{w : \|w - \widetilde{w}_t\|_{\widetilde{A}_t} \leq \widetilde{\beta}_t\right\}$$

with

$$\widetilde{\beta}_t = 2\xi\sqrt{\alpha\log(\kappa^2 t)\left(\sum_{s=1}^{t}\widetilde{\sigma}_t^2(x_s)\right) + \log\left(\frac{1}{\delta}\right)} + \left(1+\frac{1}{\sqrt{1-\varepsilon}}\right)\sqrt{\lambda}F.$$

Proof. For simplicity, we omit the subscript t. We begin by noticing that

$$\begin{aligned}
(\widetilde{w}-w_\star)^{\intercal}\widetilde{A}(\widetilde{w}-w_\star) &= (\widetilde{w}-w_\star)^{\intercal}\widetilde{A}(\widetilde{A}^{-1}\widetilde{\Phi}^{\intercal}y - w_\star) \\
&= (\widetilde{w}-w_\star)^{\intercal}\widetilde{A}(\widetilde{A}^{-1}\widetilde{\Phi}^{\intercal}(\Phi w_\star + \eta - w_\star) \\
&= (\widetilde{w}-w_\star)^{\intercal}\widetilde{A}(\underbrace{\widetilde{A}^{-1}\widetilde{\Phi}^{\intercal}\Phi w_\star - w_\star}_{\text{bias}}) + (\widetilde{w}-w_\star)^{\intercal}\widetilde{A}^{1/2}\underbrace{\widetilde{A}^{-1/2}\widetilde{\Phi}^{\intercal}\eta}_{\text{variance}}.
\end{aligned}$$

Bounding the bias. We first focus on the first term, which is difficult to analyze due to the mismatch $\widetilde{\Phi}^\intercal\Phi$. We have that

$$\begin{aligned}\widetilde{A}(\widetilde{A}^{-1}\widetilde{\Phi}^\intercal\Phi w_\star - w_\star) &= \widetilde{\Phi}^\intercal\Phi w_\star - \widetilde{\Phi}^\intercal\widetilde{\Phi}w_\star - \lambda w_\star \\ &= \widetilde{\Phi}^\intercal\Phi(I-P)w_\star + \widetilde{\Phi}^\intercal\Phi P w_\star - \widetilde{\Phi}^\intercal\widetilde{\Phi}_\star^w - \lambda w_\star \\ &= \widetilde{\Phi}^\intercal\Phi(I-P)w_\star - \lambda w_\star.\end{aligned}$$

Therefore,

$$\begin{aligned}(\widetilde{w}-w_\star)^\intercal\widetilde{A}(\widetilde{A}^{-1}\widetilde{\Phi}^\intercal\Phi w_\star - w_\star) &= (\widetilde{w}-w_\star)^\intercal\widetilde{\Phi}^\intercal\Phi(I-P)w_\star - \lambda(\widetilde{w}-w_\star)^\intercal w_\star \\ &\le \|\widetilde{w}-w_\star\|_{\widetilde{A}}\left(\|\widetilde{A}^{-1/2}\widetilde{\Phi}^\intercal\Phi(I-P)w_\star\| + \lambda\|w^*\|_{\widetilde{A}^{-1}}\right) \\ &\le \|\widetilde{w}-w_\star\|_{\widetilde{A}}\left(\|\widetilde{A}^{-1/2}\widetilde{\Phi}^\intercal\Phi(I-P)w_\star\| + \tfrac{\lambda}{\sqrt{\lambda}}\|w^*\|\right).\end{aligned}$$

Then, we have that

$$\begin{aligned}\|\widetilde{A}^{-1/2}\widetilde{\Phi}^\intercal\Phi(I-P)w_\star\| &\le \|\widetilde{A}^{-1/2}\widetilde{\Phi}^\intercal\|\|\Phi(I-P)\|\|w_\star\| \\ &\le \sqrt{\lambda_{\max}(\widetilde{\Phi}\widetilde{A}^{-1}\widetilde{\Phi}^\intercal)}\sqrt{\lambda_{\max}(\Phi(I-P)^2\Phi^\intercal)}\|w_\star\|.\end{aligned}$$

It is easy to see that

$$\lambda_{\max}(\widetilde{\Phi}\widetilde{A}^{-1}\widetilde{\Phi}^\intercal) = \lambda_{\max}(\widetilde{\Phi}(\widetilde{\Phi}^\intercal\widetilde{\Phi}+\lambda I)^{-1}\widetilde{\Phi}^\intercal) \le 1.$$

To bound the other term we use the following result by [CR18].

Proposition 18. *If $\mathcal{I}_t$ is ε-accurate w.r.t. Φ_t, then*

$$I - P_t \preceq I - \Phi_t S_t(S_t^\intercal\Phi_t^\intercal\Phi_t S_t + \lambda I)^{-1}S_t^\intercal\Phi_t^\intercal \preceq \frac{\lambda}{1-\varepsilon}(\Phi_t\Phi_t^\intercal + \lambda I)^{-1}.$$

Since from Theorem 20, we have that $\mathcal{I}_t$ is ε-accurate, by Proposition 18, we have that

$$\Phi(I-P)^2\Phi^\intercal = \Phi(I-P)\Phi^\intercal \preceq \frac{\lambda}{1-\varepsilon}\Phi(\Phi^\intercal\Phi+\lambda I)^{-1}\Phi^\intercal \preceq \frac{\lambda}{1-\varepsilon}I.$$

Putting it all together, we obtain

$$(\widetilde{w}-w_\star)^\intercal\widetilde{A}(\widetilde{A}^{-1}\widetilde{\Phi}^\intercal\Phi w_\star - w_\star) \le \left(1+\frac{1}{\sqrt{1-\varepsilon}}\right)\|\widetilde{w}-w_\star\|_{\widetilde{A}}\sqrt{\lambda}\|w_\star\|.$$

Bounding the variance. We use the the following self-normalized martingale concentration inequality by [AYPS11]. It can be trivially extended to RKHSs in the case of finite sets such as our $\mathcal{A}$. Note that if the reader is interested in infinite sets, [CG17] provide a generalization with slightly worse constants.

Proposition 19 ([AYPS11]). *Let $\{\mathcal{F}_t\}_{t=0}^{\infty}$ be a filtration, let $\{\eta_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that η_t is $\mathcal{F}_t$-measurable and zero-mean ξ-subgaussian; let $\{\Phi_t\}_{t=1}^{\infty}$ be an $\mathcal{H}$-valued stochastic process such that Φ_t is $\mathcal{F}_{t-1}$-measurable, and let I be the identity operator on $\mathcal{H}$. For any $t \geq 1$, define*

$$A_t = \Phi_t^\intercal \Phi_t + \lambda I \quad \textit{and} \quad V_t = \Phi_t^\intercal \eta_t.$$

Then, for any $\delta > 0$, with probability at least $1-\delta$, for all $t \geq 0$,

$$\|V_t\|_{A_t^{-1}}^2 \leq 2\xi^2 \log\left(\frac{\det(A_t/\lambda)}{\delta}\right).$$

Recalling the definition of $\alpha \geq 1$ from Theorem 20, we reformulate

$$\begin{aligned}(\widetilde{w} - w_\star)^\intercal \widetilde{A}^{1/2} \widetilde{A}^{-1/2} \widetilde{\Phi}\eta &\leq \|\widetilde{w} - w_\star\|_{\widetilde{A}} \|\widetilde{\Phi}\eta\|_{\widetilde{A}^{-1}} \\ &= \|\widetilde{w} - w_\star\|_{\widetilde{A}} \|\widetilde{\Phi}^\intercal \eta\|_{(\widetilde{\Phi}^\intercal\widetilde{\Phi}+\lambda I)^{-1}} \\ &= \|\widetilde{w} - w_\star\|_{\widetilde{A}} \|\widetilde{\Phi}^\intercal \eta/\lambda\|_{(\widetilde{\Phi}^\intercal\widetilde{\Phi}/\lambda+I)^{-1}}.\end{aligned}$$

We now make a remark that requires temporal notation. Note that we cannot directly apply Proposition 19 to $\widetilde{\Phi}_t\eta_t = P_t\Phi_t\eta_t$. In particular, for $s < t$ we have that $\widetilde{\Phi}_s\eta_s = P_t\Phi_s\eta_s$ is not $\mathcal{F}_{s-1}$ measurable, since P_t depends on all randomness up to time t. However, since P_t is always a projection matrix we know that the variance of the projected process is bounded by the variance of the original process, in particular,

$$\begin{aligned}\|\widetilde{\Phi}^\intercal \eta/\lambda\|_{(\widetilde{\Phi}^\intercal\widetilde{\Phi}/\lambda+I)^{-1}} &= \sqrt{\eta^\intercal \widetilde{\Phi}(\widetilde{\Phi}^\intercal\widetilde{\Phi}/\lambda + I)^{-1}\widetilde{\Phi}^\intercal\eta/\lambda} = \sqrt{\eta^\intercal \widetilde{\Phi}\widetilde{\Phi}^\intercal(\widetilde{\Phi}\widetilde{\Phi}^\intercal/\lambda + I)^{-1}\eta/\lambda} \\ &\overset{(a)}{=} \sqrt{\eta^\intercal(I - \lambda(\widetilde{\Phi}\widetilde{\Phi}^\intercal/\lambda + I)^{-1})\eta/\lambda} = \sqrt{\eta^\intercal(I - \lambda(\Phi P \Phi^\intercal/\lambda + I)^{-1})\eta/\lambda} \\ &\overset{(b)}{\leq} \sqrt{\eta^\intercal(I - \lambda(\Phi\Phi^\intercal/\lambda + I)^{-1})\eta/\lambda} \overset{(c)}{=} \|\Phi^\intercal\eta/\lambda\|_{(\Phi^\intercal\Phi/\lambda+I)^{-1}},\end{aligned}$$

where in (a) we added and subtracted λI from $\widetilde{\Phi}\widetilde{\Phi}^\intercal$, in (b) we used the fact that $\|P\| \leq 1$ for all projection matrices, and in (c) we reversed the reformulation from (a). We can finally use Proposition 19 to obtain

$$\begin{aligned}\|\Phi^\intercal\eta/\lambda\|_{(\Phi^\intercal\Phi/\lambda+I)^{-1}} &\leq \sqrt{2\xi^2 \log(\mathrm{Det}(\Phi^\intercal\Phi/\lambda + I)/\delta)} \\ &= \sqrt{2\xi^2 \log(\mathrm{Det}(A/\lambda)/\delta)}.\end{aligned}$$

While above is a valid bound on the radius of the confidence interval, it is still not satisfactory. In particular, we can use Sylvester's identity to reformulate

$$\log\det(A/\lambda) = \log\det(\Phi^\intercal\Phi/\lambda + I) = \log\det(\Phi\Phi^\intercal/\lambda + I) = \log\det(\widehat{K}/\lambda + I).$$

Computing the radius would require constructing the matrix $\widehat{K} \in \mathbb{R}^{t \times t}$ and this is way too expensive. Instead, we obtain a cheap but still a small enough upper bound as follows,

$$\begin{aligned}
\log\det(\widehat{K}_t/\lambda + I) &\leq \mathrm{Tr}(\widehat{K}_t(\widehat{K}_t + \lambda I)^{-1})(1 + \log(\|\widehat{K}_t\| + 1)) \\
&\leq \mathrm{Tr}(\widehat{K}_t(\widehat{K}_t + \lambda I)^{-1})(1 + \log(\mathrm{Tr}\,\widehat{K}_t + 1)) \\
&\leq \mathrm{Tr}(\widehat{K}_t(\widehat{K}_t + \lambda I)^{-1})(1 + \log(\kappa^2 t + 1)) \\
&= (1 + \log(\kappa^2 t + 1)) \sum_{s=1}^{t} \sigma_t^2(x_s) \\
&\leq \alpha(1 + \log(\kappa^2 t + 1)) \sum_{s=1}^{t} \widetilde{\sigma}_t^2(x_s) \\
&\leq 2\alpha \log(\kappa^2 t) \sum_{s=1}^{t} \widetilde{\sigma}_t^2(x_s),
\end{aligned}$$

where $\widetilde{\sigma}_t^2(x_s)$ can be computed efficiently and it is actually already done by the algorithm at every step! Putting it all together, we get that

$$\begin{aligned}
\|\widetilde{w} - w_\star\|_{\widetilde{A}} &\leq 2\xi \sqrt{\alpha \log(\kappa^2 t) \left(\sum_{s=1}^{t} \widetilde{\sigma}_t^2(x_s) \right) + \log(1/\delta)} + \left(1 + \frac{1}{\sqrt{1-\varepsilon}}\right) \sqrt{\lambda} \|w_\star\| \\
&\leq 2\xi \sqrt{\alpha \log(\kappa^2 t) \left(\sum_{s=1}^{t} \widetilde{\sigma}_t^2(x_s) \right) + \log(1/\delta)} + \left(1 + \frac{1}{\sqrt{1-\varepsilon}}\right) \sqrt{\lambda} F = \widetilde{\beta}_t.
\end{aligned}$$

□

7.3.3.2 Bounding the Regret

The regret analysis is straightforward. Assume that $w_\star \in \widetilde{C}_t$ is satisfied (i.e., the event from Theorem 22 holds) and remember that by the definition, $\phi_t = \mathrm{argmax}_{x_i \in \mathcal{A}} \max_{w \in \widetilde{C}_t} \phi_i^\mathsf{T} w$. We also define $\overline{w}_{t,i} = \mathrm{argmax}_{w \in \widetilde{C}_t} \phi_i^\mathsf{T} w$ as the auxiliary vector which encodes the optimistic behaviour of the algorithm. With a slight abuse of notation, we also use $\star$ as a subscript to indicate the (unknown) index of the optimal arm, so that $\overline{w}_{t,\star} = \mathrm{argmax}_{w \in \widetilde{C}_t} \phi_\star^\mathsf{T} w$. Since $w_\star \in \widetilde{C}_t$, we have that

$$\phi_t^\mathsf{T} \overline{w}_{t,t} \geq {\phi_\star}^\mathsf{T} \overline{w}_{t,\star} \geq {\phi_\star}^\mathsf{T} w_*.$$

We can now bound the instantaneous regret r_t as

$$\begin{aligned}
r_t &= \phi_\star^\intercal w_\star - \phi_t^\intercal w_\star \leq \phi_t^\intercal \overline{w}_{t,t} - \phi_t^\intercal w_\star \\
&= \phi_t^\intercal (\overline{w}_{t,t} - \widehat{w}_t) + \phi_t^\intercal (\widehat{w}_t - w_\star) \\
&= \phi_t^\intercal \widetilde{A}_t^{-1/2} \widetilde{A}_t^{1/2} (\overline{w}_{t,t} - \widehat{w}_t) + \phi_t^\intercal \widetilde{A}_{t-1}^{-1/2} \widetilde{A}_t^{1/2} (\widehat{w}_t - w_\star) \\
&\leq \sqrt{\phi_t^\intercal \widetilde{A}_t^{-1} \phi_t} \left(\|\overline{w}_{t,t} - \widehat{w}_t\|_{\widetilde{A}_t} + \|\widehat{w}_t - w_\star\|_{\widetilde{A}_t} \right) \\
&\leq 2\widetilde{\beta}_t \sqrt{\phi_t^\intercal \widetilde{A}_t^{-1} \phi_t}.
\end{aligned}$$

Summing over t and taking the max over $\widetilde{\beta}_t$, we get

$$R_t \leq 2\widetilde{\beta}_T \sum_{t=1}^{T} \sqrt{\phi_t^\intercal \widetilde{A}_t^{-1} \phi_t} \leq 2\widetilde{\beta}_T \sqrt{T} \sqrt{\sum_{t=1}^{T} \phi_t^\intercal \widetilde{A}_t^{-1} \phi_t} \leq 2\widetilde{\beta}_T \sqrt{T} \sqrt{\alpha \sum_{t=1}^{T} \phi_t^\intercal A_t^{-1} \phi_t}.$$

We can now use once again Proposition 15 to obtain

$$R_T \leq 2\widetilde{\beta}_T \sqrt{\alpha T \sum_{t=1}^{T} \phi_t^\intercal A_t^{-1} \phi_t} = 2\widetilde{\beta}_T \sqrt{\alpha T \sum_{t=1}^{T} \sigma_t^2(\widetilde{x}_t)} \leq 2\widetilde{\beta}_T \sqrt{2\alpha T \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T)}.$$

We can also further upper bound $\widetilde{\beta}_T$ as

$$\begin{aligned}
\widetilde{\beta}_T &= 2\xi \sqrt{\alpha \log(\kappa^2 T) \left(\sum_{s=1}^{T} \widetilde{\sigma}_t^2(x_s) \right) + \log(1/\delta)} + \left(1 + \frac{1}{\sqrt{1-\varepsilon}} \right) \sqrt{\lambda} F \\
&\leq 2\xi \sqrt{\alpha^2 \log(\kappa^2 T) \left(\sum_{s=1}^{T} \sigma_t^2(x_s) \right) + \log(1/\delta)} + \left(1 + \frac{1}{\sqrt{1-\varepsilon}} \right) \sqrt{\lambda} F \\
&\leq 2\xi\alpha \sqrt{\widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T) + \log(1/\delta)} + \left(1 + \frac{1}{\sqrt{1-\varepsilon}} \right) \sqrt{\lambda} F.
\end{aligned}$$

Putting it together, we obtain

$$\begin{aligned}
R_T &\leq 2 \left(2\xi\alpha \sqrt{\widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T) + \log(1/\delta)} \right) \sqrt{2\alpha T \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T)} \\
&\quad + 2 \left(\left(1 + \frac{1}{\sqrt{1-\varepsilon}} \right) \sqrt{\lambda} F \right) \sqrt{2\alpha T \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T)} \\
&\leq 2\xi (2\alpha)^{3/2} \left(\widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T) + \log(1/\delta) \right) + 2 \left(2\sqrt{\alpha} \sqrt{\lambda} F \right) \sqrt{2\alpha T \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T)} \\
&\leq 2(2\alpha)^{3/2} \left(\sqrt{T} \xi \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T) + \sqrt{T} \log(1/\delta) + \sqrt{T \lambda F^2 \widehat{\mathcal{N}}(\lambda, \widetilde{X}_T) \log(\kappa^2 T)} \right).
\end{aligned}$$

Bibliography

[ACW16] Haim Avron, Kenneth L Clarkson, and David P Woodruff. Faster kernel ridge regression using sketching and preconditioning. *arXiv preprint arXiv:1611.03220*, 2016.

[Alv16] Alexandre Alves. Stacking machine learning classifiers to identify higgs bosons at the lhc. *CoRR*, abs/1612.07725, 2016.

[AM15a] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.

[AM15b] Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel methods with statistical guarantees. In *Neural Information Processing Systems*, 2015.

[Aro50] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[ASW13] Haim Avron, Vikas Sindhwani, and David Woodruff. Sketching structured matrices for faster nonlinear regression. In *Advances in Neural Information Processing Systems*, pages 2994–3002, 2013.

[AYPS11] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Neural Information Processing Systems*, 2011.

[Bac13] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *COLT*, volume 30 of *JMLR Proceedings*, pages 185–209. JMLR.org, 2013.

[Bac17] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017.

[BB08] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[BLB04] Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Advanced Lectures on Machine Learning*. 2004.

[BMEWL11] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR*, 2011.

[BPR07] F. Bauer, S. Pereverzev, and L. Rosasco. On regularization algorithms in learning theory. *Journal of complexity*, 23(1):52–72, 2007.

[BSW14] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5:4308, 2014.

[CARR16] Raffaello Camoriano, Tomás Angles, Alessandro Rudi, and Lorenzo Rosasco. Nytro: When subsampling meets early stopping. In *Artificial Intelligence and Statistics*, pages 1403–1411, 2016.

[CAS16] Jie Chen, Haim Avron, and Vikas Sindhwani. Hierarchically compositional kernels for scalable nonparametric learning. *CoRR*, abs/1608.00860, 2016.

[CCL+19] Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 533–557, Phoenix, USA, 25–28 Jun 2019. PMLR.

[CDV07] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[CG17] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, 2017.

[CLM+15] Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform Sampling for Matrix Approximation. In *ITCS*, pages 181–190. ACM, 2015.

[CLV17a] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Distributed adaptive sampling for kernel matrix approximation. In *AISTATS*, 2017.

[CLV17b] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Distributed adaptive sampling for kernel matrix approximation. In *International Conference on Artificial Intelligence and Statistics*, 2017.

[CLV17c] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Second-order kernel online convex optimization with adaptive sketching. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 645–653, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[CLV17d] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Second-order kernel online convex optimization with adaptive sketching. In *International Conference on Machine Learning*, 2017.

[COCF16] Kurt Cutajar, Michael Osborne, John Cunningham, and Maurizio Filippone. Preconditioning kernel matrices. In *International Conference on Machine Learning*, pages 2529–2538, 2016.

[CR18] Daniele Calandriello and Lorenzo Rosasco. Statistical and computational trade-offs in kernel k-means. In *Neural Information Processing Systems*, 2018.

[CRR18] Luigi Carratino, Alessandro Rudi, and Lorenzo Rosasco. Learning with sgd and random features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10213–10224. Curran Associates, Inc., 2018.

[CS02] Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39(1):1–49, 2002.

[CS09] Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.

[CY10] A. Caponnetto and Yuan Yao. Adaptive rates for regularization operators in learning theory. *Analysis and Applications*, 08, 2010.

[DB16] Aymeric Dieuleveut and Francis Bach. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399, 2016.

[DFB17] Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *The Journal of Machine Learning Research*, 18(1):3520–3570, 2017.

[DGBSX12] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

[DGL13] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[DHK08] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*, 2008.

[DMIMW12] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

[DVRC06] Ernesto De Vito, Lorenzo Rosasco, and Andrea Caponnetto. Discretization error analysis for tikhonov regularization. *Analysis and Applications*, 4(01):81–99, 2006.

[DXH+14] Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, pages 3041–3049, 2014.

[FM12] Gregory E Fasshauer and Michael J McCourt. Stable evaluation of gaussian radial basis function interpolants. *SIAM Journal on Scientific Computing*, 34(2):A737–A762, 2012.

[FSC+16] X Yu Felix, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016.

[GLPW16] Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *The SIAM Journal of Computing*, pages 1–28, 2016.

[GOSS16] Alon Gonen, Francesco Orabona, and Shai Shalev-Shwartz. Solving ridge regression using sketched preconditioned svrg. *arXiv preprint arXiv:1602.02350*, 2016.

[GRO+08] L. Lo Gerfo, Lorenzo Rosasco, Francesca Odone, Ernesto De Vito, and Alessandro Verri. Spectral Algorithms for Supervised Learning. *Neural Computation*, 20(7):1873–1897, 2008.

[HAS+14] Po-Sen Huang, Haim Avron, Tara N. Sainath, Vikas Sindhwani, and Bhuvana Ramabhadran. Kernel methods match deep neural networks on timit. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 205–209, 2014.

[HCKB19] Jonathan H. Huggins, Trevor Campbell, Mikołaj Kasprzak, and Tamara Broderick. Scalable Gaussian process inference with finite-data mean and variance guarantees. In *International Conference on Artificial Intelligence and Statistics*, apr 2019.

[HKAK06] Elad Hazan, Adam Tauman Kalai, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. In *Conference on Learning Theory*, 2006.

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel Methods in Machine Learning. *Annals of Statistics*, 36(3), 2008.

[HXGD14] Raffay Hamid, Ying Xiao, Alex Gittens, and Dennis DeCoste. Compact random feature maps. In *International Conference on Machine Learning*, pages 19–27, 2014.

[KCCB19] Ilja Kuzborskij, Leonardo Cella, and Nicolò Cesa-Bianchi. Efficient linear bandits through matrix sketching. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[KMT09] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Ensemble Nystrom Method. In *NIPS*, pages 1060–1068. Curran Associates, Inc., 2009.

[LOSC18] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian process meets big data: A Review of scalable GPs. Technical report, jul 2018.

[LR85] Tze L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[LR17a] Junhong Lin and Lorenzo Rosasco. Generalization properties of doubly online learning algorithms. *arXiv preprint arXiv:1707.00577*, 2017.

[LR17b] Junhong Lin and Lorenzo Rosasco. Optimal rates for learning with nyström stochastic gradient methods. *arXiv preprint arXiv:1710.07797*, 2017.

[LR17c] Junhong Lin and Lorenzo Rosasco. Optimal rates for multi-pass stochastic gradient methods. *Journal of Machine Learning Research*, 18(97):1–47, 2017.

[LRRC18] Junhong Lin, Alessandro Rudi, Lorenzo Rosasco, and Volkan Cevher. Optimal rates for spectral algorithms with least-squares regression over hilbert spaces. *Applied and Computational Harmonic Analysis*, 2018.

[LS19] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. 2019.

[LSS13] Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013.

[MB17] Siyuan Ma and Mikhail Belkin. Diving into the shallows: a computational perspective on large-scale shallow learning. In *Advances in Neural Information Processing Systems*, pages 3778–3787, 2017.

[MGL+17] Avner May, Alireza Bagheri Garakani, Zhiyun Lu, Dong Guo, Kuan Liu, Aurelien Bellet, Linxi Fan, Michael Collins, Daniel J. Hsu, Brian Kingsbury, Michael Picheny, and Fei Sha. Kernel approximation methods for speech recognition. *CoRR*, abs/1701.03577, 2017.

[MK18] Mojmír Mutný and Andreas Krause. Efficient high-dimensional Bayesian optimization with additivity and quadrature Fourier features. In *Neural Information Processing Systems*, 2018.

[MM17] Cameron Musco and Christopher Musco. Recursive Sampling for the Nyström Method. In *NIPS*, 2017.

[Ora14] Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.

[PVRB18a] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Exponential convergence of testing error for stochastic gradient methods. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 250–296, 2018.

[PVRB18b] Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8125–8135. Curran Associates, Inc., 2018.

[QCR05] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[QCRW07] Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Christopher K. I. Williams. Approximation methods for gaussian process regression. *Large-scale kernel machines*, pages 203–224, 2007.

[RCCR18] Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5677–5687. Curran Associates, Inc., 2018.

[RCR13] Alessandro Rudi, Guillermo D Canas, and Lorenzo Rosasco. On the sample complexity of subspace learning. In *Advances in Neural Information Processing Systems*, pages 2067–2075, 2013.

[RCR15] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015.

[RCR17] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. FALKON: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, pages 3891–3901, 2017.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[Rob52] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952.

[RR08] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[RR09] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.

[RR17] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

[RS80] Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics: Vol.: 1.: Functional Analysis*. Academic press, 1980.

[RV15] Lorenzo Rosasco and Silvia Villa. Learning with incremental iterative regularization. In *Advances in Neural Information Processing Systems*, pages 1630–1638, 2015.

[RW06] Carl Edward. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[Saa03] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[SBC17] Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. Lower bounds on regret for noisy Gaussian process bandit optimization. In *Conference on Learning Theory*, 2017.

[SC08] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[SHM+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[SHS+09] Ingo Steinwart, Don R Hush, Clint Scovel, et al. Optimal rates for regularized least squares regression. In *COLT*, 2009.

[SIVA17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. pages 4278–4284, 2017.

[SKKS10] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning*, 2010.

[SS00] Alex J. Smola and Bernhard Schölkopf. Sparse Greedy Matrix Approximation for Machine Learning. In *ICML*, pages 911–918. Morgan Kaufmann, 2000.

[SS02] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. MIT Press, 2002.

[SS15] Bharath Sriperumbudur and Zoltán Szabó. Optimal rates for random fourier features. In *Advances in Neural Information Processing Systems*, pages 1144–1152, 2015.

[SSSSC11] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[SWL03] Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse gaussian process regression. In *Artificial Intelligence and Statistics 9*, number EPFL-CONF-161318, 2003.

[SZ03] Steve Smale and Ding-Xuan Zhou. Estimating the approximation error in learning theory. *Analysis and Applications*, 1(01):17–41, 2003.

[SZ13] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

[Tho33] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

[Tro12] Joel A Tropp. User-Friendly Tools for Random Matrices: An Introduction. 2012.

[Tro15] Joel Aaron Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

[TRVR16] Stephen Tu, Rebecca Roelofs, Shivaram Venkataraman, and Benjamin Recht. Large scale kernel learning using block coordinate descent. *arXiv preprint arXiv:1602.05310*, 2016.

[Vap99] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[VKM+13] Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. In *Uncertainty in Artificial Intelligence*, 2013.

[VRC+05] Ernesto De Vito, Lorenzo Rosasco, Andrea Caponnetto, Umberto De Giovannini, and Francesca Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6(May):883–904, 2005.

[Wah90] Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[WGKJ18] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754, 2018.

[Woo14] David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

[WS01] Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688, 2001.

[YLK17] Xiaotian Yu, Michael R. Lyu, and Irwin King. CBRAP: Contextual bandits with random projection. In *AAAI Conference on Artificial Intelligence*, 2017.

[YPW15] Yun Yang, Mert Pilanci, and Martin J Wainwright. Randomized sketches for kernels: Fast and optimal non-parametric regression. *arXiv preprint arXiv:1501.06195*, 2015.

[YRC07] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

[ZDW13] Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Divide and Conquer Kernel Ridge Regression. In *COLT*, volume 30 of *JMLR Proceedings*, pages 592–617. JMLR.org, 2013.