Mathematics of Double Descent

by

J.C. van der Voort

to obtain the degree of Master of Science
at the Delft University of Technology.

Student number:	4709101	
Project duration:	December 1, 2021 – July 1, 2022	
Thesis committee:	Dr. ir. G.N.J.C. Bierkens,	TU Delft, supervisor
	Dr. M. Loog,	TU Delft

An electronic version of this thesis is available at
`https://repository.tudelft.nl/islandora/search/?collection=education`.

Abstract.
Recently, there has been an increase in literature about the Double Descent phenomenon for heavily over-parameterized models. Double Descent refers to the shape of the test risk curve, which can show a second descent in the over-parameterized regime, resulting in the remarkable combination of both low training and low test risk. However, much is still unknown about this behaviour. In this thesis we consider Double Descent and more specifically 'beneficial overfitting', meaning that the lowest test risk as a function of the number of parameters is achieved in the over-parameterized regime. We are mainly interested in under what conditions beneficial overfitting occurs. We start by exploring the test risk behaviour for simple linear regression models, with isotropic Gaussian, general Gaussian and sub-Gaussian covariates. For random feature selection and isotropic covariance, beneficial overfitting occurs for a large signal-to-noise ratio. For deterministic feature selection and isotropic covariance, beneficial overfitting occurs if we select features corresponding to the lowest weights. Without feature selection, beneficial overfitting occurs if the eigenvalues of the covariance matrix have a long, flat tail. In the second part of this thesis we check whether the same or similar results can be applied to other models as well. Specifically, we look at kernel regression, random Fourier features and a classification model. It seems that the linear regression results agree with the random Fourier features model, linear and quadratic kernel regression and classification model, but are not applicable for the Gaussian kernel regression case. Hence, more factors need to be considered, besides the eigenvalue behaviour of the covariance or kernel matrix and the way in which features are selected, to fully explain Double Descent and beneficial overfitting.

Contents

1 Introduction **1**
- 1.1 Setting . . . 2
- 1.2 Overview . . . 3

2 Isotropic Gaussian covariates **4**
- 2.1 Setting . . . 4
- 2.2 Main Result . . . 4
- 2.3 Implications of Main Result . . . 5
- 2.4 Special cases . . . 6
  - 2.4.1 No feature selection . . . 6
  - 2.4.2 Constant parameterization rate . . . 6
  - 2.4.3 Under- and over-parameterized limit . . . 7
  - 2.4.4 Noiseless case . . . 8
- 2.5 Numerical experiments . . . 8
  - 2.5.1 Influence of choice of true parameter vector . . . 10
  - 2.5.2 Influence of signal-to-noise ratio . . . 11
- 2.6 Infinite dimensional case . . . 12
  - 2.6.1 Constructing our own distribution . . . 12
- 2.7 When do we expect beneficial overfitting? . . . 14

3 General Gaussian covariates **15**
- 3.1 Under-parameterized regime . . . 15
- 3.2 Over-parameterized regime . . . 16

4 Sub-Gaussian covariates **19**
- 4.1 Setting . . . 19
- 4.2 Main Result . . . 20
- 4.3 Implications of Main Result . . . 20
- 4.4 Special case: identity covariance matrix . . . 21
- 4.5 Examples of eigenvalue sequences . . . 22
- 4.6 Numerical experiments . . . 23
  - 4.6.1 Identity covariance matrix . . . 24
  - 4.6.2 Polynomial decay of eigenvalues . . . 24
  - 4.6.3 Exponential decay of eigenvalues . . . 26
  - 4.6.4 Influence of normalizing true parameter vector . . . 27
  - 4.6.5 Influence of label noise . . . 27
- 4.7 Infinite dimensional case . . . 28
- 4.8 When do we expect beneficial overfitting? . . . 28

5 Kernel regression **29**
- 5.1 Setting 1 . . . 29
  - 5.1.1 Special case: linear kernel . . . 31
- 5.2 Numerical experiments for setting 1 . . . 31
  - 5.2.1 Linear kernel . . . 32
  - 5.2.2 Quadratic kernel . . . 33
  - 5.2.3 Gaussian kernel . . . 34
- 5.3 Comparison to previous results for setting 1 . . . 36
- 5.4 Setting 2 . . . 38

5.5 Numerical experiments for setting 2 . . . 38
5.6 Comparison to previous results for setting 2 . . . 40

6 Fourier features **42**
6.1 Setting . . . 42
6.2 Main Result . . . 43
6.3 Implications of Main Result . . . 44
6.4 Special cases . . . 44
6.4.1 Highly over-parameterized limit . . . 44
6.4.2 No feature selection . . . 44
6.5 Numerical experiments . . . 44
6.6 Comparison to previous results . . . 47

7 Classification model **48**
7.1 Setting . . . 48
7.2 Main Result . . . 49
7.3 Numerical experiments . . . 49
7.4 Comparison to previous results . . . 50

8 Conclusion and Discussion **51**

9 Appendix **58**
9.1 Proof of Theorem 1 . . . 58
9.2 Proof of Lemma 1 . . . 62
9.3 Proof of Theorem 4 . . . 63
9.4 R code . . . 76

1 Introduction

Recently, heavily over-parameterized machine learning models, such as in Zhang et al. (2017) [1] and Brock et al. (2021) [2], have shown unexpectedly high accuracy, which cannot be explained by existing theoretical results. This has resulted in an increase in literature about over-parameterized models, see e.g. [3], [4], [5], [6] and [7]. However, there is still much unknown about the performance of these over-parameterized machine learning models.

Classically, the performance of machine learning algorithms is determined by a trade-off between the bias and variance of the model estimate (see e.g. Hastie et al. (2008) [8]). In the under-parameterized regime, where the number of parameters is smaller than the number of data points, we often observe a U-shaped curve, see the left part of Figure 1.1. For a small number of parameters we have low variance and high bias, resulting in high test risk. If we increase the number of parameters, then the test risk will decrease and we will reach the lowest test risk. Further increasing the number of parameters will result in high variance and small bias, with an explosion in variance at the interpolation threshold. Even further increasing the number of parameters means that we are overfitting the model: the model is very good at fitting the training data, resulting in a low training risk, but will have worse performance on the unseen test data, resulting in a high test risk.

However, some machine learning algorithms are actually able to achieve both low training risk and low test risk. In these modern machine learning algorithms, the number of parameters is often very large, much larger than the number of data points available. Hence, we would expect these models to be overfitting, but surprisingly this is not always the case. There are a number of other historical examples of literature in which this phenomenon also occurs, see e.g. the list described in Loog et al. (2020) [9]. This phenomenon is described by the so-called Double Descent risk curve, which has been named and extensively described in the paper of Belkin et al. (2019) [3]. The name Double Descent refers to the fact that the test risk, as a function of the number of parameters, first decreases in the under-parameterized regime according to the classical U-curve, but then also decreases a second time in the over-parameterized regime, see Figure 1.1 below, which is Figure 1B from Belkin et al. (2019) [3].

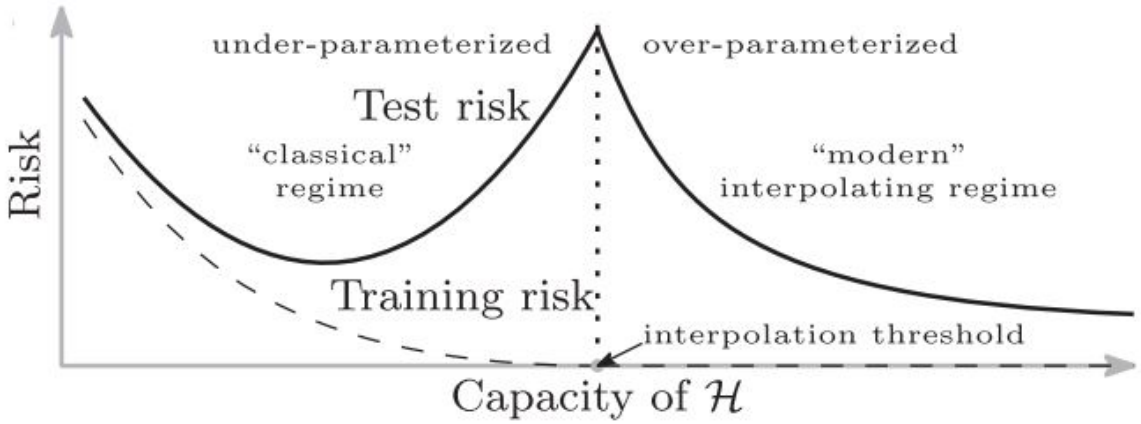

Figure 1.1: Figure 1B from Belkin et al. (2019) [3], showing the Double Descent risk curve. The capacity of $\mathcal{H}$ refers to how large the class $\mathcal{H}$ of possible predictors is, which can be measured by for example the number of parameters.

The reason why this second descent in the over-parameterized regime is possible is explained by the so-called 'inductive bias' [3], which refers to the smoothness or the regularity of a function, which can be measured by for example the norm of the parameter vector in linear regression problems. Here the bias means that we have a preference for the smoothest or simplest solution. In this sense, we are still looking for the simplest solution, not with the least amount of parameters, but with the smallest parameter vector norm. Hence finding the solution with minimum norm in the over-parameterized regime is related to finding the solution with lowest test risk in this regime.

In this thesis we will mainly focus on the occurrence of Double Descent in a simple setting, namely the linear regression case with (sub)-Gaussian covariates, which is easier to analyze and more well-understood. This setting has been discussed in a.o. [4], [5] and [6]. The case for Uniform covariates is considered in [7], in the asymptotic regime as the input dimension diverges to infinity. Double Descent is not limited to regression models. It has also been shown in classification models, see [10] and [11].

The goal of this thesis is to provide an introduction and overview of the Double Descent phenomenon in a simple setting, and check whether similar results can be applied to other models. Hence, we first consider the linear regression setting with increasingly more general choices for the vector of covariates. We will not prove any new theoretical results, but we will consider some of the existing theoretical proofs. Furthermore, we try to find connections between the theoretical results and verify them through Monte Carlo simulations performed in R. After establishing results for the simple linear regression case, we check whether the same or similar results can be applied to other models. Specifically, we consider kernel regression, random Fourier features and classification.

We will distinguish between Double Descent, benign overfitting and 'beneficial overfitting'. Double Descent refers to the shape of the risk curve as a function of the number of parameters, as we have described above. Benign overfitting refers to the over-parameterized solution having 'near-optimal prediction accuracy', according to the definition used in [5], with which they mean that the excess test risk is close to zero, and converges to zero as the number of data points diverges to infinity. In practical applications, using heavily over-parameterized models is only interesting if they provide improved performance compared to under-parameterized models. Hence, the main question we are interested in and try to answer in this thesis is:

Under what conditions does the optimal test risk lie in the over-parameterized regime?

Here 'optimal' refers to the global minimum of the test risk curve as a function of the number of parameters. This is different from benign overfitting, as we require better performance in the over-parameterized regime relative to the under-parameterized regime, whereas benign overfitting is only concerned with the absolute performance of the over-parameterized solution, regardless of the under-parameterized regime. Hence, we will refer to this as 'beneficial overfitting' and so we are mainly interested in under what conditions beneficial overfitting occurs.

1.1 Setting

Throughout this thesis we will mostly consider (variations of) the linear regression model, which is given by

$$y = x^T\theta + \varepsilon, \tag{1.1}$$

with response variable $y \in \mathbb{R}$, unknown parameter vector $\theta \in \mathbb{R}^p$, vector of covariates $x \in \mathbb{R}^p$ and noise term $\varepsilon \in \mathbb{R}$. We assume x is generated according to some probability distribution P_x and $\varepsilon \sim N(0, \sigma^2)$ for some $\sigma > 0$. We generate n iid training data points, denoted by $(x_i, y_i)_{i=1}^n$, where $x_i \sim P_x$ and $y_i = x_i^T\theta + \varepsilon_i$, with ε_i independent of x_i and θ the unknown parameter. We can collect the data in a design matrix $X \in \mathbb{R}^{n\times p}$ with rows $x_i^T \in \mathbb{R}^p$ and response vector $y \in \mathbb{R}^n$ with entries y_i. Then we find the following matrix-vector form of equation (1.1),

$$y = X\theta + \varepsilon, \tag{1.2}$$

where $\varepsilon \in \mathbb{R}^n$ is the noise vector with entries ε_i. As an estimator for θ, in the case that $p < n$, we will consider the usual least-squares estimator $\hat{\theta}$, which is defined as

$$\hat{\theta} = \text{argmin}_{\theta \in \mathbb{R}^p} ||y - X\theta||^2.$$

This minimization problem is well-known, and results in the normal equations $X^T X\theta = X^T y$, which has the solution

$$\hat{\theta} = (X^T X)^\dagger X^T y = X^T (XX^T)^\dagger y = X^\dagger y,$$

where $\dagger$ denotes the pseudo-inverse. For simplicity, we assume that $\text{rank}(X) = \min\{p, n\}$, so that degenerate cases are excluded. Then the pseudo-inverse $X^\dagger$ is given by

$$X^\dagger = \begin{cases} (X^T X)^{-1} X^T & \text{if } p \leq n \\ X^T (XX^T)^{-1} & \text{if } p > n \end{cases} \tag{1.3}$$

In the over-parameterized regime ($p > n$), least-squares minimization has no unique solution. It is well-known that the pseudo-inverse solution $\hat{\theta} = X^\dagger y$ is the solution with minimum norm. This is crucial for the Double Descent behaviour, as it creates a form of 'inductive bias' [3], which allows for the second descent to happen. The minimum norm solution solves the following minimization problem

$$\hat{\theta} = \text{argmin}_{\theta \in \mathbb{R}^p} ||\theta||_2 \quad \text{s.t.} \quad X\theta = y.$$

Since $X\theta = y$, the minimum norm solution is also called the interpolating solution. In order to determine the accuracy of the estimate $\hat{\theta}$, we look at the expected quadratic loss

$$R(\hat{\theta}) = \mathbb{E}_{x,y}(y - x^T \hat{\theta})^2,$$

where the expectation is taken with respect to $x \sim P_x$ and $y = x^T \theta + \varepsilon$. This is also called the test risk. Some papers, such as [5], instead look at the excess risk, which is defined as

$$R_{excess}(\hat{\theta}) = \mathbb{E}_{x,y}(y - x^T \hat{\theta})^2 - \mathbb{E}_{x,y}(y - x^T \theta)^2,$$

where θ is the true parameter vector. The excess risk shows how good the performance is compared to the optimal case.

1.2 Overview

In the subsequent chapters of this thesis, we consider the linear regression problem from equation (1.1). In each chapter we consider a different distribution for the covariates vector, which are increasingly more general. In Chapter 2 we consider the isotropic Gaussian case where $x \sim N(0, I_d)$. Furthermore, we look at the influence of random feature selection, some limit cases and the infinite dimensional setting. In Chapter 3 we assume more general Gaussian covariates, that is $x \sim N(0, \Sigma)$ with Σ a diagonal matrix with positive entries on the diagonal. In Chapter 4 we consider covariates with a more general sub-Gaussian distribution. Once we have established an understanding of Double Descent in these simpler settings, we leave the linear regression case and see whether we can apply the same or similar results in other models. In Chapter 5 we start with kernel regression, where we discuss the linear kernel, quadratic kernel and Gaussian kernel. We consider 2 settings: applying a kernel estimator to finite dimensional linear regression and applying a finite dimensional kernel estimator to an infinite dimensional kernel regression problem. Related to kernel regression, in Chapter 6 we consider random Fourier features. That is, we assume that we can express the solution in terms of Fourier basis functions of the form $\phi(x) = e^{-i\omega x}$. The previously mentioned chapters describe Double Descent for regression problems. In Chapter 7 we briefly look at Double Descent in a simple classification problem. We compare the results, answer the main question and give recommendations for future research in Chapter 8. Finally, in Chapter 9 the Appendix is given, where we have included proofs and the R code for our numerical experiments.

2 Isotropic Gaussian covariates

In this chapter we consider the linear regression problem from equation (1.1) with isotropic Gaussian covariates. This is the setting which is described in Belkin et al. (2020) [4], including random feature selection. They discuss both a 'prescient' choice and a random choice of features. They show that, for the random choice, beneficial overfitting occurs if the signal-to-noise ratio is large. No beneficial overfitting occurs for the 'prescient' choice. We will verify their results experimentally, consider some limit cases and discuss the infinite dimensional setting.

2.1 Setting

First, we introduce some notation. For a subset $P \subset \{1, \dots, d\}$ and vector $v \in \mathbb{R}^d$, define

$$v_P := (v_j : j \in P)^T \in \mathbb{R}^{|P|}$$

which is the sub-vector of v using only entries with index in P. Similarly, for a matrix $X \in \mathbb{R}^{n \times d}$, define

$$X_P := [x_P^{(1)} \cdots x_P^{(n)}]^T \in \mathbb{R}^{n \times |P|}$$

which is the n by $|P|$ design matrix using only entries with index in P.

Belkin et al. (2020) [4] consider the linear regression problem (see equation (1.1)) with $x \sim N(0, I_d)$. The goal is to find the estimate $\hat{\theta}$ using only a subset $P \subset \{1, \dots, d\}$ of the d features in the vector x, such that $|P| = p$. Then $\hat{\theta}$ is defined as

$$\hat{\theta}_P = X_P^\dagger y, \qquad \hat{\theta}_{P^c} = 0$$

with $X_P \in \mathbb{R}^{n \times p}$ the design matrix using the column indices in P. Notice that

$$X\hat{\theta} = (X_P X_{P^c})(\hat{\theta}_P, \hat{\theta}_{P^c})^T = X_P \hat{\theta}_P + X_{P^c}\hat{\theta}_{P^c} = X_P \hat{\theta}_P.$$

From this we can see that the minimization problem to construct $\hat{\theta}_P$ is the same as for $\hat{\theta}$, but restricted to using indices in P. Hence $X_P^\dagger$ is equal to the expression given in (1.3) with X replaced by X_P.

2.2 Main Result

The main result in [4] provides an exact expression for the test risk of the minimum norm solution $\hat{\theta}$, restricting ourselves to a subset of p out of the d features.

Theorem 1 (Theorem 1 and Corollary 1 in [4])**.** *Assume $x \sim N(0, I_d)$, $\varepsilon \sim N(0, \sigma^2)$ and $y = x^T\theta + \varepsilon$ for some $\theta \in \mathbb{R}^d$. Choose $p \in \{0, \dots, d\}$ and $P \subset \{1, \dots, d\}$ such that $|P| = p$. Consider the min-norm solution $\hat{\theta}$, for which $\hat{\theta}_P = X_P^\dagger y$ and $\hat{\theta}_{P^c} = 0$. Then the test risk of $\hat{\theta}$ is*

$$R_{det}(\hat{\theta}) = \begin{cases} (||\theta_{P^c}||^2 + \sigma^2)(1 + \frac{p}{n-p-1}) & \text{if } p \le n-2 \\ ||\theta_P||^2(1 - \frac{n}{p}) + (||\theta_{P^c}||^2 + \sigma^2)(1 + \frac{n}{p-n-1}) & \text{if } p \ge n+2 \end{cases} \tag{2.1}$$

and $R_{det}(\hat{\theta}) = \infty$ otherwise. If we take P a uniformly random subset of $\{1, \dots, d\}$ with $|P| = p$, then

$$R_{rand}(\hat{\theta}) = \begin{cases} ((1 - \frac{p}{d})||\theta^*||^2 + \sigma^2)(1 + \frac{p}{n-p-1}) & \text{if } p \le n-2 \\ ||\theta^*||^2(1 - \frac{n}{d}(2 - \frac{d-n-1}{p-n-1})) + \sigma^2(1 + \frac{n}{p-n-1}) & \text{if } p \ge n+2 \end{cases} \tag{2.2}$$

and $R_{rand}(\hat{\theta}) = \infty$ otherwise.

Proof. See the proof in Appendix 9.1.

2.3 Implications of Main Result

In this section we consider the implications of Theorem 1. We focus on R_{rand}, since we can more easily analyze these expressions analytically. We investigate when Double Descent and beneficial overfitting can occur.

In the under-parameterized regime ($p \leq n-2$), we have

$$R_{rand}(p) = ((1-\frac{p}{d})||\theta^*||^2 + \sigma^2)(1+\frac{p}{n-p-1}).$$

Assuming that $d > n-1$, above expression has a strictly positive derivative if

$$\frac{d}{dp}R_{rand}(p) = -\frac{1}{d}||\theta^*||^2\frac{n-1}{n-p-1} + ((1-\frac{p}{d})||\theta^*||^2 + \sigma^2)\frac{n-1}{(n-p-1)^2} > 0.$$

$$-(n-p-1)||\theta^*||^2 + d(1-\frac{p}{d})||\theta^*||^2 + d\sigma^2 > 0.$$

$$-(n-1)||\theta^*||^2 + p||\theta^*||^2 + d||\theta^*||^2 - p||\theta^*||^2 + d\sigma^2 > 0.$$

$$(d-n+1)||\theta^*||^2 + d\sigma^2 > 0.$$

This is always satisfied when we choose d large enough such that $d > n-1$. For $d < n-1$ overfitting would not be possible, as $p < n$ always. Hence we assume $d > n-1$, so $\frac{d}{dp}R_{rand}(p) > 0$ and $R_{rand}(p)$ is increasing in p from $p = 0$ to $p = n-2$. So the minimum value, denoted by R^*_{under}, is achieved at $p = 0$,

$$R^*_{under} = R_{rand}(0) = ||\theta^*||^2 + \sigma^2.$$

In the over-parameterized regime ($p \geq n+2$), we have

$$R_{rand}(p) = ||\theta^*||^2(1-\frac{n}{d}(2-\frac{d-n-1}{p-n-1})) + \sigma^2(1+\frac{n}{p-n-1}).$$

Then for the derivative wrt p we have

$$\frac{d}{dp}R_{rand}(p) = -\frac{n}{d}||\theta^*||^2\frac{d-n-1}{(p-n-1)^2} - \sigma^2\frac{n}{(p-n-1)^2} < 0.$$

This is negative when we choose d large enough such that $d > n+1$. Since $R'_{rand}(p) < 0$, $R_{rand}(p)$ is decreasing in p from $p = n+2$ to $p = d$. So the minimum value, denoted by R^*_{over}, is achieved at $p = d$,

$$R^*_{over} = R_{rand}(p) = ||\theta^*||^2(1-\frac{n}{d}) + \sigma^2(1+\frac{n}{d-n-1}).$$

Assuming $d > n+1$, beneficial overfitting occurs if $R^*_{over} < R^*_{under}$, that is

$$(1-\frac{n}{d})||\theta^*||^2 + (1+\frac{n}{d-n-1})\sigma^2 < ||\theta^*||^2 + \sigma^2.$$

Hence for beneficial overfitting we need the following condition to be satisfied

$$\frac{||\theta^*||^2}{\sigma^2} > \frac{d}{d-n-1} \tag{2.3}$$

So when the signal-to-noise ratio $\frac{||\theta^*||^2}{\sigma^2}$ is large enough, the global minimum of the prediction risk is achieved in the over-parameterized regime and beneficial overfitting occurs. This agrees well with machine learning practice in which we often assume a small amount of label noise, resulting in a large signal-to-noise ratio. In the next section, we check whether condition (2.3) also holds in some special cases. In the section after that, we show the Double Descent behaviour in numerical experiments.

2.4 Special cases

In this section we will consider some special cases of Theorem 1. We look at the case when there is no feature selection, the case of constant parameterization rate as $d \to \infty$, the under- and over-parameterized limit and the noiseless case ($\sigma = 0$).

2.4.1 No feature selection

In case of no feature selection, we have $p = d$. Then $P = \{1, \dots, d\}$ and $P^c = \emptyset$. So

$$||\theta^*_P||^2 = ||\theta^*||^2, \qquad ||\theta^*_{P^c}||^2 = 0.$$

Now the result of Theorem 1 reduces to

$$R(\hat{\theta}) = \begin{cases} \sigma^2(1 + \frac{p}{n-p-1}) & \text{if } p \leq n-2 \\ ||\theta^*||^2(1 - \frac{n}{p}) + \sigma^2(1 + \frac{n}{p-n-1}) & \text{if } p \geq n+2 \end{cases}$$

and $R(\hat{\theta}) = \infty$ otherwise. Let us investigate in which regime the optimal test risk lies. In the under-parameterized case we have

$$R(p) = \sigma^2 \left(1 + \frac{p}{n-p-1}\right) = \sigma^2 \frac{n-1}{n-p-1}.$$

Hence $R'(p) = \sigma^2 \frac{n-1}{(n-p-1)^2} > 0$. We see that $R(p)$ is increasing in p, so the minimum is at $p = 0$ and $R^*_{under} = R(0) = \sigma^2$, where R^*_{under} denotes the optimal test risk value in the under-parameterized regime. In the over-parameterized case, we have

$$R(p) = ||\theta^*||^2(1 - \frac{n}{p}) + \sigma^2(1 + \frac{n}{p-n-1}) > 0 + \sigma^2 = \sigma^2.$$

Hence, without feature selection, the minimum test risk is achieved in the under-parameterized regime and we do not expect beneficial overfitting.

2.4.2 Constant parameterization rate

Another interesting case is when we have a constant parameterization rate. This is a common assumption in the literature (see e.g. [6] and [7]). Let the parameterization rate γ be defined as

$$\gamma = \lim_{d \to \infty} \frac{p(d)}{n(d)}.$$

For finite p, n and deterministic choice of the set P, the test risk satisfies formula (2.1). We can express this test risk in terms of γ as follows

$$\lim_{d\to\infty}(1 + \frac{p}{n-p-1}) = \lim_{d\to\infty} \frac{n-1}{n-p-1} = \lim_{d\to\infty} \frac{1 - 1/n(d)}{1 - p(d)/n(d) - 1/n(d)} = \frac{1}{1-\gamma}.$$

$$\lim_{d\to\infty}(1 - \frac{n(d)}{p(d)}) = 1 - \frac{1}{\gamma}.$$

$$\lim_{d\to\infty}(1 + \frac{n}{p-n-1}) = \lim_{d\to\infty} \frac{p-1}{p-n-1} = \lim_{d\to\infty} \frac{p(d)/n(d) - 1/n(d)}{p(d)/n(d) - 1 - 1/n(d)} = \frac{\gamma}{\gamma - 1}.$$

So we find

$$R_{det}(\hat{\theta}) = \begin{cases} (||\theta_{P^c}||^2 + \sigma^2)\frac{1}{1-\gamma} & \text{if } \gamma < 1 \\ ||\theta_P||^2(1 - \frac{1}{\gamma}) + (||\theta_{P^c}||^2 + \sigma^2)\frac{\gamma}{\gamma-1} & \text{if } \gamma > 1 \end{cases} \tag{2.4}$$

For the case where we take P to be a uniformly random subset of $\{1, \dots, d\}$ of size $p(d)$, define

$$\rho := \lim_{d \to \infty} \frac{p(d)}{d}$$

Then

$$\lim_{d\to\infty} \mathbb{E}||\theta_P||^2 = \lim_{d\to\infty} \frac{p(d)}{d}||\theta||^2 = \rho||\theta||^2.$$

$$\lim_{d\to\infty} \mathbb{E}||\theta_{P^c}||^2 = \lim_{d\to\infty} \left(1 - \frac{p(d)}{d}\right)||\theta||^2 = (1-\rho)||\theta||^2.$$

Taking expectation with respect to P of formula (2.4), we find

$$R_{rand}(\hat{\theta}) = \begin{cases} ((1-\rho)||\theta||^2 + \sigma^2)\frac{1}{1-\gamma} & \text{if } \gamma < 1 \\ \rho||\theta||^2(1-\frac{1}{\gamma}) + ((1-\rho)||\theta||^2 + \sigma^2)\frac{\gamma}{\gamma-1} & \text{if } \gamma > 1 \end{cases}$$

We can rewrite this as

$$R_{rand}(\hat{\theta}) = \begin{cases} \frac{1-\rho}{1-\gamma}||\theta||^2 + \frac{1}{1-\gamma}\sigma^2 & \text{if } \gamma < 1 \\ \left(\rho\frac{\gamma-1}{\gamma} + (1-\rho)\frac{\gamma}{\gamma-1}\right)||\theta||^2 + \frac{\gamma}{\gamma-1}\sigma^2 & \text{if } \gamma > 1 \end{cases}$$

Notice that $\rho = 1$ corresponds to no feature selection. In that case

$$R_{rand}(\hat{\theta}) = \mathbb{E}(y - x^T\hat{\theta})^2 = \begin{cases} \sigma^2\frac{1}{1-\gamma} & \text{if } \gamma < 1 \\ ||\theta||^2(1-\frac{1}{\gamma}) + \sigma^2\frac{\gamma}{\gamma-1} & \text{if } \gamma > 1 \end{cases}$$

This is also one of the settings that is discussed in Hastie et al. (2020) [6], who state that

$$\mathbb{E}||\theta - \hat{\theta}||^2 = \begin{cases} \sigma^2\frac{\gamma}{1-\gamma} & \text{if } \gamma < 1 \\ ||\theta||^2(1-\frac{1}{\gamma}) + \sigma^2\frac{1}{\gamma-1} & \text{if } \gamma > 1 \end{cases}$$

Using that $\mathbb{E}(y - x^T\hat{\theta})^2 = \mathbb{E}||\theta - \hat{\theta}||^2 + \sigma^2$, we see that this agrees with the formula for $R_{rand}(\hat{\theta})$. Let us investigate whether beneficial overfitting is possible in this case of constant parameterization rate and no feature selection. In the under-parameterized regime ($\gamma < 1$), we have

$$R_{rand}(\gamma) = \sigma^2\frac{1}{1-\gamma},$$

which has derivative

$$\frac{d}{d\gamma}R_{rand}(\gamma)\sigma^2\frac{1}{(1-\gamma)^2} > 0.$$

Hence, $R_{rand}(\hat{\theta})$ is increasing in γ with a minimum at $\gamma = 0$, so

$$R^*_{under} = R_{rand}(0) = \sigma^2$$

In the over-parameterized regime ($\gamma > 1$), we have

$$R_{rand}(\gamma) = ||\theta||^2(1-\frac{1}{\gamma}) + \sigma^2\frac{\gamma}{\gamma-1} > \sigma^2\left(1 + \frac{1}{\gamma-1}\right) > \sigma^2 = R^*_{under}$$

Hence, for a constant parameterization rate as $d \to \infty$, the minimal test risk always lies in the under-parameterized regime.

2.4.3 Under- and over-parameterized limit

Let us now consider the under- and over-parameterized limit, both for R_{det} and R_{rand}. In the under-parameterized limit we have $p \to 0$. This yields

$$\lim_{p\to 0} R_{det}(\hat{\theta}) = \lim_{p\to 0}(||\theta^*_{P^c}||^2 + \sigma^2)(1 + \frac{p}{n-p-1}) = ||\theta^*||^2 + \sigma^2.$$

$$\lim_{p\to 0} R_{rand}(\hat{\theta}) = \lim_{p\to 0}((1-\frac{p}{d})||\theta^*||^2 + \sigma^2)(1 + \frac{p}{n-p-1}) = ||\theta^*||^2 + \sigma^2.$$

In the over-parameterized limit, we have $p \to \infty$, which yields

$$\lim_{p\to\infty} R_{det}(\hat{\theta}) = \lim_{p\to\infty} ||\theta^*_P||^2(1-\frac{n}{p}) + (||\theta^*_{P^c}||^2 + \sigma^2)(1+\frac{n}{n-p-1}) = ||\theta^*||^2 + \sigma^2.$$

For $R_{rand}(\hat{\theta})$ we first let $p \to d$ and then take $d \to \infty$. Then

$$\lim_{d\to\infty}\lim_{p\to d} R_{rand}(\hat{\theta}) = \lim_{d\to\infty}\lim_{p\to d} ||\theta^*||^2(1-\frac{n}{d}(2-\frac{d-n-1}{p-n-1})) + \sigma^2(1+\frac{n}{p-n-1})$$

$$= \lim_{d\to\infty} ||\theta^*||^2(1-\frac{n}{d}) + \sigma^2(1+\frac{n}{d-n-1}) = ||\theta^*||^2 + \sigma^2.$$

Notice that in the limit, there is no difference between a deterministic choice of P or a random choice of P. This makes sense intuitively: when we select no features ($p = 0$) or all possible features ($p = \infty$), then it does not matter which selection procedure we used. The limits also tell us that if we just keep increasing the number of parameters indefinitely, we will not reach 0 test risk, but a constant value of $||\theta^*||^2 + \sigma^2$.

2.4.4 Noiseless case

In the noiseless case we have $\sigma = 0$. This may seem like an uninteresting scenario, but in machine learning we often assume very small label noise for our data sets. Substituting $\sigma = 0$ in the formulas of Theorem 1, we get

$$R_{det}(\hat{\theta}) = \begin{cases} ||\theta^*_{P^c}||^2(1+\frac{p}{n-p-1}) & \text{if } p \le n-2 \\ ||\theta_P||^2(1-\frac{n}{p}) + ||\theta^*_{P^c}||^2(1+\frac{n}{p-n-1}) & \text{if } p \ge n+2 \end{cases}$$

$$R_{rand}(\hat{\theta}) = \begin{cases} ||\theta^*||^2(1-\frac{p}{d})(1+\frac{p}{n-p-1}) & \text{if } p \le n-2 \\ ||\theta^*||^2(1-\frac{n}{d}(2-\frac{d-n-1}{p-n-1})) & \text{if } p \ge n+2 \end{cases}$$

In this case, condition (2.3) for beneficial overfitting for $R_{rand}(\hat{\theta})$ is always satisfied as the signal-to-noise ratio is infinite.

2.5 Numerical experiments

In this section we will verify the results of Theorem 1 and investigate the influence of some of the parameters. The R code for this and all other numerical experiments can be found in Appendix 9.4. In this experiment, we perform $M = 100$ Monte Carlo iterations, use $n = 40$ training data points, set dimension $d = 100$ and we vary the number of parameters p from 1 to 100. This is similar to the setup in [4]. We use the following Monte Carlo scheme:

- Simulate training data $X \in \mathbb{R}^{n\times d}$ with rows drawn from $N(0, I_d)$.
- (deterministic choice): take $P = \{1, \dots, p\}$
  (random choice): sample P uniformly from $\{1, \dots, d\}$ s.t. $|P| = p$.
- Sample label noise $\varepsilon \in \mathbb{R}^n$ with entries drawn from $N(0, \sigma^2)$.
- Generate true labels: $y = X\theta^* + \varepsilon$.
- Calculate least-squares / min-norm solution $\hat{\theta}$

  $$(p \ge n): \quad \hat{\theta}_P = X_P(X_P X_P^T)^{-1}y, \qquad (p < n): \quad \hat{\theta}_P = (X_P^T X_P)^{-1}X_P^T y$$

  Furthermore, set $\hat{\theta}_{P^c} = 0$.

- Calculate training and test risk

$$R_{train} = \frac{1}{n}\sum_{i=1}^{n}(y_i - X[i, 1:d]\hat{\theta})^2, \qquad R_{test} = \sigma^2 + ||\hat{\theta} - \theta^*||^2$$

We repeat above steps $M = 100$ times and take averages over the R_{train} and R_{test} values. Similar to the setting in [4], we take for the true parameter vector

$$\theta_j^* \propto \frac{1}{j}, \qquad \text{s.t.} \qquad ||\theta^*||^2 = 1,$$

and label noise $\sigma = 1/5$. For R_{det}, where we take $P = \{1, \ldots, p\}$, using $\theta_j^* \propto \frac{1}{j}$ corresponds to choosing the p most important directions (i.e. directions with the largest weights). This would be the case for example if we first perform a LASSO regression to select the features. In Figures 2.1 and 2.2 we plot the average training and test risks over $M = 100$ Monte Carlo iterations, both for R_{det} and R_{rand}.

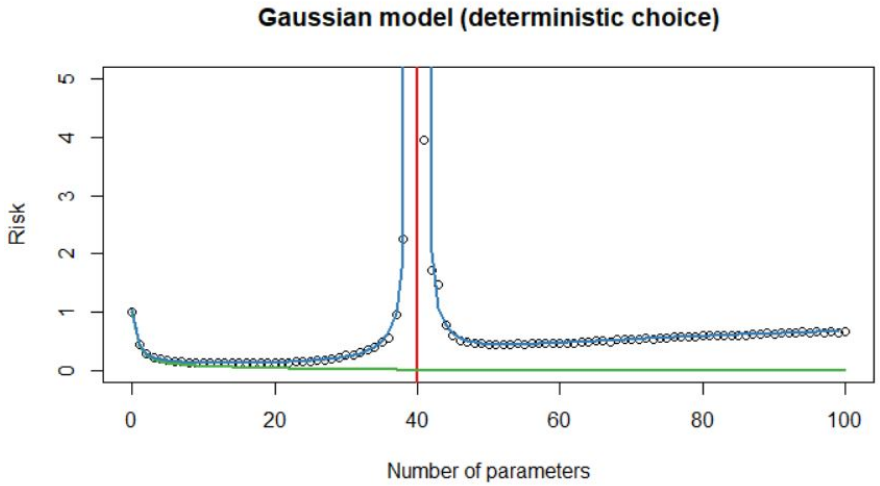

Figure 2.1: Deterministic choice of P, with $\theta_j^* \propto \frac{1}{j}$. Shown are the averages over $M = 100$ Monte Carlo iterations. Green line: training risk, blue line: theoretical test risk, black dots: experimental test risk, red line: interpolation threshold.

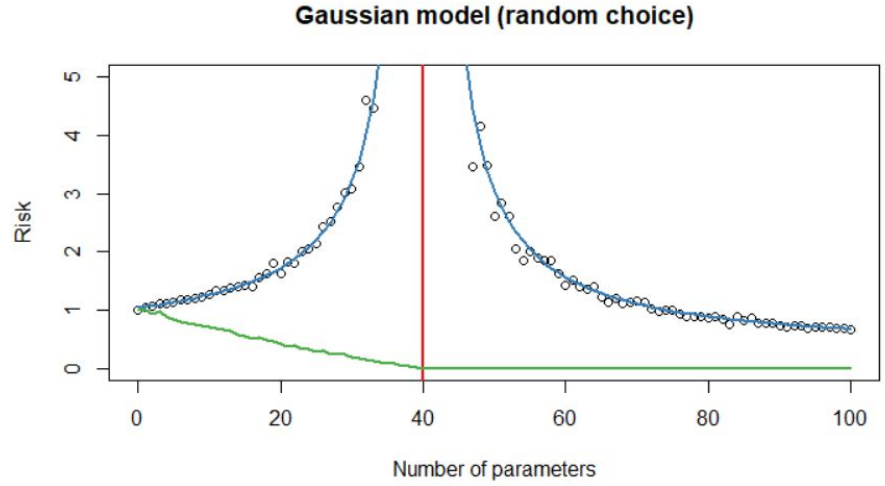

Figure 2.2: Uniformly random choice of P, with $\theta_j^* \propto \frac{1}{j}$. Shown are the averages over $M = 100$ Monte Carlo iterations. Green line: training risk, blue line: theoretical test risk, black dots: experimental test risk, red line: interpolation threshold.

From these graphs, we see that the experimental results agree very well with the theoretical results from Theorem 1. Furthermore, there is a clear difference in test risk behaviour between the deterministic choice of P and the uniformly random choice of P. Both show Double Descent behaviour, but only for the random choice of P there is beneficial overfitting where the minimum test risk lies in the over-parameterized regime.

2.5.1 Influence of choice of true parameter vector

Let us now investigate the influence of the choice of the true parameter vector. Previously, we used $\theta_j^* \propto \frac{1}{j}$. In this subsection we will also consider the case where

$$\theta_j^* \propto j \qquad \text{s.t.} \qquad ||\theta^*|| = 1.$$

In Figures 2.3 and 2.4 we plot average test risks over $M = 100$ Monte Carlo iterations for both the deterministic and random choice of P, using $\sigma = 1/5$. In this setting, for R_{det} where $P = \{1, \ldots, p\}$, using $\theta_j^* \propto j$ corresponds to choosing the least important directions (so the directions with the lowest weights).

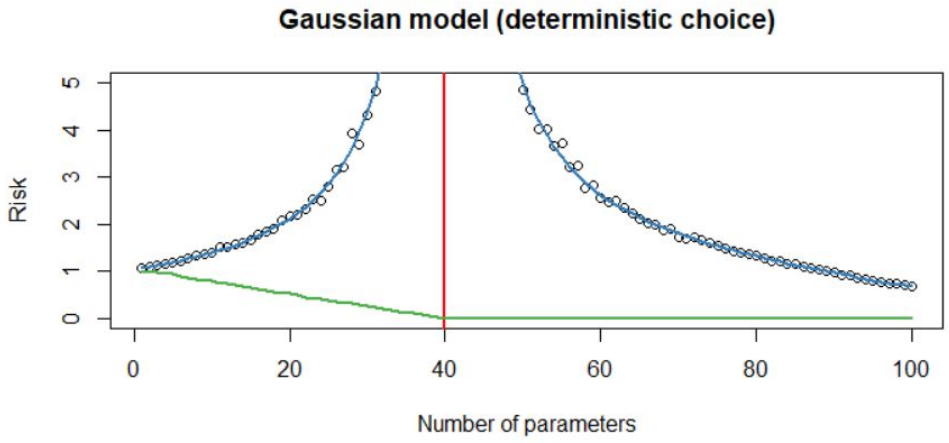

Figure 2.3: Deterministic choice of P, with $\theta_j^* \propto j$. Shown are the averages over $M = 100$ Monte Carlo iterations. Green line: training risk, blue line: theoretical test risk, black dots: experimental test risk, red line: interpolation threshold.

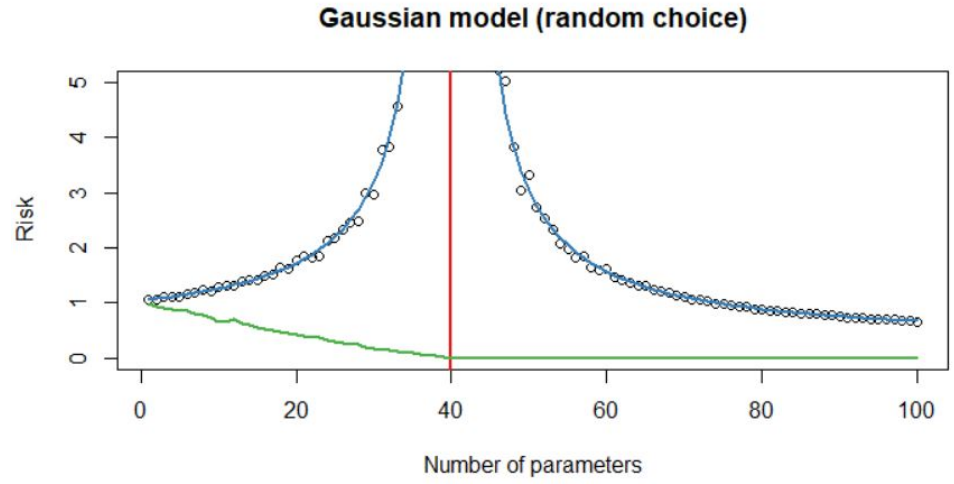

Figure 2.4: Uniformly random choice of P, with $\theta_j^* \propto j$. Shown are the averages over $M = 100$ Monte Carlo iterations. Green line: training risk, blue line: theoretical test risk, black dots: experimental test risk, red line: interpolation threshold.

This time both choices of P show similar behaviour: beneficial overfitting occurs with the minimum test risk lying in the over-parameterized regime. For R_{rand}, there is no difference in test risk behaviour if we use $\theta_j^* \propto j$ instead of $\theta_j^* \propto \frac{1}{j}$. This is as expected, since we are choosing the features at random, so the behaviour of the weights does not matter. For R_{det}, there is a clear difference in test risk behaviour. It seems that selecting features with the lowest weights (weak features) is beneficial for obtaining beneficial overfitting.

2.5.2 Influence of signal-to-noise ratio

Previously, we used $||\theta^*|| = 1$ and $\sigma = 1/5$. This means that we have a signal-to-noise ratio of 25. Recall from condition (2.3) that beneficial overfitting occurs if

$$\frac{||\theta^*||^2}{\sigma^2} > \frac{d}{d-n-1} = \frac{100}{39}$$

where $d = 100$ and $n = 40$. So for $\sigma = 1/5$, we would expect beneficial overfitting. In general, if we fix $||\theta^*||^2 = 1$, then σ should satisfy

$$\sigma < \sqrt{\frac{d-n-1}{d}}.$$

We will verify this through a Monte Carlo experiment. We try 3 different values for σ

$$\sigma_1 = \frac{1}{2}\sqrt{\frac{d-n-1}{d}}, \qquad \sigma_2 = \sqrt{\frac{d-n-1}{d}}, \qquad \sigma_3 = 2\sqrt{\frac{d-n-1}{d}}.$$

We expect beneficial overfitting for σ_1 and no beneficial overfitting for σ_3. For σ_2 the minimum in both regimes should be the same. Below in Figure 2.5 we plot the test risk averaged over $M = 100$ Monte Carlo iterations, with $d = 100$ and $n = 40$.

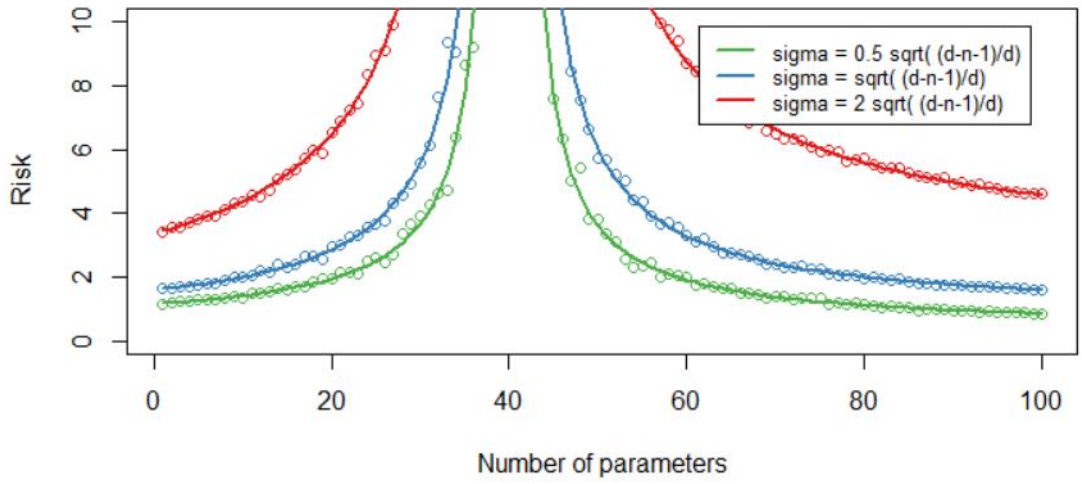

Figure 2.5: Test risk for different values of σ. Dots are the experimental values, lines are the theoretical values.

In Figure 2.5 we see that indeed for σ_1 we have beneficial Double Descent. For σ_3 the optimal test risk is in the under-parameterized regime. For σ_2 the minimum of both regimes is the same. Furthermore, experimental results agree very well with the theoretical test risk.

2.6 Infinite dimensional case

Until now, we have considered the finite dimensional linear regression case from equation (1.1), in which the results depend on the dimension d. The choice of d is somewhat arbitrary, as in practice there could be many more features that we did not take into account. Therefore, a more realistic scenario would be to consider the infinite dimensional case, where $d = \infty$. Notice that we cannot directly take the limit $d \to \infty$ of the results from Theorem 1. Indeed, if we consider the expression for R_{rand} in Theorem 1, then we have that $\mathbb{P}(i \in P) = \frac{p}{d}$ for the i-th feature. Now if we just let $d \to \infty$, then we would not select any features at all and we would have $\hat{\theta} = 0$. So it is clear that we need a different approach for this infinite dimensional setting and also a different choice of the distribution for selecting the features. The linear regression problem in this infinite dimensional case is

$$y = \langle x, \theta \rangle_{\mathcal{H}} + \varepsilon$$

with $y, \varepsilon \in \mathbb{R}$ and $x, \theta \in \mathcal{H}$. Here $\mathcal{H}$ is an infinite-dimensional Hilbert space with corresponding inner product and norm induced by this inner product, defined as

$$\langle x, \theta \rangle_{\mathcal{H}} := \sum_{j=1}^{\infty} x_i \theta_i, \qquad ||x||_{\mathcal{H}} := \sqrt{\langle x, x \rangle_{\mathcal{H}}} = \sqrt{\sum_{j=1}^{\infty} x_i^2}.$$

For the infinite-dimensional regression problem to be well-defined, we need $\langle x, \theta \rangle_{\mathcal{H}} < \infty$ with probability 1. Therefore, we impose the following conditions on x and θ,

$$\mathbb{P}(||x||_{\mathcal{H}} < \infty) = 1 \quad \text{and} \quad ||\theta||_{\mathcal{H}} < \infty.$$

Then by Cauchy-Schwarz $\langle x, \theta \rangle_{\mathcal{H}} \leq ||x||_{\mathcal{H}} ||\theta||_{\mathcal{H}} < \infty$. Notice that if $\mathbb{E}||x||_{\mathcal{H}}^2 = \text{tr}(\Sigma) < \infty$, then the condition for x is satisfied. We will tackle the infinite dimensional case in 2 different ways.

1. By constructing our own probability distribution on (subsets of) the natural numbers. As discussed before, the Uniform distribution cannot be applied in the infinite dimensional case and so we construct our own distribution for selecting the features. This will depend on some very specific assumptions and will only serve as an illustrative example of Double Descent in infinite dimensions.

2. By considering the more general kernel regression, which can deal with infinite (feature space) dimensions. This setting will be discussed in Chapter 5, where we will look at both approximating an infinite dimensional problem with a finite linear combination of basis functions and the other way around, where we approximate a finite dimensional problem with an infinite dimensional kernel.

2.6.1 Constructing our own distribution

In the infinite dimensional case, we cannot use the Uniform distribution for feature selection, as we need to select at random p numbers from a set with an infinite number of elements, $\mathbb{N}$. So we define our own probability distribution on $\mathbb{N}$. Note that we will be making some very specific assumptions about the choice of this distribution and the choice of parameter vector θ^*, but we are only interested in a first indication of what Double Descent in infinite dimensions could look like. Let $P \subset \mathbb{N}$ be of cardinality p. Let us define a probability measure μ_P as follows

$$P \mapsto \mu_P \in \mathcal{P}(\{0,1\}^{\mathbb{N}})$$

where $\mathcal{P}(\{0,1\}^{\mathbb{N}})$ denotes the space of probability measures on $\{0,1\}^{\mathbb{N}}$. Here $\{0,1\}^{\mathbb{N}}$ is the set of all subsets of $\mathbb{N}$, meaning that it contains functions $f : \mathbb{N} \to \{0,1\}$ such that every subset S of $\mathbb{N}$ can be constructed, as for example we can map every integer $n \in \mathbb{N}$ to either 0, meaning that it is excluded from S, or 1, meaning that it is included in S. For $j \in \mathbb{N}$, define

$$\mu_P(j) = \mathbb{P}(j \in P).$$

We want the expected cardinality of P to be equal to p, so μ_P should satisfy

$$p = \mathbb{E}|P| = \mathbb{E}\sum_{j\in\mathbb{N}} \mathbb{1}\{j \in P\} = \sum_{j\in\mathbb{N}} \mathbb{E}\mathbb{1}\{j \in P\} = \sum_{j\in\mathbb{N}} \mathbb{P}(j \in P).$$

Furthermore, $\mathbb{P}(j \in P)$ should be increasing in p and should be bounded by 1, hence we require

$$\lim_{p\to\infty} \mathbb{P}(j \in P) \leq 1.$$

The condition $\mathbb{P}(j \in P) \geq 0$ is already satisfied, since $\sum_{j\in\mathbb{N}} \mathbb{P}(j \in P) = 0$ if and only if $\mathbb{P}(j \in P) = 0$ for all $j \in \mathbb{N}$. Based on these described conditions, we will try the following choice of $\mathbb{P}(j \in P)$ and check whether beneficial overfitting occurs,

$$\mathbb{P}(j \in P) = c(p)(a^{-1/p})^j$$

with $c(p)$ and $a > 1$ to be determined. Then

$$p = \sum_{j=1}^{\infty} \mathbb{P}(j \in P) = c(p)\sum_{j=1}^{\infty}(a^{-1/p})^j = c(p)\left(\frac{1}{1-a^{-1/p}} - 1\right) = c(p)\frac{1}{a^{1/p}-1}.$$

So $c(p) = p(a^{1/p} - 1)$ and

$$\mathbb{P}(j \in P) = p(a^{1/p} - 1)(a^{-1/p})^j.$$

Next, we check the limit

$$\lim_{p\to\infty} \mathbb{P}(j \in P) = \lim_{p\to\infty} p(a^{1/p} - 1)(a^{-1/p})^j = \lim_{p\to\infty} \frac{a^{(-j+1)/p} - a^{-j/p}}{1/p} = \lim_{x\to 0} \frac{a^{(-j+1)x} - a^{-jx}}{x}.$$

Applying L'Hopital gives

$$\lim_{p\to\infty} \mathbb{P}(j \in P) = \lim_{x\to 0} \left(a^{(-j+1)x}\ln(a)(-j+1) - a^{-jx}\ln(a)(-j)\right)$$

$$= (-j+1)\ln(a) + j\ln(a) = \ln(a) \leq 1,$$

if we choose $a \in (1, e]$. For simplicity, we take $a = e$. Hence, we randomly select feature x_j with probability

$$\mathbb{P}(j \in P) = p(e^{1/p} - 1)(e^{-1/p})^j.$$

Notice that this means that we select small natural numbers with higher probability than large natural numbers. In order to find an expression for the test risk that we can analytically compute, we make a convenient choice for the true parameter θ. We take θ exponentially decreasing, $\theta_j = e^{-\alpha j}$, with $\alpha > 0$. Then

$$||\theta||^2 = \sum_{j=1}^{\infty} \theta_j^2 = \sum_{j=1}^{\infty} e^{-2\alpha j} = \sum_{j=1}^{\infty}(e^{-2\alpha})^j = \frac{1}{e^{2\alpha}-1} < \infty,$$

since it is a geometric series and $e^{-2\alpha} \leq 1$. Furthermore

$$\mathbb{E}||\theta_P||^2 = \sum_{j=1}^{\infty} \theta_j^2 \mathbb{P}(j \in P) = p(e^{1/p} - 1)\sum_{j=1}^{\infty}(e^{-2\alpha})^j(e^{-1/p})^j$$

$$= p(e^{1/p} - 1)\sum_{j=1}^{\infty}(e^{-(2\alpha+1/p)})^j = p(e^{1/p} - 1)\frac{1}{e^{2\alpha+1/p}-1} = \frac{p(e^{1/p}-1)}{e^{2\alpha+1/p}-1},$$

and

$$\mathbb{E}||\theta_{P^c}||^2 = ||\theta||^2 - \mathbb{E}||\theta_P||^2 = \frac{1}{e^{2\alpha}-1} - \frac{p(e^{1/p}-1)}{e^{2\alpha+1/p}-1}.$$

We can apply formula (2.1) from Theorem 1, as this result does not depend on the dimension d. Taking expectation of this result with respect to P, we then find

$$R_{rand}(\hat{\theta}) = \begin{cases} \left(\frac{1}{e^{2\alpha}-1} - \frac{p(e^{1/p}-1)}{e^{2\alpha+1/p}-1} + \sigma^2\right)(1+\frac{p}{n-p-1}) & \text{if } p \leq n-2 \\ \frac{p(e^{1/p}-1)}{e^{2\alpha+1/p}-1}(1-\frac{n}{p}) + (\frac{1}{e^{2\alpha}-1} - \frac{p(e^{1/p}-1)}{e^{2\alpha+1/p}-1} + \sigma^2)(1+\frac{n}{p-n-1}) & \text{if } p \geq n+2 \end{cases}$$

and $R_{rand}(\hat{\theta}) = \infty$ otherwise. This is an explicit formula and we can plot this. We take $\sigma = 0.1$ and $\alpha = 1$. We then find the plot in Figure 2.6 below.

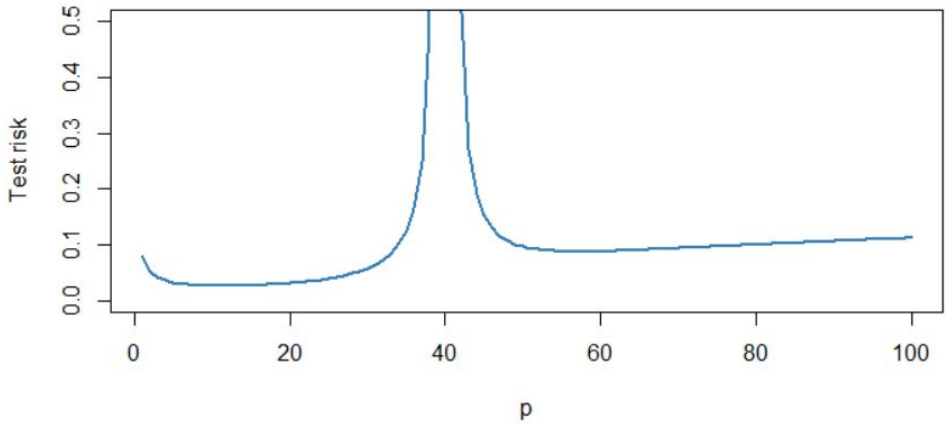

Figure 2.6: Theoretical test risk in the infinite dimensional case for $\mu_P(j) \propto e^{-j}$, exponentially decaying θ and $\sigma = 0.1$

In this case, we see Double Descent behaviour, but no beneficial overfitting: the minimum test risk still lies in the under-parameterized regime. Note that this is a somewhat arbitrary way to tackle the infinite dimensional case, as it depends on a very specific choice of distribution for selecting the features and a specific choice of the true parameter vector θ. A more constructive way to approach the infinite dimensional case is by looking at kernel regression, which we will consider in Chapter 5. However, our artificial example shows that it is possible to observe Double Descent also in the infinite-dimensional setting.

2.7 When do we expect beneficial overfitting?

We conclude this chapter with a short overview of when beneficial overfitting is likely to occur. Note that these conclusions are based on the isotropic Gaussian case and thus may not hold more generally.

We expect beneficial overfitting if:

- Eigenvalues of the covariance matrix are all equal to 1, features are selected at random and the SNR is larger than $\frac{d}{d-n-1}$.
- Eigenvalues of the covariance matrix are all equal to 1 and least important features (with the lowest weights) are selected.

We do not expect beneficial overfitting if:

- Eigenvalues of the covariance matrix are all equal to 1 and no feature selection.
- Eigenvalues of the covariance matrix are all equal to 1, features are selected at random and SNR is smaller than $\frac{d}{d-n-1}$.
- Eigenvalues of the covariance matrix are all equal to 1 and most important features (with the largest weights) are selected.

3 General Gaussian covariates

In this chapter we look at the case where we violate the isotropic Gaussian assumption from the previous chapter, meaning that $x \sim N(0, \Sigma)$ with $\Sigma \neq I_d$. We still assume Σ is a diagonal matrix, but now with general positive entries on the diagonal,

$$\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2).$$

This is one of the settings described in Hastie et al. (2020) [6]. However, their result is rather complicated compared to the result from the previous chapter, as it involves integrals with respect to the empirical distribution of the eigenvalues. Hence, we first try to re-use the result of Theorem 1 by transforming the parameter vector. When this transformation fails for the over-parameterized case, we will try to re-use the proof of Theorem 1 from the previous chapter. When this also breaks down, it seems that using the more complicated integrals in the result of Hastie et al. (2020) [6] for this general case is inevitable. We treat the under- and over-parameterized regime separately, starting with the under-parameterized regime.

3.1 Under-parameterized regime

In the under-parameterized regime, we would like to express the linear regression problem for x in terms of $z \sim N(0, I_d)$, since Theorem 1 only holds for isotropic Gaussian random variables. So we are looking for a matrix A such that $x = Az$. Since $z \sim N(0, I_d)$, we have

$$\mathbb{E}(Az) = 0, \qquad \text{Cov}(Az) = AA^T = \Sigma.$$

So for $x \sim N(0, \Sigma)$, we have to take $A = \Sigma^{1/2}$. Now the original linear regression problem for $z \sim N(0, I_d)$ is

$$y = z^T\beta + \varepsilon,$$

with β the unknown parameter and $\varepsilon \sim N(0, \sigma^2)$. The linear regression problem for $x = Az \sim N(0, \Sigma)$ is then

$$y = x^T\theta + \varepsilon = (Az)^T\theta + \varepsilon = z^T(A\theta) + \varepsilon.$$

So we have to choose the transformation $\beta = A\theta = \Sigma^{1/2}\theta$. To be in the same setting as in the previous chapter, we select p features by specifying a set $P \subset \{1, \ldots, d\}$ with $|P| = p$. Consider the min-norm solution $\hat{\beta}$ for which $\hat{\beta}_P = Z_P^\dagger y$ and $\hat{\beta}_{P^c} = 0$, where Z is the design matrix with rows z_i^T, $i \in \{1, \ldots, n\}$. Applying Theorem 1 to the linear regression problem for z, we find

$$R_{det}(\hat{\beta}) = \mathbb{E}(y - z^T\hat{\beta})^2 = (||\beta_{P^c}||^2 + \sigma^2)\left(1 + \frac{p}{n-p-1}\right),$$

$$R_{rand}(\hat{\beta}) = ((1 - \frac{p}{d})||\beta^*||^2 + \sigma^2)\left(1 + \frac{p}{n-p-1}\right),$$

for $p \leq n-2$. We can relate this to the test risk for $\hat{\theta}$ as follows

$$\mathbb{E}(y - z^T\hat{\beta})^2 = \mathbb{E}(y - (\Sigma^{-1/2}x)^T\Sigma^{1/2}\hat{\theta})^2 = \mathbb{E}(y - x^T\Sigma^{-1/2}\Sigma^{1/2}\hat{\theta})^2 = \mathbb{E}(y - x^T\hat{\theta})^2.$$

So in fact the test risk for $\hat{\beta}$ is equal to the test risk for $\hat{\theta}$. Furthermore,

$$||\beta_{P^c}||^2 = ||\Sigma_{P^c}^{1/2}\theta_{P^c}||^2, \qquad ||\beta_P||^2 = ||\Sigma_P^{1/2}\theta_P||^2.$$

Now we can state a generalized version of Theorem 1 for the case $p \leq n$.

Theorem 2 (Generalized version of Theorem 1 for $p \leq n$)**.** *Let $x \sim N(0, \Sigma)$, $\varepsilon \sim N(0, \sigma^2)$ and $y = x^T\theta + \varepsilon$. Choose $p \in \{0, \ldots, d\}$ and $P \subset \{1, \ldots, d\}$ such that $|P| = p$. Consider the min-norm solution $\hat{\theta}_P = X_P^\dagger y$ and $\hat{\theta}_{P^c} = 0$. Then, for $p \leq n-2$, the test risk of $\hat{\theta}$ is*

$$R_{det}(\hat{\theta}) = \mathbb{E}(y - x^T\hat{\theta})^2 = (||\Sigma_{P^c}^{1/2}\theta_{P^c}||^2 + \sigma^2)\left(1 + \frac{p}{n-p-1}\right),$$

and $R_{det}(\hat{\theta}) = \infty$ for $p = n-1, n$. If we take P a uniformly random subset of $\{1, \ldots, d\}$ with $|P| = p$, then

$$R_{rand}(\hat{\theta}) = ((1 - \frac{p}{d})||\Sigma^{1/2}\theta^*||^2 + \sigma^2)\left(1 + \frac{p}{n-p-1}\right),$$

and $R_{rand}(\hat{\theta}) = \infty$ for $p = n-1, n$.

3.2 Over-parameterized regime

In the over-parameterized regime, we cannot use the transformation $\beta = \Sigma^{1/2}\theta$, because of identifiability issues. To see this, we first look at how the design matrix for x, denoted by X, and the design matrix for z, denoted by Z, are related. We have

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} (\Sigma^{1/2}z_1)^T \\ (\Sigma^{1/2}z_2)^T \\ \vdots \\ (\Sigma^{1/2}z_n)^T \end{pmatrix} = \begin{pmatrix} z_1^T\Sigma^{1/2} \\ z_2^T\Sigma^{1/2} \\ \vdots \\ z_n^T\Sigma^{1/2} \end{pmatrix} = Z\Sigma^{1/2}.$$

Now, assuming $\beta = \Sigma^{1/2}\theta$, is it true that also $\hat{\beta} = \Sigma^{1/2}\hat{\theta}$? Recall that $\hat{\beta} = Z^\dagger y$ and $\hat{\theta} = X^\dagger y$. Then

$$\Sigma^{1/2}\hat{\theta} = \Sigma^{1/2}X^\dagger y = \Sigma^{1/2}(Z\Sigma^{1/2})^\dagger y.$$

If it is true that $(Z\Sigma^{1/2})^\dagger = (\Sigma^{1/2})^\dagger Z^\dagger$, then we have

$$\Sigma^{1/2}\hat{\theta} = \Sigma^{1/2}(\Sigma^{1/2})^\dagger Z^\dagger y = \Sigma^{1/2}(\Sigma^{1/2})^{-1}\hat{\beta} = \hat{\beta}.$$

However, the property $(AB)^\dagger = B^\dagger A^\dagger$ only holds if A has full column-rank and B has full row-rank. In this case $A = Z$ and $B = \Sigma^{1/2}$. Clearly, $\Sigma^{1/2}$ has full row-rank, since Σ is diagonal with non-zero diagonal entries. On the other hand, the matrix $Z \in \mathbb{R}^{n\times p}$ has full row rank, as by assumption $\text{rank}(Z) = \min(n, p) = n$, but not full column rank (since $n < p$). So in the over-parameterized regime, we run into identifiability issues: after transformation we cannot transform back to find $\hat{\theta}$, since transforming the parameter vector does not mean we can also transform the minimum-norm solutions.

Now that it is clear that simply transforming the problem does not work, we will try to use the proof of Belkin et al. (2020) [4], also stated in Appendix 9.1. Recall that $x \sim N(0, \Sigma)$ with

$$\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2) \in \mathbb{R}^{d\times d}.$$

First, we rewrite the test risk,

$$\mathbb{E}(y - x^T\hat{\theta})^2 = \mathbb{E}(x^T(\theta - \hat{\theta}) + \varepsilon)^2 = \mathbb{E}(x^T(\theta - \hat{\theta})^2) + 2\mathbb{E}(x^T(\theta - \hat{\theta})\varepsilon) + \mathbb{E}(\varepsilon^2)$$
$$= \sigma^2 + \mathbb{E}((\theta - \hat{\theta})^T xx^T(\theta - \hat{\theta})) = \sigma^2 + (\theta - \hat{\theta})^T\Sigma(\theta - \hat{\theta}) = \sigma^2 + ||\theta - \hat{\theta}||_\Sigma^2.$$

This is similar to the isotropic Gaussian case, but now we take the norm $||.||_\Sigma$ with respect to the matrix Σ. We want to express this in terms of $||\theta_P - \hat{\theta}_P||_{\Sigma_P}^2$ and $||\theta_{P^c} - \hat{\theta}_{P^c}||_{\Sigma_{P^c}}^2$. Write

$$\Sigma = \begin{pmatrix} \Sigma_P & O_{p\times(d-p)} \\ O_{(d-p)\times p} & \Sigma_{P^c} \end{pmatrix} \quad \text{and} \quad \theta = \begin{pmatrix} \theta_P \\ \theta_{P^c} \end{pmatrix}.$$

For notational convenience, let $\alpha = \theta - \hat{\theta}$. Then

$$(\theta - \hat{\theta})^T \Sigma (\theta - \hat{\theta}) = \alpha^T \Sigma \alpha = (\alpha_P^T, \alpha_{P^c}^T) \begin{pmatrix} \Sigma_P & O_{p\times(d-p)} \\ O_{(d-p)\times p} & \Sigma_{P^c} \end{pmatrix} \begin{pmatrix} \alpha_P \\ \alpha_{P^c} \end{pmatrix}$$

$$= (\alpha_P^T \Sigma_P + \alpha_{P^c}^T O_{(d-p)\times p}, \alpha_P^T O_{p\times(d-p)} + \alpha_{P^c}^T \Sigma_{P^c}) \begin{pmatrix} \alpha_P \\ \alpha_{P^c} \end{pmatrix}$$

$$= (\alpha_P^T \Sigma_P + \alpha_{P^c} O_{(d-p)\times p}) \alpha_P + (\alpha_{P^c}^T O_{p\times(d-p)} + \alpha_{P^c}^T \Sigma_{P^c}) \alpha_{P^c}$$

$$= \alpha_P^T \Sigma_P \alpha_P + \alpha_{P^c}^T \Sigma_{P^c} \alpha_{P^c} = ||\alpha_P||_{\Sigma_P}^2 + ||\alpha_{P^c}||_{\Sigma_{P^c}}^2$$

$$= ||\theta_P - \hat{\theta}_{P^c}||_{\Sigma_P}^2 + ||\theta_{P^c} - \hat{\theta}_{P^c}||_{\Sigma_{P^c}}^2.$$

Since $\hat{\theta}_{P^c} = 0$, we find

$$R(\hat{\theta}) = \sigma^2 + \mathbb{E}||\theta_P - \hat{\theta}_P||_{\Sigma_P}^2 + ||\theta_{P^c}||_{\Sigma_{P^c}}^2.$$

We consider the over-parameterized regime ($p \geq n$). Then the pseudo-inverse of X_P is given by $X_P^\dagger = X_P^T (X_P X_P^T)^{-1}$. Let $\eta := y - X_P \theta_P \in \mathbb{R}^n$. We have

$$\theta_P - \hat{\theta}_P = \theta_P - X_P^T (X_P X_P^T)^{-1} y = \theta_P - X_P^T (X_P X_P^T)^{-1} (X_P \theta_P + \eta)$$

$$= (I - X_P^T (X_P X_P^T)^{-1} X_P) \theta_P - X_P^T (X_P X_P^T)^{-1} \eta.$$

Notice that $(I - X_P^T (X_P X_P^T)^{-1} X_P)\theta_P$ is in the null space of X_P and $-X_P^T (X_P X_P^T)^{-1}\eta$ is in the row space of X_P. Since $\text{Row}(X_P) \perp \text{Null}(X_p)$, we have by the Pythagorean theorem,

$$||\theta_P - \hat{\theta}_P||^2 = ||(I - X_P^T (X_P X_P^T)^{-1} X_P)\theta_P||^2 + ||X_P^T (X_P X_P^T)^{-1} \eta||^2.$$

However, this does not apply to $||.||_\Sigma$. In that case we get an additional term,

$$||\theta_P - \hat{\theta}_P||_{\Sigma_P}^2 = ||(I - X_P^T (X_P X_P^T)^{-1} X_P)\theta_P||_{\Sigma_P}^2 + ||X_P^T (X_P X_P^T)^{-1} \eta||_{\Sigma_P}^2$$

$$-2\langle\ (I - X_P^T (X_P X_P^T)^{-1} X_P)\theta_P\ ,\ X_P^T (X_P X_P^T)^{-1} \eta\ \rangle_{\Sigma_P}.$$

Unfortunately, the arguments from the proof in Appendix 9.1 break down after this point. In the proof we used the rotational symmetry of the multivariate standard normal distribution. However, for the more general covariance matrix Σ, the rotational symmetry is lost, as the different directions of the covariates x are weighted with the respective variance in that direction. Notice that the second part of the proof, where we use the inverse Wishart distribution, can be re-used, since we can use a more general scale matrix Σ.

Now that transforming the parameter vector and re-using the proof of Theorem 1 have not helped us, we resort to the result from Hastie et al. (2020) [6]. They consider the more general case, where $x \sim P_x$ for some distribution P_x, in particular we assume $x \sim N(0, \Sigma)$, with Σ any symmetric and positive definite covariance matrix, in the asymptotic regime where $\frac{p}{n} \to \gamma$ as $n, p \to \infty$. Their result is more complicated than the analytical result from Theorem 1, as it depends on the empirical distribution $\hat{H}_n(s)$ of the eigenvalues of Σ, and the reweighted version $\hat{G}_n(s)$, which are defined as

$$\hat{H}_n(s) = \frac{1}{p} \sum_{i=1}^{p} \mathbb{1}\{s \geq \lambda_i\}, \qquad \hat{G}_n(s) = \frac{1}{||\theta||^2} \sum_{i=1}^{p} (\theta^T v_i)^2 \mathbb{1}\{s \geq \lambda_i\}$$

with $\{\lambda_1, \ldots, \lambda_p\}$ the eigenvalues of $\Sigma \in \mathbb{R}^{p\times p}$, sorted in decreasing order, and $\{v_1, \ldots, v_p\}$ the eigenvectors of Σ. We will state the result of [6] in the following theorem and consider the special case $\Sigma = I$. We will not use this result any further, as it is difficult to derive from this result the particular conditions that could lead to beneficial overfitting.

Theorem 3 (Theorem 2 from [6]). *Assume the covariates satisfy $x = \Sigma^{1/2} z$, with z having independent entries with mean 0, variance 1 and finite moments. Let $\gamma = \lim_{d\to\infty} \frac{p(d)}{n(d)}$. Assume that as $d \to \infty$, also $p(d) \to \infty$ and $n(d) \to \infty$. Assume there exists a constant $M > 0$ such that*

$$\lambda_1 = ||\Sigma|| \leq M, \qquad \int \frac{1}{s} d\hat{H}_n(s) < M, \qquad |1-\gamma| \geq \frac{1}{M}, \qquad \frac{1}{M} \leq \gamma \leq M$$

Let $c_0(\hat{H}_n, \gamma)$ be the solution to

$$1 - \frac{1}{\gamma} = \int \frac{1}{1 + c_0 \gamma s} d\hat{H}_n(s).$$

Assume $\hat{H}_n$ and $\hat{G}_n$ converge in distribution to H resp. G as $n \to \infty$. Then for $d \to \infty$, almost surely

$$R(\hat{\theta}) = B(\hat{\theta}) + V(\hat{\theta}) \longrightarrow \mathcal{B}(H, G, \gamma) + \mathcal{V}(H, \gamma),$$

with

$$\mathcal{B}(\hat{H}_n, \hat{G}_n, \gamma) = ||\theta||^2 \left(1 + \gamma c_0 \frac{\int \frac{s^2}{(1+c_0\gamma s)^2} d\hat{H}_n(s)}{\int \frac{s}{(1+c_0\gamma s)^2} d\hat{H}_n(s)}\right) \int \frac{s}{(1+c_0\gamma s)^2} d\hat{G}_n(s),$$

$$\mathcal{V}(\hat{H}_n, \gamma) = \sigma^2 \gamma c_0 \frac{\int \frac{s^2}{(1+c_0\gamma s)^2} d\hat{H}_n(s)}{\int \frac{s}{(1+c_0\gamma s)^2} d\hat{H}_n(s)}.$$

Proof. See the proof in [6].

Note: in [6] the factor c_0 seems to be forgotten in the expression for $\mathcal{V}(\hat{H}_n, \gamma)$. The expressions for Theorem 3 are not very insightful, but we can greatly simplify the expressions if we consider the case $\Sigma = I$. Then $\lambda_i = 1$ and $v_i = e_i$, so we have

$$\hat{H}_n(s) = \frac{1}{p} \sum_{i=1}^{p} \mathbb{1}\{\lambda_i \leq s\} = \mathbb{1}\{1 \leq s\},$$

$$\hat{G}_n(s) = \frac{1}{||\theta||^2} \sum_{i=1}^{p} (\theta^T v_i)^2 \mathbb{1}\{\lambda_i \leq s\} = \frac{1}{||\theta||^2} \sum_{i=1}^{p} \theta_i^2 \mathbb{1}\{1 \leq s\} = \frac{\mathbb{1}\{1 \leq s\} \sum_{i=1}^{p} \theta_i^2}{||\theta||^2} = \mathbb{1}\{1 \leq s\}.$$

Let $\Delta(s_i)$ be the increment at $s = s_i$. By definition of the integral, we then have

$$\int \frac{1}{1 + c_0 \gamma s} d\hat{H}_n(s) = \sum_i \frac{1}{1 + c_0 \gamma s_i} \Delta(s_i) = \frac{1}{1 + c_0 \gamma}.$$

Now c_0 solves

$$1 - \frac{1}{\gamma} = \frac{1}{1 + c_0 \gamma} \qquad , \quad c_0 = \frac{1}{\gamma(\gamma - 1)}.$$

Furthermore

$$\int \frac{s^2}{(1 + c_0 \gamma s)^2} d\hat{H}_n(s) = \frac{1}{(1 + c_0 \gamma)^2},$$

$$\int \frac{s}{(1 + c_0 \gamma s)^2} d\hat{H}_n(s) = \int \frac{s}{(1 + c_0 \gamma s)^2} d\hat{G}_n(s) = \frac{1}{(1 + c_0 \gamma)^2}.$$

The expression for the bias term $B(\hat{\theta})$ now becomes, as $d \to \infty$

$$B(\hat{\theta}) \to ||\theta||^2 (1 + \gamma c_0) \frac{1}{(1 + c_0 \gamma)^2} = ||\theta||^2 \frac{1}{1 + \gamma c_0} = ||\theta||^2 \frac{1}{1 + \frac{\gamma}{\gamma(\gamma-1)}} = ||\theta||^2 (1 - \frac{1}{\gamma}).$$

For the variance term $V(\hat{\theta})$ we find, as $d \to \infty$

$$V(\hat{\theta}) \to \sigma^2 \gamma \frac{1}{\gamma(\gamma - 1)} = \sigma^2 \frac{1}{\gamma - 1}.$$

These are the same expressions as mentioned in [6] and also what we found in Section 2.4.2.

4 Sub-Gaussian covariates

In this chapter we again look at the linear regression problem in (1.1), but now with sub-Gaussian covariates. This setting is discussed in Bartlett et al. (2020) [5]. They derive an upper bound on the excess risk of the minimum norm solution and show that the occurrence of beneficial overfitting depends on the structure of the covariance matrix of the data. It turns out that over-parameterization only improves test performance when there are many low variance directions (i.e. many weak features), more than the number of data points. In this chapter we look at their main result, consider some special cases and verify the results experimentally.

4.1 Setting

Bartlett et al. (2020) [5] discuss the linear regression problem as in (1.1). They assume sub-Gaussian covariates: $x = \Sigma^{1/2} z$ with $z \sim \text{subG}(\sigma_x^2)$, which means that for all $a \in \mathbb{R}^p$

$$\mathbb{E}(e^{a^T z}) \leq \exp\left(\frac{\sigma_x^2 ||a||^2}{2}\right).$$

We consider the over-parameterized regime where $p > n$. We generate n training data points $(x_i, y_i)_{i=1}^n$, where $x_i \sim P_x$ such that $\mathbb{E}(xx^T) = \Sigma$ and y_i generated according to equation (1.1). Consider the eigendecomposition of the covariance matrix of the data

$$\Sigma = \mathbb{E}(x_i x_i^T) = \sum_{i=1}^{p} \lambda_i v_i v_i^T,$$

with λ_i the eigenvalues of $\Sigma \in \mathbb{R}^{p \times p}$ in decreasing order ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$) and v_i the (orthonormal) eigenvectors of Σ. We assume $\text{rank}(\Sigma) > n$. Notice that we want the rank to be strictly larger than n, as we will need this in the proof in Appendix 9.3 when we remove one of the eigenvectors of Σ and still want $\text{rank}(\Sigma)$ to be at least equal to n.

Before we state the main result of [5], we first define the notion of effective rank, which will play an important role in the main result and its proof. Assume $\text{rank}(\Sigma) = p > n$ and let $k \in \{0, 1, \ldots, p-1\}$. We need the following two notions of effective rank

$$r_k(\Sigma) := \frac{\Sigma_{i=k+1}^p \lambda_i}{\lambda_{k+1}}, \qquad R_k(\Sigma) := \frac{(\Sigma_{i=k+1}^p \lambda_i)^2}{\sum_{i=k+1}^p \lambda_i^2}.$$

We can view $r_k(\Sigma)$ as the effective rank of the covariance after the k heaviest directions are dropped. The following lemma describes how the effective ranks are related to the usual rank of the matrix Σ.

Lemma 1 (Lemma 5 in [5]). *If rank$(\Sigma) = p$, then*

$$r_0(\Sigma) = \textit{rank}(\Sigma) s(\Sigma), \qquad R_0(\Sigma) = \textit{rank}(\Sigma) S(\Sigma)$$

with

$$s(\Sigma) = \frac{\frac{1}{p}\sum_{i=1}^p \lambda_i}{\lambda_1}, \qquad S(\Sigma) = \frac{(\frac{1}{p}\sum_{i=1}^p \lambda_i)^2}{\frac{1}{p}\sum_{i=1}^p \lambda_i^2}$$

where s, S lie between $\frac{1}{p}$ (when $\lambda_2 \approx 0$) and 1 (when all λ_i are equal). Furthermore

$$1 \leq r_k(\Sigma) \leq R_k(\Sigma) \leq p.$$

Proof. See Appendix 9.2.

4.2 Main Result

Using the previously defined notions of effective rank, we can state the main theorem, which is Theorem 4 from Bartlett et al. (2020) [5]. This theorem gives upper and lower bounds for the excess risk of the minimum-norm interpolating estimator $\hat{\theta}$. We are mainly interested in the upper bound.

Theorem 4 (Upper bound from Theorem 4 in [5])**.** *Assume covariates $x = \Sigma^{1/2}z$, with $z \sim subG(\sigma_x^2)$ with mean 0 and unit variance. Assume rank$(\Sigma) = p > n$. There exist constants $b, c > 1$ such that for any $\delta \in [e^{-n/c}, 1]$ we have with probability at least $1 - \delta$*

$$R_{excess}(\hat{\theta}) \leq c||\theta^*||^2||\Sigma|| \max\left\{\sqrt{\frac{r_0(\Sigma)}{n}}, \frac{r_0(\Sigma)}{n}, \sqrt{\frac{\log(1/\delta)}{n}}\right\} + c\log(1/\delta)\sigma^2\left(\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}\right)$$

if $k^* \leq n$, where k^* is defined as

$$k^* := \min\{k \geq 0 : r_k(\Sigma) \geq bn\}$$

and $k^* = \infty$ if $\{k \geq 0 : r_k(\Sigma) \geq bn\} = \emptyset$.

Proof. See Appendix 9.3.

Notice that for $\delta = e^{-n/c}$ and $n \to \infty$, the upper bound in the theorem holds with probability $\lim_{n\to\infty} 1 - e^{-n/c} = 1$. The value k^* in the theorem we can view as the number of largest eigenvalues that we have to skip before the effective rank gets as large as (a constant times) n. We can also write k^* as

$$k^* = \min\left\{k \geq 0 : \sum_{i=k+1}^{p} \lambda_i \geq b\lambda_{k+1}n\right\}.$$

From this representation we see that k^* is the minimum value of k for which, after removing the k largest eigenvalues, the tail of the eigenvalues ($\sum_{i=k+1}^{p} \lambda_i$) fits more than n times in the largest remaining eigenvalue (λ_{k+1}). So, if k^* is large, then we have to remove many large eigenvalues before the tail of the eigenvalues is large enough compared to the remaining largest eigenvalue. If k^* is small, then we only have to remove a few large eigenvalues before the tail of the eigenvalues is large enough compared to the remaining largest eigenvalue. Hence for k^* to be small, we want a small value of λ_{k+1} and a large value for $\sum_{i=k+1}^{p} \lambda_i$. This is the case when we have a long tail of relatively small eigenvalues. We will see that this is an important property if we want beneficial overfitting to occur.

4.3 Implications of Main Result

For beneficial overfitting, we want the test risk in the over-parameterized regime to be small, compared to the under-parameterized regime, and so we hope that also the upper bound from Theorem 4 becomes small. Set $p = p(n)$ and suppose that $p(n) \to \infty$ as $n \to \infty$ (which is the case in the over-parameterized regime as $p > n$). We want the expressions in front of $||\theta^*||$ and σ^2 in Theorem 4 to become small as $n \to \infty$. We thus require

$$\lim_{n\to\infty}\left(\frac{r_0(\Sigma)}{n} + \frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}\right) = 0. \tag{4.1}$$

The behaviour of this limit is fully determined by the behaviour of the eigenvalues of Σ. Hence, Bartlett et al. (2020) [5] call a covariance matrix Σ satisfying condition (4.1) (asymptotically) benign. Note that this condition is different from beneficial overfitting, but it in fact implies beneficial overfitting: if the excess test risk in the over-parameterized regime is zero, then it is always lower than (or equal to) the under-parameterized risk. Informally, to satisfy condition (4.1), we need:

- $r_0(\Sigma)$ has to be small compared to the sample size n. This is satisfied when the trace of Σ is small compared to n, meaning that the total sum of the eigenvalues cannot be too large.

- k^* has to be small compared to n. We saw in the previous section that for k^* to be small, we need a long tail of relatively small eigenvalues of the covariance matrix Σ.
- $R_{k^*}(\Sigma)$ has to be large compared to n. Since from Lemma 1 we know that $r_k(\Sigma) \leq R_k(\Sigma)$, this is satisfied when $r_{k^*}(\Sigma)$ is large compared to n. Since by definition $r_{k^*}(\Sigma) \geq bn$ with $b > 1$, we know that $R_{k^*}(\Sigma)$ is large compared to n.

From step 1 of the proof in Appendix 9.3, we know that we can write the excess risk as

$$R_{excess}(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \theta^*)^T \Sigma (\hat{\theta} - \theta^*)$$

where $\Sigma = V \Lambda V^T = \sum_{i=1}^p \lambda_i v_i v_i^T$. This expression of the excess risk shows how the error in direction i (which is $\hat{\theta}_i - \theta_i^*$) impacts the prediction accuracy. This error is weighted with the eigenvalue λ_i. So having many small eigenvalues, meaning that there are many low variance directions, has only a small impact on prediction accuracy. However, having small eigenvalues, and in particular a long, flat tail of eigenvalues, is crucial to satisfy condition (4.1) for beneficial overfitting. So it seems that the larger eigenvalues are important for the prediction accuracy and the small eigenvalues are important for beneficial overfitting.

Condition (4.1) agrees with the result of Section 2.4.1 for no feature selection. There we considered the usual test risk $R(\hat{\theta})$, for which we showed that the optimal test risk in the under-parameterized regime, denoted by R^*_{under}, is

$$R^*_{under} = R(p = 0) = \sigma^2.$$

Hence, if the test risk is non-increasing for $p > n$, and using that $p(n) \to \infty$ as $n \to \infty$, we need for beneficial overfitting that

$$\lim_{n \to \infty} R(\hat{\theta}) \leq \sigma^2.$$

We can relate this to the excess risk as follows

$$R_{excess}(\hat{\theta}) = \mathbb{E}_{x,y}(y - x^T \hat{\theta})^2 - \mathbb{E}_{x,y}(y - x^T \theta^*)^2 = R(\hat{\theta}) - \sigma^2.$$

So for beneficial overfitting the excess risk has to satisfy

$$\lim_{n \to \infty} R_{excess}(\hat{\theta}) = \lim_{n \to \infty} [R(\hat{\theta}) - \sigma^2] \leq \sigma^2 - \sigma^2 = 0.$$

This is similar to condition (4.1), where we require the upper bound on the excess risk to converge to 0. Note however that convergence of the excess risk does not necessarily mean that also the upper bound has to converge. Vice versa, we do have that convergence of the upper bound implies convergence of the excess risk. Hence, if condition (4.1) is satisfied, then we must also have beneficial overfitting.

4.4 Special case: identity covariance matrix

In this section we will consider the case where Σ is the identity matrix and check condition (4.1). If $\Sigma = I_p$, we can find simple expressions for $r_k(\Sigma)$ and $R_k(\Sigma)$. Since I_p has rank p, we have $\text{rank}(\Sigma) = p$. Furthermore, all λ_i are equal to 1. Therefore

$$r_k(I_p) = \frac{\sum_{i=k+1}^p \lambda_i}{\lambda_{k+1}} = p - k, \qquad R_k(I_p) = \frac{(\sum_{i=k+1}^p \lambda_i)^2}{\sum_{i=k+1}^p \lambda_i^2} = \frac{(p-k)^2}{p-k} = p - k.$$

We can now simplify the expressions in Theorem 4. The expression for k^* becomes

$$k^* = \min\{k \geq 0 : p - k \geq bn\}.$$

Since $p - k$ is decreasing in k, we have either $k^* = 0$ or $k^* = \infty$. Since we assume $k^* \leq n$, we take $k^* = 0$. Plugging in $r_0(\Sigma) = R_0(\Sigma) = p$ and $k^* = 0$ in the formula of Theorem 4, we find that with probability at least $1 - \delta$

$$R_{excess}(\hat{\theta}) \leq c||\theta^*||^2||I_p|| \max\left\{\sqrt{\frac{p}{n}}, \frac{p}{n}, \sqrt{\frac{\log(1/\delta)}{n}}\right\} + c\log(1/\delta)\sigma^2 \frac{n}{p}.$$

We can simplify this further, since $||I_p|| = \lambda_{max}(I_p) = 1$ and using that $\delta \geq e^{-n/c}$ and $c > 1$, we find

$$\sqrt{\frac{\log(1/\delta)}{n}} < \sqrt{1/c} < 1.$$

Since $p > n$, we have

$$\frac{p}{n} > \sqrt{\frac{p}{n}} > 1.$$

So the expression for the upper bound reduces to

$$R_{excess}(\hat{\theta}) \leq c||\theta^*||^2 \frac{p}{n} + c\sigma^2 \log(1/\delta)\frac{n}{p},$$

with probability at least $1 - \delta$. Let us now check condition (4.1). This condition reduces to

$$\lim_{n\to\infty} \left(\frac{p}{n} + \frac{n}{p}\right) = 0.$$

Clearly, this condition is not satisfied and hence we do not expect beneficial overfitting in the identity covariance case. This agrees with the result in Section 2.4.1, where we found that in case of no feature selection, beneficial overfitting is not possible. In the next section we will consider more general examples of eigenvalue sequences.

4.5 Examples of eigenvalue sequences

In this section we give 2 examples of eigenvalue sequences of Σ that correspond to a benign covariance matrix, and check under which assumptions they satisfy condition (4.1).

We start with considering slowly decaying or polynomially decaying eigenvalues: $\lambda_i = i^{-\alpha}$ for $\alpha > 0$. The case $\alpha = 0$ has been discussed in the previous section. This setting is shown in Theorem 31 in [5].

Lemma 2 (Theorem 31 in [5]). *If $\lambda_i = i^{-\alpha}$ with $\alpha > 0$, and dimension $p = p(n)$, then Σ is benign if and only if:*
(1)

$$\alpha \in (0, 1) \quad \textit{and} \quad \omega(n) = p(n) = o(n^{1/(1-\alpha)})$$

(2)

$$\alpha = 1 \quad \textit{and} \quad \omega(e^{\sqrt{n}}) = p(n) = o(e^n)$$

Proof. See Appendix I in [5]. The proof consists of checking for which values of α and $p = p(n)$ condition (4.1) is satisfied, while lower- and upper bounding the summations in the effective ranks by integrals.

Here the notation $f(n) = o(g(n))$ means that

$$\frac{f(n)}{g(n)} \to 0, \qquad \text{as } n \to \infty,$$

and $f(n) = \omega(g(n))$ means that

$$\frac{f(n)}{g(n)} \to \infty, \qquad \text{as } n \to \infty.$$

Lemma 2 shows that for slowly decaying eigenvalues, beneficial overfitting occurs if either (1) decay is slower than $\frac{1}{i}$ and $p(n)$ is approximately between n and $n^{1/(1-\alpha)}$, or (2) decay is exactly $\frac{1}{i}$ and $p(n)$ is approximately between $e^{\sqrt{n}}$ and e^n. Notice that for $\alpha = 1$ the Lemma imposes a condition on the number of parameters $p(n)$ which is computationally very expensive. For polynomial decay faster than $\frac{1}{i}$, no beneficial overfitting is possible.

The second eigenvalue sequence we consider are rapidly decaying or exponentially decaying eigenvalues: $\lambda_i = e^{-i} + \Delta_n$, with Δ_n a small perturbation term. This perturbation term does not change the decaying behaviour of the eigenvalues, as it is independent of the eigenvalue index i. However, this term is necessary for beneficial overfitting as it ensures that there is a long flat tail of small eigenvalues, with a value of around Δ_n. Without this term, the eigenvalues would decay to 0. This setting is also shown in Theorem 31 in [5], see the lemma below.

Lemma 3 (Theorem 31 in [5]). *If $\lambda_i = e^{-i} + \Delta_n$, with Δ_n a small perturbation and dimension $p = p(n)$, then Σ is benign if and only if:*

$$p(n) = \omega(n) \quad \textit{and} \quad \omega(ne^{-n}) = \Delta_n p(n) = o(n)$$

Proof. See Appendix I in [5]. The proof consists of checking for which values of Δ_n and $p = p(n)$ condition (4.1) is satisfied.

Lemma 3 shows that for rapidly decaying eigenvalues, beneficial overfitting occurs if $p \gg n$ and the perturbation Δ_n is small compared to the sample size n, but not exponentially small.

In general, we see that beneficial overfitting can occur for both slowly and rapidly decaying eigenvalues, as long as the small eigenvalues decay slowly, as prescribed by condition (4.1), i.e. when the sequence of eigenvalues has a long, flat tail. This is obvious for slowly decaying eigenvalues, and for rapidly decaying eigenvalues this is ensured by the perturbation term Δ_n. We thus require many low variance (unimportant) directions for beneficial overfitting, which is the case when we are in the highly over-parameterized regime.

4.6 Numerical experiments

In this section we discuss the experimental results in which we look at three different choices of the covariance matrix. These choices are based on the types of eigenvalue behaviour that may or may not result in beneficial overfitting, as discussed in the previous two sections, namely the identity eigenvalues, polynomially decaying and exponentially decaying eigenvalues. For all choices, we take Σ to be a diagonal matrix, such that the eigenvalues are equal to the diagonal entries of the matrix. We look at:

1. Identity covariance: $\Sigma = I_p$, so $\lambda_1 = \ldots = \lambda_p = 1$.
2. Slowly / polynomially decaying eigenvalues: Σ diagonal with eigenvalues $\lambda_i = i^{-\alpha}$ for $\alpha = \frac{1}{2}$, $\alpha = 1$ and $\alpha = 2$, which is discussed in Lemma 2.
3. Rapidly / exponentially decaying eigenvalues: Σ diagonal with $\lambda_i = e^{-i} + \Delta_n$, which is discussed in Lemma 3.

Furthermore, we will look at the influence of normalizing the true parameter vector and the influence of the label noise σ.

We perform a Monte Carlo simulation, with $M = 100$ Monte Carlo iterations, dimension $d = 100$ and p ranging from 1 to 100. Furthermore, we use $\sigma = 1$ and true parameters $\theta_j^* \propto 1$ such that $||\theta^*|| = 1$. We change the number of data points n depending on the situation. For the identity covariance case, we take $n = 40$, the same as in Chapter 2. For the polynomial and exponential decay of eigenvalues, we take n to be a small value, such that we can look at the situation where p is much larger than n. Notice that having such small n is not a realistic scenario, but the graphs will only be an illustrative example of the results of Lemma 2 and 3. We perform the following Monte Carlo scheme:

- Sample new training data $x_i \sim N(0, \Sigma)$ and construct design matrix $X \in \mathbb{R}^{n \times p}$ with rows x_i^T.
- Sample new label noise $\varepsilon \sim N(0, \sigma^2 I_n)$.

- Generate true labels $y = X\theta^* + \varepsilon$.
- Calculate least-squares/min-norm solution

$$(p \geq n): \quad \hat{\theta} = X^T(XX^T)^{-1}y, \qquad (p < n): \quad \hat{\theta} = (X^TX)^{-1}X^Ty.$$

- Compute excess training risk

$$R_{training} = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\hat{\theta})^2 - \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\theta^*)^2$$

- Compute excess test risk

$$R_{test} = (\hat{\theta} - \theta^*)^T\Sigma(\hat{\theta} - \theta^*)$$

We repeat above steps $M = 100$ times and take averages over the $R_{training}$ and R_{test} values.

4.6.1 Identity covariance matrix

The first setting we investigate is the case in which $\Sigma = I_p$. We take isotropic Gaussian covariates: $x \sim N(0, I_p)$. In this special case all λ_i are equal to 1 and $r_0(\Sigma) = R_0(\Sigma) = \text{rank}(\Sigma) = p$. We plot the excess training and test risk in Figure 4.1. We may not expect beneficial overfitting, as condition (4.1) is not satisfied. Indeed, we see that the test risk in the over-parameterized regime does not fall below the minimum test risk in the under-parameterized regime.

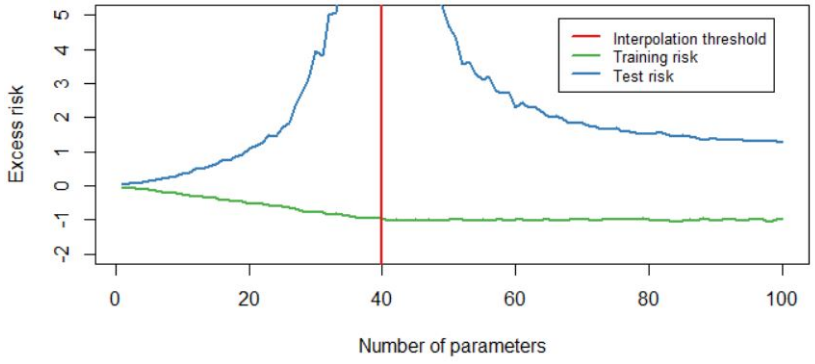

Figure 4.1: Excess training and test risk for the isotropic Gaussian case.

4.6.2 Polynomial decay of eigenvalues

The second case we will look at, is polynomial decay of the eigenvalues. This means that we choose $\lambda_i = i^{-\alpha}$ for some $\alpha > 0$. We know that for $\alpha \in (0, 1]$, beneficial overfitting occurs (for a suitably chosen p). Notice that the case $\alpha = 0$ results in the identity covariance case, which was shown in the previous subsection. We will investigate different values of α, namely: $\alpha = \frac{1}{2}$, $\alpha = 1$ and $\alpha = 2$. We expect to observe beneficial overfitting for $\alpha = \frac{1}{2}$ and $\alpha = 1$, but not for $\alpha = 2$. We take Σ to be a diagonal matrix with λ_i on the diagonal and covariates $x \sim N(0, \Sigma)$.

Case $\alpha = \frac{1}{2}$

We start with the case $\alpha = \frac{1}{2}$. Then Σ is a diagonal matrix with entries $\Sigma_{i,i} = \frac{1}{\sqrt{i}}$. According to Lemma 2 we expect beneficial overfitting if $p = p(n)$ satisfies

$$\omega(n) = p(n) = o(n^{1/(1-\alpha)}) = o(n^2).$$

We will plot the test risk averaged over $M = 100$ Monte Carlo iterations, against the number of parameters. We take $n = 4$. The results can be seen in Figure 4.2. Comparing the value of the excess

test risk at $p = 0$ and $p = 100 \gg 4$, we observe beneficial overfitting: the value at $p = 100$ is lower than the minimum value in the under-parameterized regime, as we would expect. Notice that p is larger than the upper bound of $n^2 = 4^2 = 16$, which does not contradict Lemma 4.2, as the bounds for $p(n)$ hold for large n.

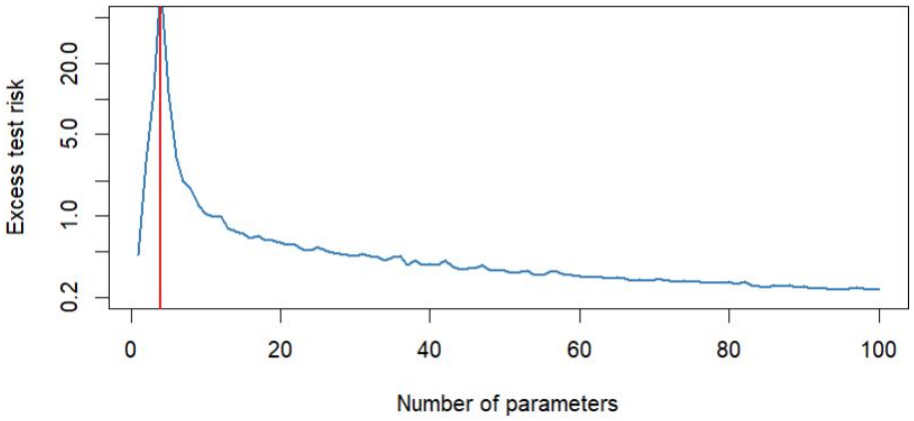

Figure 4.2: Logarithm of the excess test risk (blue line) and interpolation threshold (red line) for polynomial eigenvalue decay with $\alpha = \frac{1}{2}$.

Case $\alpha = 1$

We take Σ to be a diagonal matrix with diagonal entries $\Sigma_{i,i} = i^{-1}$. According to Lemma 2 we expect beneficial overfitting if $p = p(n)$ satisfies

$$\omega(e^{\sqrt{n}}) = p(n) = o(e^n)$$

In this case, we take $n = 4$, as $p = \omega(e^{\sqrt{n}})$ implies a heavy cost on the computational time. The result can be seen in Figure 4.3 below. Indeed, if we compare the value of the excess test risk for $p = 0$ and $p = 100 \gg e^{\sqrt{4}}$, then the value at $p = 100$ is lower than the minimum value in the under-parameterized case. In practice however, as n can be very large, the condition $p(n) = \omega(e^{\sqrt{n}})$ will impose a too heavy toll on the computational time and is therefore not feasible.

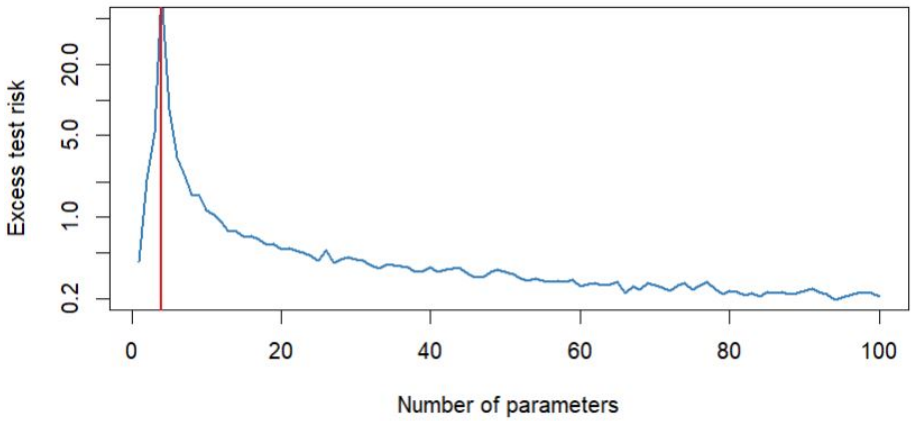

Figure 4.3: Logarithm of excess test risk (blue line) and interpolation threshold (red line) for polynomial eigenvalue decay with $\alpha = 1$.

Case $\alpha = 2$

We take Σ to be a diagonal matrix with diagonal entries $\lambda_i = i^{-2}$. According to Lemma 2 we do not expect beneficial overfitting in case $\alpha = 2$. Indeed, even if we take $n = 4$ such that p is much larger than n, we still do not observe beneficial overfitting, as seen in Figure 4.4: the value of the test risk at $p = 100$ is still larger than the value at $p = 0$.

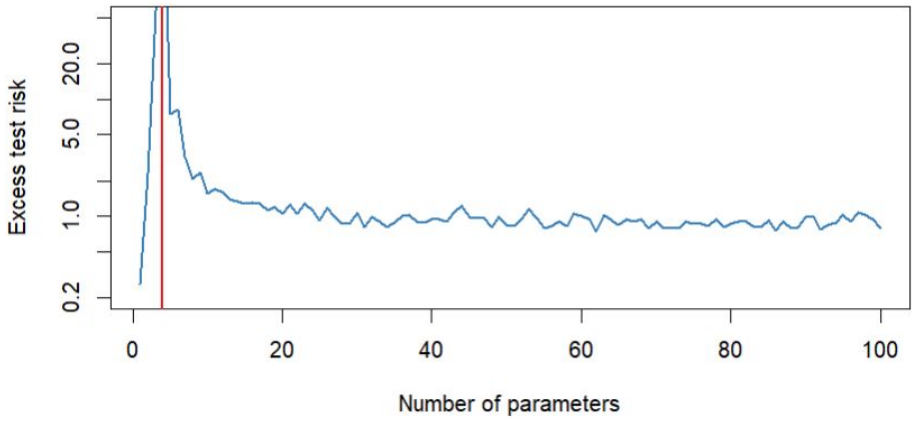

Figure 4.4: Logarithm of excess test risk (blue line) and interpolation threshold (red line) for polynomial eigenvalue decay with $\alpha = 2$.

4.6.3 Exponential decay of eigenvalues

Next, we look at exponential decay of eigenvalues: $\lambda_i = e^{-i} + \Delta_n$, with Δ_n a small perturbation. We take Σ to be the diagonal matrix with entries $\Sigma_{i,i} = e^{-i} + \Delta_n$. We choose $\Delta_n = \frac{1}{n^3}$. We know from Lemma 3 that beneficial overfitting occurs if

$$p(n) = \omega(n) \quad \text{and} \quad \omega(ne^{-n}) = \Delta_n p(n) = o(n)$$

If we take $n = 5$, then for $p = 100$ indeed $p \gg 5$ and $ne^{-n} = 5e^{-5} < \Delta_n p(n) = 100/(5^3) = 0.8 < 5 = n$, so the choice of $\Delta_n = \frac{1}{n^3}$ is small enough. The training risk is shown in Figure 4.5. We observe that the test risk at $p = 100$ is indeed lower than at $p = 0$, so we have beneficial overfitting.

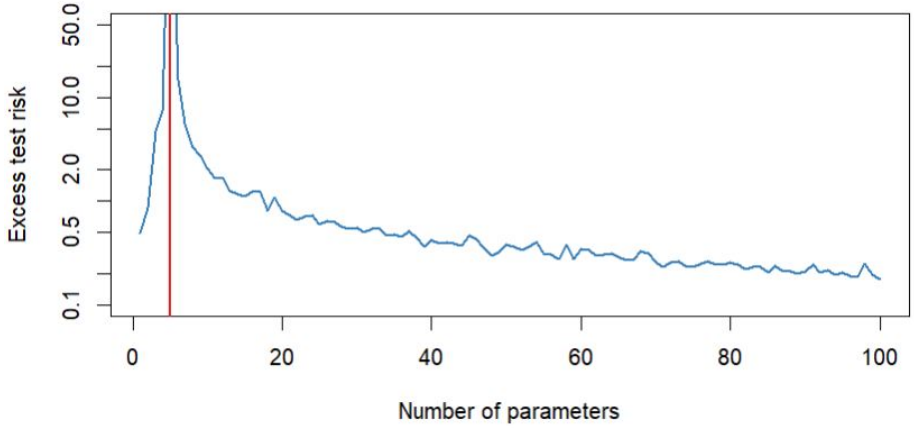

Figure 4.5: Logarithm of the excess test risk (blue line) and interpolation threshold (red line) for exponential decay with perturbation term $\Delta_n = \frac{1}{n^3}$.

4.6.4 Influence of normalizing true parameter vector

In this subsection we will investigate the influence of normalizing the true parameter vector θ^*. We set Σ equal to the identity matrix and take θ^* equal to

$$\text{(normalized)} \qquad \theta_j^* \propto 1 \qquad \text{s.t.} \quad ||\theta^*||^2 = 1, \qquad \text{(not normalized)} \qquad \theta_j = 1.$$

For the second choice of θ^* we have that the norm of θ^* grows with the number of parameters. From Theorem 4 we expect the training risk to diverge as the norm of θ^* diverges. The results are found in Figure 4.6. Indeed, if θ^* is normalized, we see converging behaviour. Without normalizing, the norm of θ^* grows with p and so also the test risk of the min-norm solution grows with p, consistent with Theorem 4. Hence, for beneficial overfitting, the norm of θ^* is not allowed to grow with the number of parameters p.

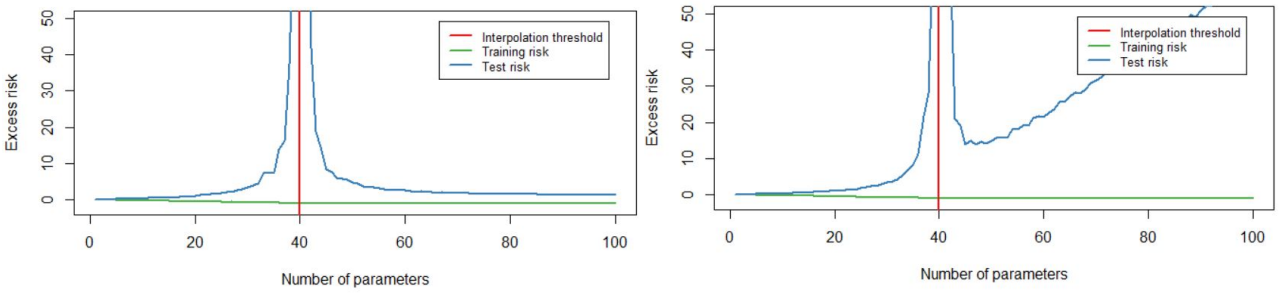

Figure 4.6: Excess training and test risk for the isotropic Gaussian case. (Left) with normalization and (Right) without normalization.

4.6.5 Influence of label noise

Finally, we investigate the influence of the label noise on the test risk. We fix Σ to be the identity matrix and $\theta_j^* \propto 1$ normalized. We will try 4 different values of σ, namely $\sigma = 0$, $\sigma = 0.1$, $\sigma = 1$ and $\sigma = 10$. The results are shown in Figure 4.7 below.

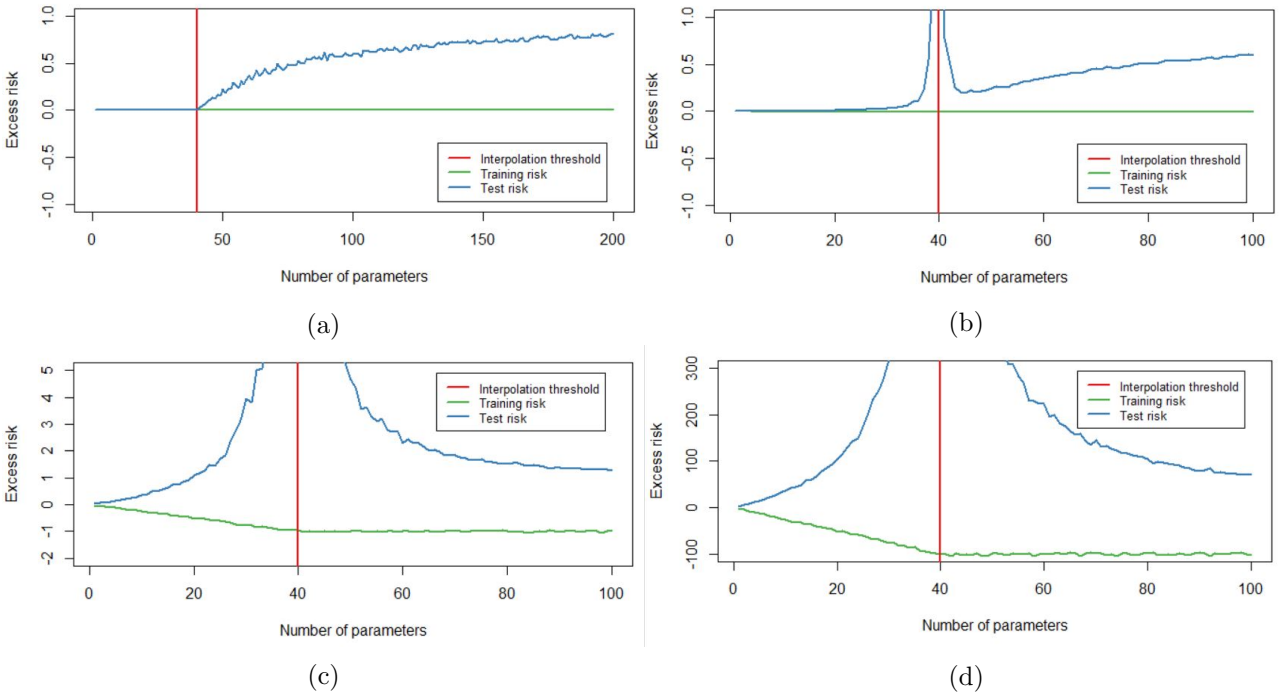

Figure 4.7: Isotropic Gaussian case, with (a) $\sigma = 0$, (b) $\sigma = 0.1$, (c) $\sigma = 1$ and (d) $\sigma = 10$.

Notice that, regardless of the value of σ, in all 4 cases the test risk converges to $||\theta^*||^2 = 1$. This is consistent with the results found in Chapter 2, where we saw that for no feature selection, we have $R(p) \to ||\theta^*||^2 + \sigma^2$ as $p \to \infty$. Since $R_{excess}(p) = R(p) - \sigma^2$, we have that $R_{excess}(p) \to ||\theta^*||^2$. In particular, notice how for $\sigma = 0$ the interpolation threshold completely disappears, which is to be expected, since this peak is caused entirely by the fact that we are trying to exactly fit noisy labels when $p = n$, which causes an explosion in the variance if $\sigma \neq 0$. Furthermore, in the under-parameterized limit we know from Chapter 2 that $R(p) \to \sigma^2$ as $p \to 0$. Hence, $R_{excess}(p) \to 0$ as $p \to 0$, which we can see in the graphs. Finally, we can conclude from the graphs that, regardless of the value of σ, no beneficial overfitting occurs in the isotropic Gaussian case, which is consistent with the results in Section 4.4. This also justifies why we can fix $\sigma = 1$ in this chapter, without influencing the beneficial overfitting behaviour.

4.7 Infinite dimensional case

In this chapter we have considered the finite dimensional version of the result in [5]. However, their result also holds in the infinite dimensional case, provided that the eigenvalues of Σ are summable, that is $\sum_{i=1}^{\infty} \lambda_i < \infty$. In this case, also the sequences of eigenvalues that are possible for beneficial overfitting change and we require more specific eigenvalue behaviour. The possible eigenvalues sequences that can occur are given in the Lemma below.

Lemma 4 (Theorem 31 from [5]). *If $\lambda_i = i^{-\alpha} \ln^{-\beta}(i+1)$ then Σ is benign if and only if*

$$\alpha = 1 \qquad \textit{and} \qquad \beta > 1.$$

If $\lambda_i = i^{-(1+\alpha_n)}$, then Σ is benign if and only if

$$\omega(1/n) = \alpha_n = o(1)$$

Proof See the proof in Appendix I of [5].

From Lemma 4 it is clear that in the infinite dimensional case, there is much less flexibility in choosing the eigenvalues of the covariance matrix if we want beneficial overfitting. Hence, we may expect to see less cases of beneficial overfitting for infinite dimensional models. Notice also that in the infinite dimensional setting of Section 2.6 we observed Double Descent behaviour, but indeed no beneficial overfitting.

4.8 When do we expect beneficial overfitting?

In this chapter we have looked at sub-Gaussian covariates with a general covariance matrix and no feature selection. We have looked at both polynomially decreasing and exponentially decreasing eigenvalue sequences of the covariance matrix. Results are summarized in the following overview.

We expect beneficial overfitting if:

- For general eigenvalue sequences that satisfy
$$\lim_{n\to\infty} \left(\frac{r_0(\Sigma)}{n} + \frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)} \right) = 0.$$
- In case the eigenvalues are of the form $\lambda_i = i^{-\alpha}$ with $\alpha \in (0,1)$ and $\omega(n) = p(n) = o(n^{1/(1-\alpha)})$.
- In case the eigenvalues are of the form $\lambda_i = i^{-\alpha}$ with $\alpha = 1$ and $\omega(e^{\sqrt{n}}) = p(n) = o(e^n)$.
- In case the eigenvalues are of the form $\lambda_i = e^{-i} + \Delta_n$, with Δ_n a small perturbation term such that $\omega(ne^{-n}) = \Delta_n p(n) = o(n)$ and $p(n) = \omega(n)$.

5 Kernel regression

Previously, we have looked at the linear regression model from equation (1.1) with different choices of the covariates vector. In this chapter we consider the more general kernel regression, which is closer to the setting for practical applications. We look at two different settings:

- **Setting 1**: we consider the finite dimensional linear regression problem, but we use a kernel estimator instead of the least-squares estimator. This setting is interesting, as now the implicit feature space dimension corresponding to the kernel function is different from the input dimension of the data, and it can even be infinite.
- **Setting 2**: we consider the infinite-dimensional Gaussian process regression, but we use an estimator that only considers the first p basis functions. This is the more realistic setting, since the true function can be very complex as it has an infinite dimensional basis of eigenfunctions, whereas for the estimator we can only use a finite amount of basis functions.

In **Setting 1** we will consider 3 choices for the kernel function: the linear kernel, the quadratic kernel and the Gaussian kernel. For each of these kernels, we will experimentally look at the behaviour of the test risk against the number of parameters. We also show that for the linear kernel we retrieve the least-squares estimator. Furthermore, we look at the eigenvalue behaviour of the kernel matrix. In **Setting 2** we again perform experiments to show the test risk behaviour. Finally, we check whether the results from previous chapters can be applied in this setting as well. Most of the theory about kernels and Gaussian processes in this chapter is based on the book by Rasmussen et al. (2006) [12].

5.1 Setting 1

In the first setup, we consider the following underlying model

$$y = f(x) + \varepsilon,$$

with y the response, $f : \mathbb{R}^d \to \mathbb{R}$ the true (but unknown) function of the data $x \in \mathbb{R}^d$ and noise term $\varepsilon \sim N(0, \sigma^2)$. For linear regression, we have that $f(x) = x^T\theta$ with $\theta \in \mathbb{R}^d$ the unknown parameter vector. First, we will derive an expression for the kernel estimator $\hat{f}(x)$, making use of Gaussian processes. We define a Gaussian process as 'a stochastic process that has Gaussian distributed finite dimensional marginal distributions' [13] (p. 428). Suppose the true function f is a draw from a centered Gaussian process,

$$f(x) \sim GP(0, k(x, x)),$$

with covariance function or kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined as

$$k(x, x') = \mathbb{E}f(x)f(x'), \qquad x, x' \in \mathbb{R}^d.$$

Let $X \in \mathbb{R}^{n\times d}$ be the training data matrix with rows $x_i^T \in \mathbb{R}^d$ and $X^* \in \mathbb{R}^{n^*\times d}$ be the test data matrix with rows $(x_i^*)^T \in \mathbb{R}^d$, with n the number of training data points and n^* the number of test data points. Define the kernel matrix as

$$K(X, X^*) = \begin{bmatrix} k(x_1, x_1^*) & \cdots & k(x_1, x_{n^*}^*) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1^*) & \cdots & k(x_n, x_{n^*}^*) \end{bmatrix} \in \mathbb{R}^{n\times n^*}.$$

Furthermore, let

$$f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \in \mathbb{R}^n, \qquad f^* = \begin{pmatrix} f(x_1^*) \\ \vdots \\ f(x_{n^*}^*) \end{pmatrix} \in \mathbb{R}^{n^*}.$$

The definition of a Gaussian process given above now implies that for each finite collection of data points $\{x_1, \ldots, x_n\}$ and $\{x_1^*, \ldots, x_{n^*}^*\}$, we have that the vectors f and f^* both have a multivariate Gaussian distribution. Hence, we can express the joint distribution of y and f^* as

$$\begin{bmatrix} y \\ f^* \end{bmatrix} = \begin{bmatrix} f + \varepsilon \\ f^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \mathrm{Cov}(f+\varepsilon, f+\varepsilon) & \mathrm{Cov}(f+\varepsilon, f^*) \\ \mathrm{Cov}(f^*, f+\varepsilon) & \mathrm{Cov}(f^*, f^*) \end{bmatrix}\right),$$

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix}\right)$$

Using properties of the Gaussian distribution, we know that $f^*|(y, X, X^*)$ again has a Gaussian distribution with mean and covariance

$$\mathbb{E}(f^*|y, X, X^*) = K(X^*, X)[K(X,X) + \sigma^2 I]^{-1} y,$$

$$\mathrm{Cov}(f^*|y, X, X^*) = K(X^*, X^*) - K(X^*, X)[K(X,X) + \sigma^2 I]^{-1} K(X, X^*).$$

Taking the mean of this posterior distribution as the estimator for f, evaluated at a single data point $x^* \in \mathbb{R}^d$, we find the following kernel estimator

$$\hat{f}(x^*) = \sum_{j=1}^{n} \alpha_j k(x_j, x^*), \qquad \text{with} \quad \alpha = [K(X,X) + \sigma^2 I]^{-1} y \tag{5.1}$$

In this setting there is no clear choice for the number of parameters p. Hence, we will consider the same setup for the feature selection as in Chapter 2, such that the result for the linear kernel will be the same as the results from Chapter 2. So we take a matrix P, where either $P = \{1, \ldots, p\}$ (deterministic choice) or P is sampled Uniformly from $\{1, \ldots, d\}$ such that $|P| = p$. Then in the estimate for the coefficients α in (5.1) we only use p features, meaning that we use $X_P \in \mathbb{R}^{n \times p}$ instead of $X \in \mathbb{R}^{n \times d}$.

In the numerical experiments we will see that the location of the interpolation threshold depends on the dimension of the implicit feature space that is implied by the kernel function. To understand this, we have to know how the kernel and this implicit feature space are related. Through Mercer's representation theorem [12], we can express the kernel function $k(x, x')$ in terms of certain basis functions as follows

$$k(x, x') = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j^*(x') =: \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}, \tag{5.2}$$

where $\phi_j : \mathbb{R}^d \to \mathbb{R}$ are the basis functions with corresponding eigenvalues $\lambda_j \in \mathbb{R}$ and $\phi(.) \in \mathcal{H}$ the vector containing these basis functions, also called the feature vector. From the representation (5.2) we see that the choice of basis functions implies a kernel function and vice versa. A popular choice (see e.g. [3]) for the basis functions ϕ_j are the Fourier basis functions, which are of the from $\phi_j(x) = e^{-i\omega_j x}$, with frequencies $\omega_j \in \mathbb{R}$. The case of random Fourier features will be discussed in Chapter 6.

We consider three choices for the kernel function: linear, quadratic and Gaussian. For each choice, we look at the feature vector, which can be seen as a mapping from the input x to the features $\phi(x)$, and its dimension, which we will refer to as implicit feature space dimension. In case of the linear kernel, defined as $k(x, x') = x^T x'$, we expect to retrieve the least-squares estimator (see Section 5.1.1). The corresponding feature mapping is

$$\phi : x \mapsto \phi_{lin}(x) = x, \qquad \phi : \mathbb{R}^d \to \mathbb{R}^d.$$

Notice that now the input dimension (dimension of x) and implicit feature space dimension (dimension of $\phi(x)$) are the same. For the quadratic kernel, $k(x, x') = (1 + x^T x')^2$, it may be checked that the corresponding feature mapping is given by

$$\phi : x \mapsto \phi_{quad}(x), \qquad \phi : \mathbb{R}^d \to \mathbb{R}^{d^*},$$

with the dimension d^* of the implicit feature space equal to (see [12]) $d^* = \binom{d+2}{2} = \frac{1}{2}(d+2)(d+1)$. Finally, for the Gaussian kernel $k(x, x') = \exp(-\frac{1}{2l^2}||x - x'||^2)$, with length scale l, the corresponding feature mapping is

$$\phi : x \mapsto \phi_{gauss}(x), \qquad \phi : \mathbb{R}^d \to \mathcal{H},$$

with $\mathcal{H}$ an infinite-dimensional Hilbert space. In this case the dimension of the feature space is infinite. This is also where the computational advantage of kernels lies. If we were to use the basis functions directly and express the true function and its estimator in terms of these basis functions, then the model $y = f(x) + \varepsilon$ in matrix-vector form would become

$$y = \Phi\theta + \varepsilon$$

with $\Phi \in \mathbb{R}^{n \times d^*}$ the feature matrix containing the feature vector $\phi(x_i)$ as rows, and θ the vector containing the weights θ_j for each basis function $\phi_j(x)$, $j \in \{1, \ldots, d^*\}$. Now d^* may be very large (for the quadratic kernel) or even infinite (for the Gaussian kernel) and so doing computations with the feature matrix Φ directly can be computationally expensive or even impossible. However, if we were to use the kernel estimator from equation (5.1), then we would have to do computations with the kernel matrix $K(X, X) \in \mathbb{R}^{n \times n}$, which is much less expensive for large ($d^* \gg n$) implicit feature space dimension.

5.1.1 Special case: linear kernel

If we choose the kernel function to be the linear kernel, then in fact we can retrieve the ordinary least-squares estimator. Indeed, let $k(x, x')$ be the linear kernel, defined as $k(x, x') = x^T x'$, with $x, x' \in \mathbb{R}^d$. Consider data vectors $\{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$ and design matrix $X \in \mathbb{R}^{n \times d}$ with rows x_i^T. Then the kernel matrix $K = K(X, X)$ has entries

$$K_{i,j} = k(x_i, x_j) = x_i^T x_j = [XX^T]_{i,j}.$$

Hence, $K(X, X) = XX^T$. If we let $\hat{y} \in \mathbb{R}^n$ be the vector containing the estimates of $f(x)$ evaluated at the data vectors $\{x_1, \ldots, x_n\}$, then we can write

$$\hat{y} = K(K + \sigma^2 I)^{-1} y = XX^T(XX^T + \sigma^2 I)^{-1} y$$

$$= X[X^T(XX^T + \sigma^2 I)^{-1} y] = X\hat{\theta}_{\sigma^2}.$$

Here $\hat{\theta}_{\sigma^2} = X^T(XX^T + \sigma^2 I)^{-1} y$ corresponds to the least-squares solution to a ridge regression problem with regularization parameter $\lambda = \sigma^2$. Taking $\sigma = 0$, we thus retrieve the least-squares estimator $\hat{\theta} = \hat{\theta}_0$.

5.2 Numerical experiments for setting 1

In this first experimental setup we will look at the test risk for 3 choices of the kernel function: (1) linear kernel, (2) quadratic kernel and (3) the Gaussian kernel. We assume the true model is a linear regression model (so $f(x) = x^T\theta$) of dimension d and approximate $f(x)$ using the kernel estimator (5.1), with possibly infinite dimensional implicit feature space dimension. We will perform a Monte Carlo experiment, where we choose $M = 10$ Monte Carlo iterations, $n = 40$ training data points, $\sigma = 0$ label noise and input dimension $d = 100$. Furthermore, we take $n^* = 100$ test data points, which we deem high enough for accurately computing the test risk, while still having low enough computational time. In each iteration, we carry out the following steps:

- Sample training and test data matrix, $X \in \mathbb{R}^{n \times d}$ and $X^* \in \mathbb{R}^{n^* \times d}$, with entries from $N(0,1)$ distribution.
- Select features from the matrix X, similar to Chapter 2, with $P = \{1, \ldots, p\}$ (deterministic) or P sampled uniformly from $\{1, \ldots, d\}$ s.t. $|P| = p$ (random), resulting in $X_P \in \mathbb{R}^{n \times p}$. The matrix X_P is only used in the estimation of the coefficients α in (5.1).
- Generate labels $y \in \mathbb{R}^n$ with entries $f(x_i) + \varepsilon$, with x_i the i-th row of X and $\varepsilon \sim N(0, \sigma^2)$.
- Construct the vector of coefficients $\alpha = (K(X_P, X_P) + \sigma^2 I_n)^{-1} y$, where $K(X_P, X_P)$ is the kernel matrix with entries $k((x_P)_i, (x_P)_j)$. Now the kernel estimator evaluated in x is given by

$$\hat{f}(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x).$$

- Calculate training and test error

$$R_{train} = \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - \hat{f}(x_i))^2, \qquad R_{test} = \frac{1}{n^*} \sum_{i=1}^{n^*} (f(x_i^*) - \hat{f}(x_i^*))^2,$$

  where x_i are the rows of X and x_i^* the rows of X^*.

As an estimate for the training (test) error we then take the average over the M training (test) error values

$$\hat{R} = \frac{1}{M} \sum_{j=1}^{M} R_j.$$

5.2.1 Linear kernel

We start with considering the linear kernel $k(x, x') = x^T x'$. Taking $\sigma = 0$, we expect to see the same graphs as for the isotropic Gaussian case from Chapter 2. We consider the deterministic choice and the random choice of feature selection. Results can be seen in Figures 5.1 and 5.2. Compared to Figures 2.1 and 2.2, we can see that they are very similar and show the same Double Descent behaviour, as expected. Beneficial overfitting occurs only for the random feature selection, as in that case the minimum test risk lies in the over-parameterized regime.

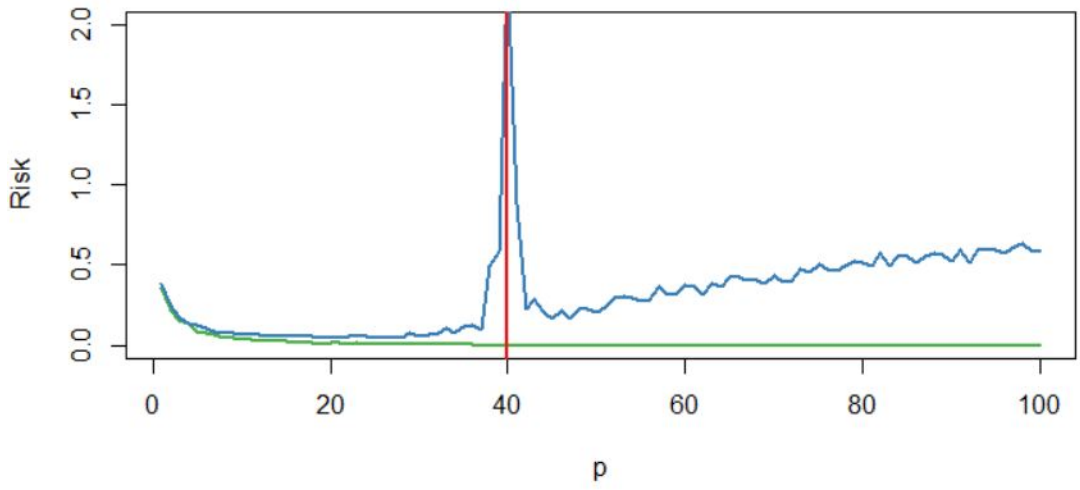

Figure 5.1: Linear kernel, with $\theta_j \propto 1/j$, $\sigma = 0$ and $P = \{1, \ldots, p\}$. Lines in the plot: (blue line) test risk, (green line) training risk, (red line) interpolation threshold.

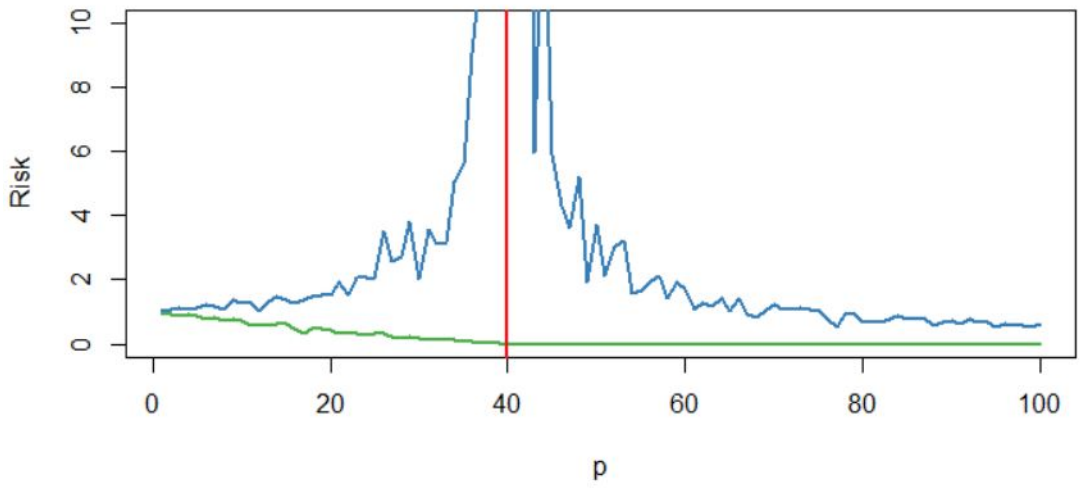

Figure 5.2: Linear kernel, with $\theta_j \propto 1/j$, $\sigma = 0$ and P sampled Uniformly from $\{1, \ldots, d\}$. Lines in the plot: (blue line) test risk, (green line) training risk, (red line) interpolation threshold.

5.2.2 Quadratic kernel

Next, we consider the quadratic kernel $k(x, x') = (1 + x^T x')^2$. As mentioned in the first part of this chapter, the quadratic kernel has implicit feature space dimension equal to $d^* = \frac{1}{2}(d+2)(d+1)$. This implies that, while we linearly increase the input dimension, the dimension of the implicit feature space will grow quadratically. We expect the interpolation threshold at $d^* = n$. Hence, we should be careful that we do not skip over the interpolation threshold, as there could be large jumps in d^* values, which would make it difficult to distinguish the under- and over-parameterized regimes. To make sure we exactly hit this interpolation threshold, one option is choosing $n = 36$, since $d^* = n$ is then realized for input dimension $d = 7$. Plots of the test risk against the number of parameters for the quadratic kernel can be found in Figures 5.3 and 5.4. Again, we observe the Double Descent behaviour. For the random selection of features, we observe beneficial overfitting. We indeed see that the interpolation threshold has changed position compared to the linear kernel, and the shape of the test risk curve is very similar to the shape for the linear kernel case.

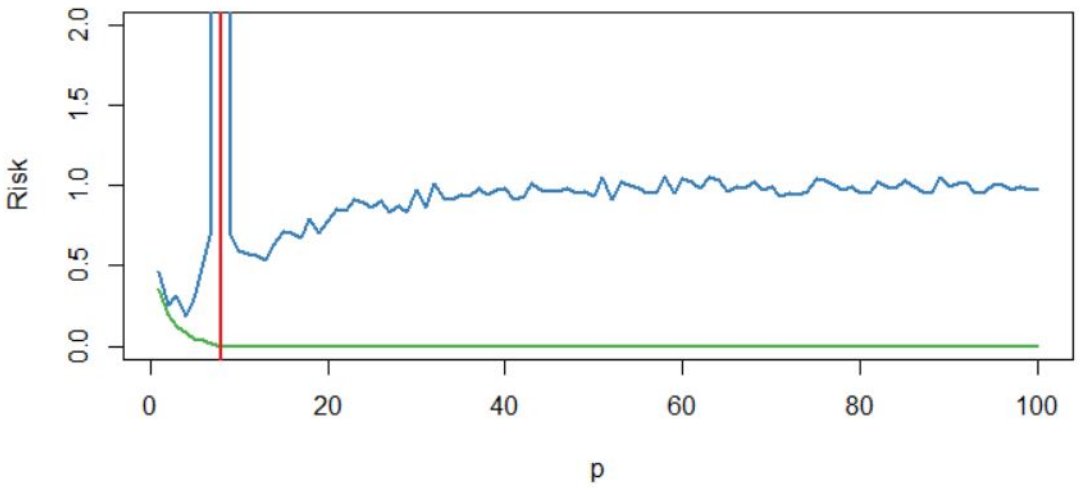

Figure 5.3: Quadratic kernel, with $\theta_j \propto 1/j$, $\sigma = 0$ and $P = \{1, \ldots, p\}$. Lines in the plot: (blue line) test risk, (green line) training risk, (red line) interpolation threshold.

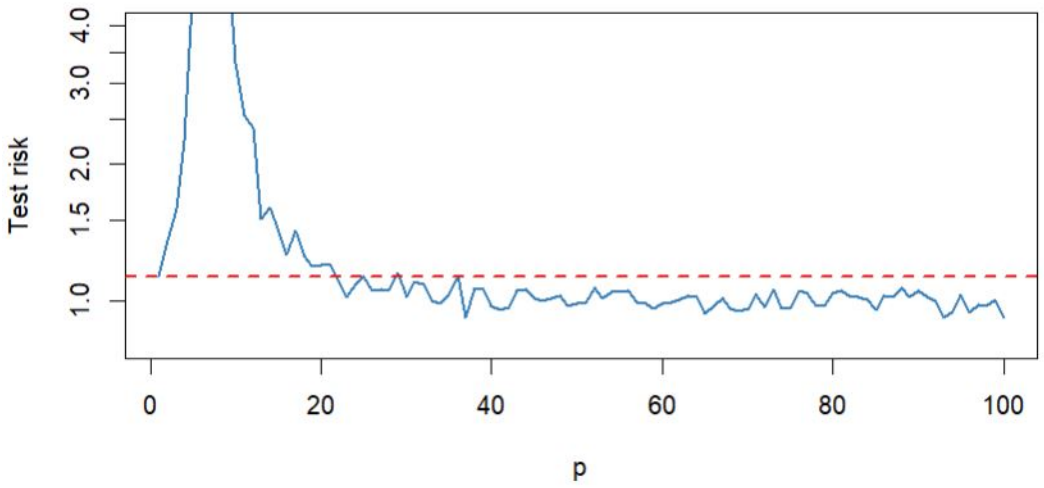

Figure 5.4: Quadratic kernel, with $\theta_j \propto 1/j$, $\sigma = 0$ and P sampled Uniformly from $\{1, \ldots, d\}$. Lines in the plot: (blue line) logarithm of the test risk, (red dotted line) test risk at $p = 0$.

5.2.3 Gaussian kernel

Finally, we consider the Gaussian kernel

$$k(x, x') = \exp\left(-\frac{1}{2l^2}||x - x'||^2\right),$$

with l the length-scale [12], which influences how many neighbouring data points we take into account. We will try two different values of the length-scale l: $l = 1$ and $l = 10$. We have seen in the first part of this chapter (Section 5.1) that for the Gaussian kernel the corresponding implicit feature space is infinite-dimensional, whereas the input space of the data is not. As the implicit feature space is infinite-dimensional, we may not expect to see an interpolation threshold peak.

Length scale 1

First we consider the case where $l = 1$. Then

$$k(x, x') = \exp\left(-\frac{1}{2}||x - x'||^2\right).$$

We look at both the deterministic feature selection and random feature selection. Logarithmic plots of the test risk against p can be seen in Figures 5.5 and 5.6. We now do not observe any Double Descent behaviour.

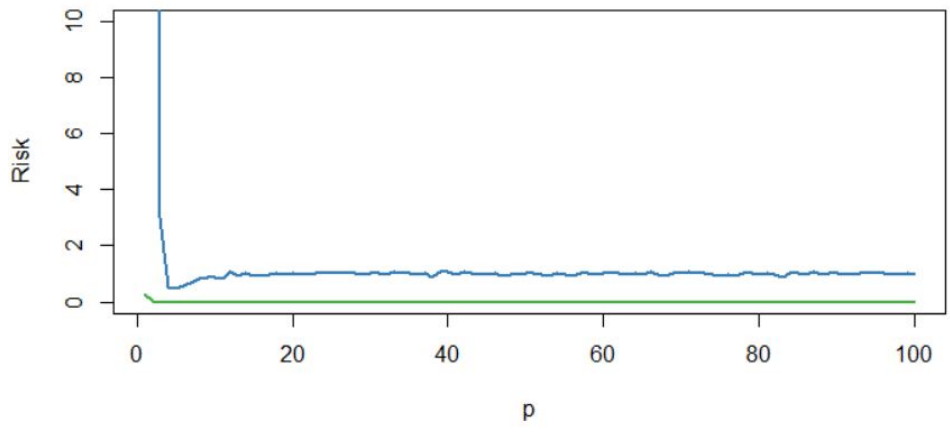

Figure 5.5: Gaussian kernel ($l = 1$), with $\theta_j \propto 1/j$, $\sigma = 0$ and $P = \{1, \ldots, p\}$. Lines in the plot: (blue line) test risk.

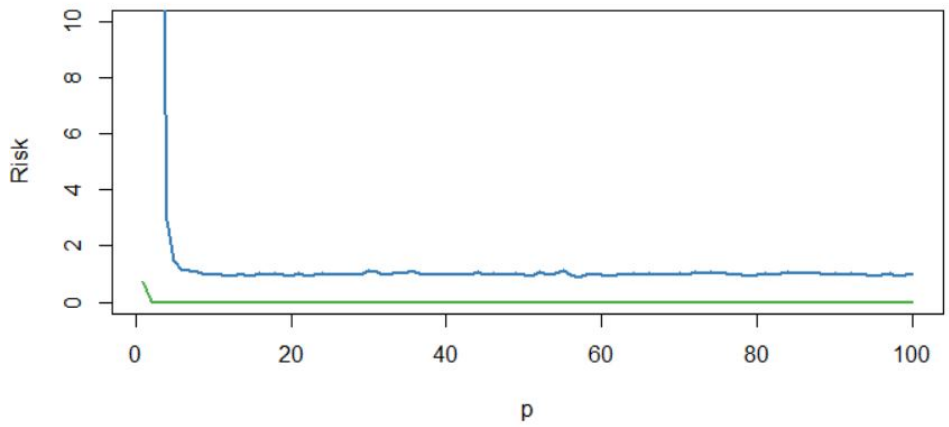

Figure 5.6: Gaussian kernel ($l = 1$), with $\theta_j \propto 1/j$, $\sigma = 0$ and P sampled Uniformly from $\{1, \ldots, d\}$. Lines in the plot: (blue line) test risk.

Length scale 10

The final case we look at is a length scale of $l = 10$. Now $k(x, x')$ approaches the value 1, meaning that we take every neighbouring data point into account for the weights $k(x, x_i)$. We then find the results in Figures 5.7 and 5.8. Now we do retrieve the Double Descent behaviour and also beneficial overfitting, as now the minimum test risk lies in the over-parameterized regime.

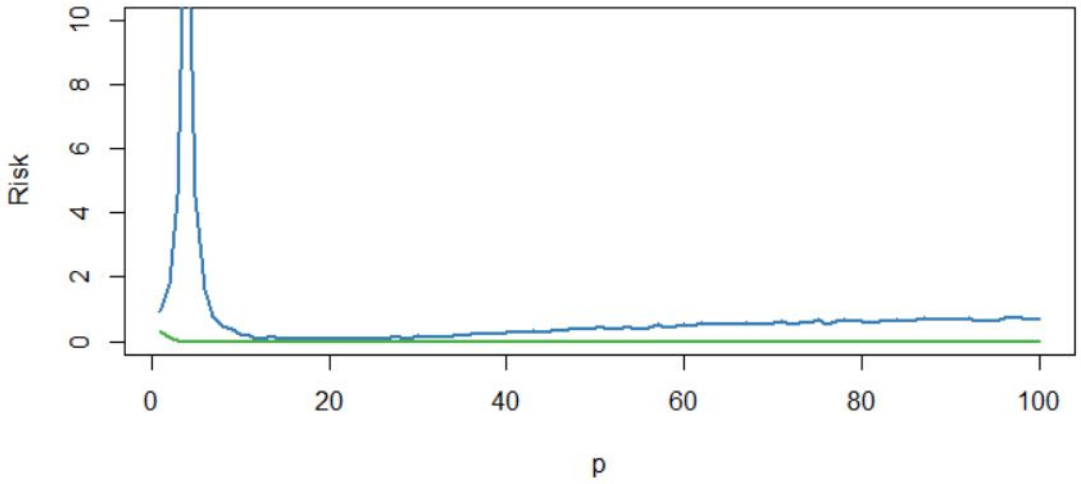

Figure 5.7: Gaussian kernel ($l = 10$), with $\theta_j \propto 1/j$, $\sigma = 0$ and $P = \{1, \ldots, p\}$. Lines in the plot: (blue line) test risk, (green line) training risk.

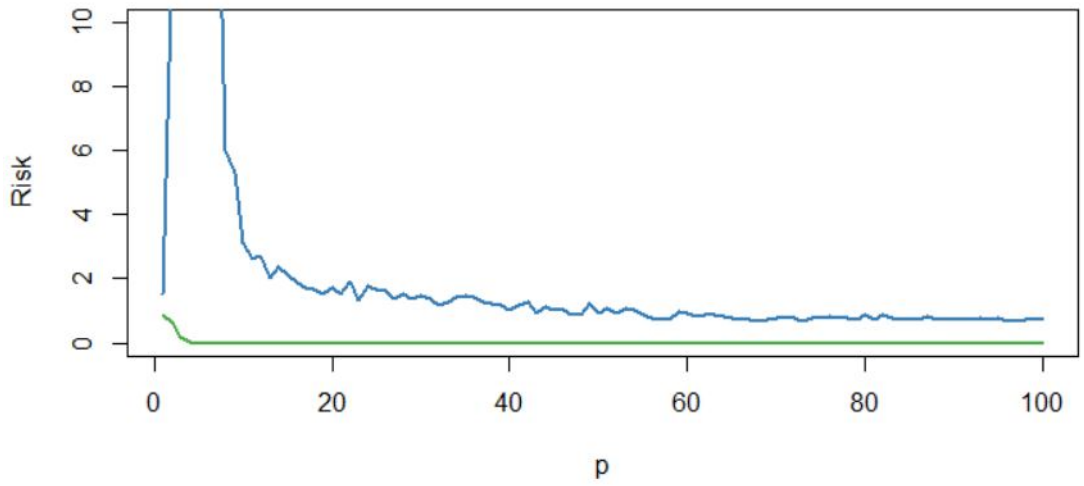

Figure 5.8: Gaussian kernel ($l = 10$), with $\theta_j \propto 1/j$, $\sigma = 0$ and P sampled Uniformly from $\{1, \ldots, d\}$. Lines in the plot: (blue line) test risk, (green line) training risk.

5.3 Comparison to previous results for setting 1

The experimental results for the linear and quadratic kernel, where data was sampled from $N(0, 1)$, agree well with the theory from Chapter 2 for an isotropic covariance matrix: the test risk shows Double Descent behaviour and beneficial overfitting only occurs for random feature selection (as $SNR = \infty$ and $\theta_j \propto 1/j$). However, in case of the Gaussian kernel, it is not immediately clear why the test risk behaviour is as given in Figures 5.5 up to 5.8. Notice especially the clear difference in behaviour for the case $l = 1$ and $l = 10$. In previous chapters we have seen that the test risk behaviour depends on the sequence of eigenvalues of the covariance matrix of the data. Based on the results from Chapter 2 we would not expect the beneficial overfitting we now observe in Figure 5.7 for the deterministic feature selection as $\theta_j \propto 1/j$. Hence, we now look instead at the behaviour of the eigenvalues of the kernel matrix $K(X, X)$, which is the covariance matrix for the Gaussian process $f(x)$.

Length scale 10

This time we first consider length scale $l = 10$. We know that for large length scale $l \to \infty$, we have

$$\lim_{l\to\infty} k(x, x') = \lim_{l\to\infty} e^{-\frac{1}{2l^2}(x-x')^2} = 1.$$

Hence $K \to 11^T$ as $l \to \infty$, with $1 \in \mathbb{R}^n$ the vector containing all ones. The eigenvalues of 11^T are easy to derive. Since each row contains all ones, we have that $\text{rank}(11^T) = 1$. This implies that there is only 1 non-zero eigenvalue. Since $n = \text{tr}(11^T) = \sum_{i=1}^n \lambda_i$ for the eigenvalues λ_i of $K(X, X)$, we must have that $\lambda_1 = n$ and $\lambda_2 = \ldots, \lambda_n = 0$. This agrees with the behaviour of the eigenvalues shown in Figure 5.9. We see that the eigenvalues first decrease nearly exponentially and then decreases slowly, resulting in a long flat tail, so according to Chapter 4 we expect beneficial overfitting, which is indeed what we observe in Figures 5.7 and 5.8. This may suggest that for kernel regression we should look at the eigenvalues of the kernel matrix instead of the data covariance matrix.

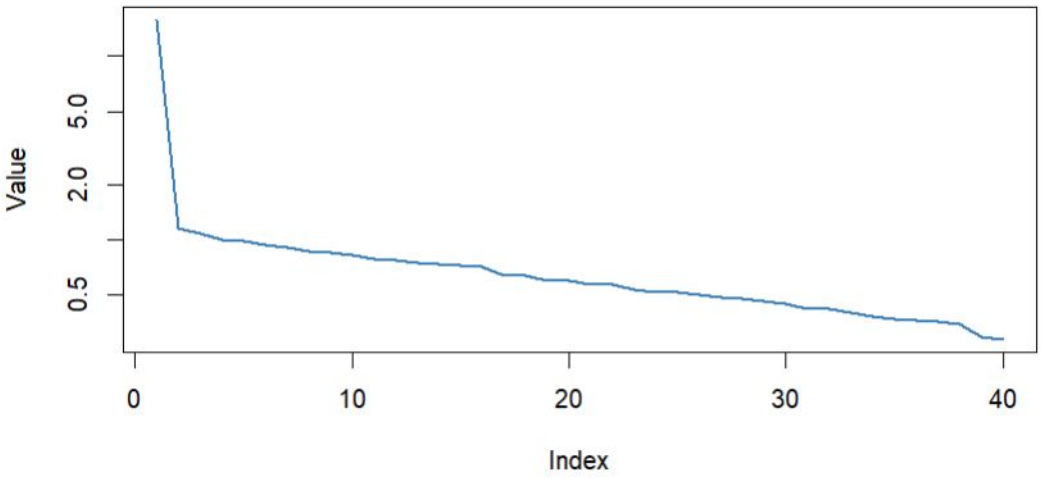

Figure 5.9: Logarithmic plot ot the eigenvalues of the kernel matrix $K(X, X)$ for length scale $l = 10$.

Length scale 1

Next, we look at the case $l = 1$. For small length scale $l \to 0$, we have

$$\lim_{l \to 0} k(x, x') = \lim_{l \to 0} e^{-\frac{1}{2l^2}(x-x')^2} = \mathbb{1}\{x = x'\},$$

where we define $k(x, x) := 1$. Hence $K \to I_n$ as $l \to 0$. So for small length scale, we expect the eigenvalues of K to be close to 1. Indeed, from the plot in Figure 5.10 we see that this is the case. If the results of Chapter 2 could be applied to the kernel matrix, then we would expect to see beneficial overfitting for the random feature selection, but (as $\theta_j \propto 1/j$) we do not expect beneficial overfitting for the deterministic feature selection. However, the results from Figures 5.5 and 5.6 do not show any Double Descent behaviour. This may suggest that only looking at the covariance matrix of the data or the kernel matrix is not enough and we need different results in this case. We will discuss this further in the Discussion part of Chapter 8.

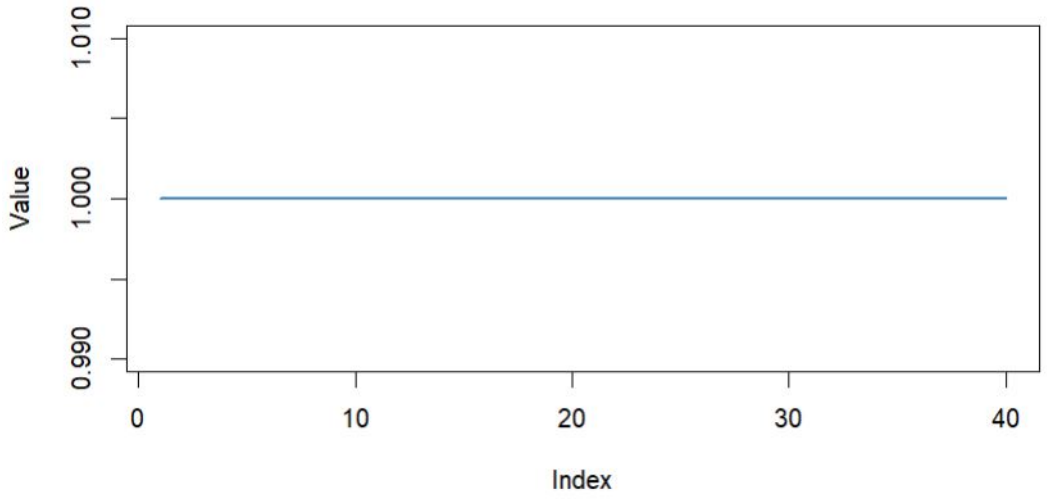

Figure 5.10: Eigenvalues of the kernel matrix $K(X, X)$ for length scale $l = 1$.

5.4 Setting 2

In the second setup of this chapter we try to approximate an infinite-dimensional problem by only using a selection of p basis functions. We assume the true function $f(x)$ is a Gaussian process with the Gaussian kernel function

$$k(x, x') = \exp\left(-\frac{1}{2l^2}(x - x')^2\right),$$

where the length scale l is a hyperparameter and $x, x' \in \mathbb{R}$. Then we know that $f(x)$ can be expressed in an infinite dimensional basis of eigenfunctions,

$$f(x) = \sum_{k=1}^{\infty} \theta_k \phi_k(x) = \langle \theta, \phi(x) \rangle_{\mathcal{H}},$$

where $x \in \mathbb{R}$ and $\theta, \phi(x) \in \mathcal{H}$ having components θ_i resp. $\phi_i(x)$. Notice that $x \in \mathbb{R}$, but we still have an infinite dimensional true model, which is different from linear regression, where x and ϕ should have the same dimension. We will now estimate this function by using only p basis functions. In that case $\hat{f}(x)$ has the following form

$$\hat{f}(x) = \sum_{k=1}^{p} \hat{\theta}_k \phi_k(x)$$

with $\hat{\theta}_k$ the coefficients to be estimated and $\phi_k(x)$ a Gaussian basis function corresponding to the Gaussian kernel. The Gaussian basis functions are of the form

$$\phi_k(x) = \exp\left(-\frac{1}{l^2}(x - v_k)^2\right)$$

where we choose the center v_k as $v_k \sim \text{Unif}([0, L])$, with L the length of the interval on which $f(x)$ is defined. For simplicity we take $L = n + n^*$, with n training and n^* test data points being the natural numbers in the interval $[0, L]$, so we only make observations at $x = 1, 2, \ldots, n + n^*$. The Gaussian basis functions are chosen such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} = \sum_{k=1}^{\infty} \phi_k(x)\phi_k(x') \propto \exp\left(-\frac{1}{2l^2}(x - x')^2\right).$$

See the computation on p.84 in [12], which consists of replacing the sum by an integral over all possible values v_k. Notice that the length scale l from the basis function $\phi_k(x)$ is scaled by a factor $\sqrt{2}$ in the kernel function. Next, we analyze the behaviour of the test risk of the estimate $\hat{f}(x)$ through numerical experiments.

5.5 Numerical experiments for setting 2

In the numerical experiment, we first simulate the Gaussian process $f(x)$ once, with seed(0) in R. We take $n = 50$ training data points and $n^* = 100$ test data points. We simulate $f(x)$ as follows:

- Let $X = (1, \ldots, L)$, with $L = n + n^*$, where n is the number of training data points and n^* the number of test data points.
- Construct the kernel matrix $K(X, X)$ with entries $K_{i,j} = k(x_i, x_j)$.
- Simulate $f(x)$ from the multivariate Gaussian distribution:

$$f(x) \sim N(0, K(X, X)).$$

Next, we start with the Monte Carlo simulation. We take $\sigma = 1$ as label noise, the number of parameters p from 1 until 500 and $M = 10$ Monte Carlo iterations. In each iteration, we perform the following actions:

- Sample $X_{train} \in \mathbb{R}^n$ from X. The remaining data points are put in $X_{test} \in \mathbb{R}^{n^*}$.
- Generate training labels
$$y_{train} = f_{train} + \varepsilon_{train},$$
where f_{train} is the vector with entries f evaluated at each training data point, and $\varepsilon_{train} \sim N(0, I_n)$. Similarly for the test labels y_{test}.
- Construct feature matrix $\Phi_{train} \in \mathbb{R}^{n \times p}$ with entries $\Phi_{i,j} = \exp(-(X_{train}[i] - v_j)^2/(l^2))$, where $v_j \sim \text{Unif}[0, L]$ and $L = n + n^*$. Similarly for Φ_{test}.
- Calculate vector of coefficients $\hat{\theta} \in \mathbb{R}^p$,
$$\hat{\theta} = \Phi_{train}^{\dagger} y_{train}$$
- Calculate estimate $\hat{y}$, which is equal to $\hat{f}$ evaluated at the training data points,
$$\hat{y}_{train} = \Phi_{train}\hat{\theta}$$
Similarly for $\hat{y}_{test}$.
- Finally, calculate training and test risk
$$R_{train} = \frac{1}{n}||y_{train} - \hat{y}_{train}||^2, \qquad R_{test} = \frac{1}{n^*}||y_{test} - \hat{y}_{test}||^2$$

As the data now ranges from 1 to $L = n + n^* = 150$, we will try the values $l = 10$ and $l = 100$ for the length scale of the Gaussian kernel. The results can be seen in Figures 5.11 and 5.12 below. We observe a clear Double Descent behaviour in the test risk, with an interpolation threshold at $n = 50$. There is no beneficial overfitting, but it seems that for a large number of parameters the test risk asymptotes to the value at $p = 0$.

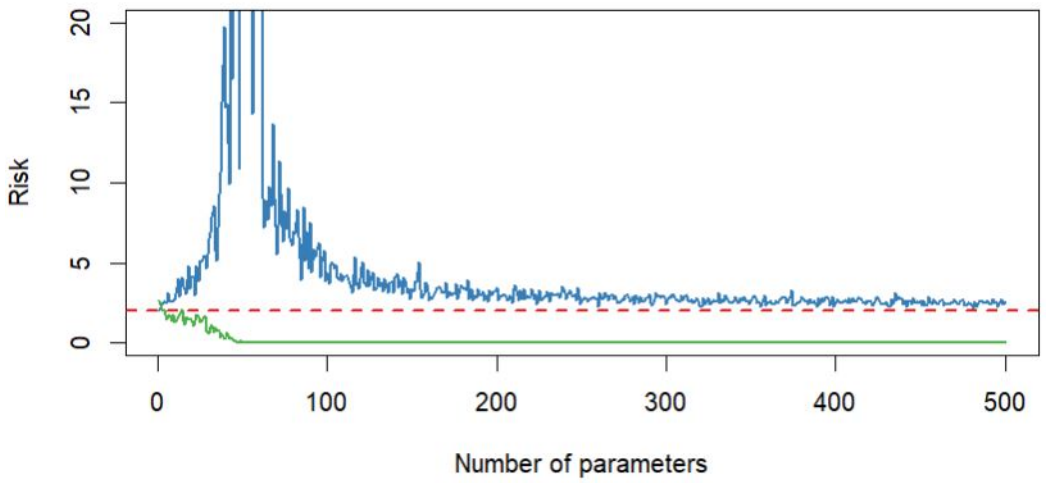

Figure 5.11: Gaussian kernel, $l = 10$, deterministic choice of features, $\sigma = 1$. Shown are: (blue line) test risk, (green line) training risk and (dotted red line) minimum test risk.

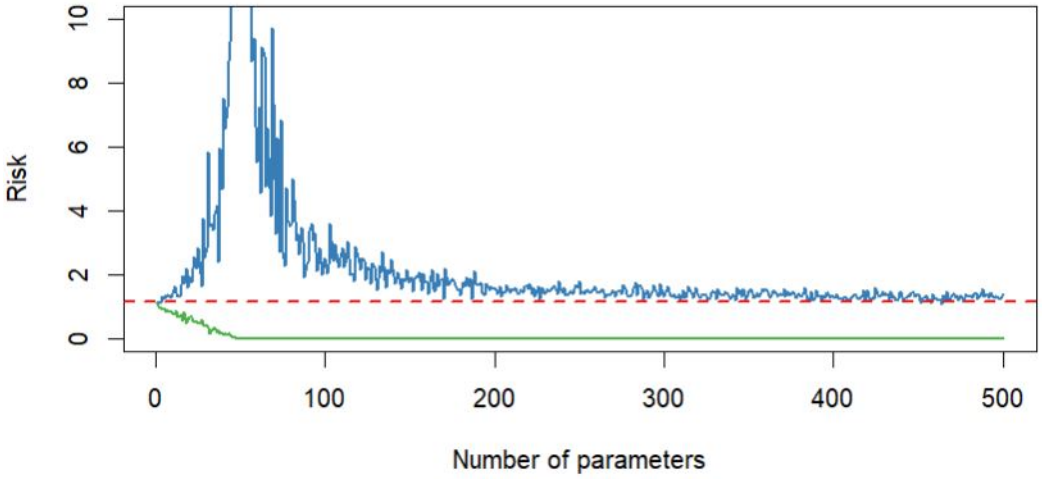

Figure 5.12: Gaussian kernel, $l = 100$, deterministic choice of features, $\sigma = 1$. Shown are: (blue line) test risk, (green line) training risk and (dotted red line) minimum test risk.

5.6 Comparison to previous results for setting 2

Based on the results from setting 1, we again consider the eigenvalues of the kernel matrix $K(X, X)$, instead of the covariance matrix of the data, for both $l = 10$ and $l = 100$. For $l = 10$ the eigenvalue decay is slower than exponential decay, see Figure 5.13, and for $l = 100$ the decay is approximately exponential, see Figure 5.14. For $l = 10$ we do not observe a long flat tail and based on the results from Chapter 4 may not expect beneficial overfitting. For $l = 100$ we do observe a long flat tail and hence we may expect beneficial overfitting. These observations do not really explain the test risk behaviour as shown in Figures 5.11 and 5.12, as strictly speaking, these figures do not show beneficial overfitting. Similar to Section 5.3 it seems that only looking at the eigenvalues of the kernel matrix is not enough to explain the Double Descent and beneficial overfitting behaviour, at least when we consider the Gaussian kernel.

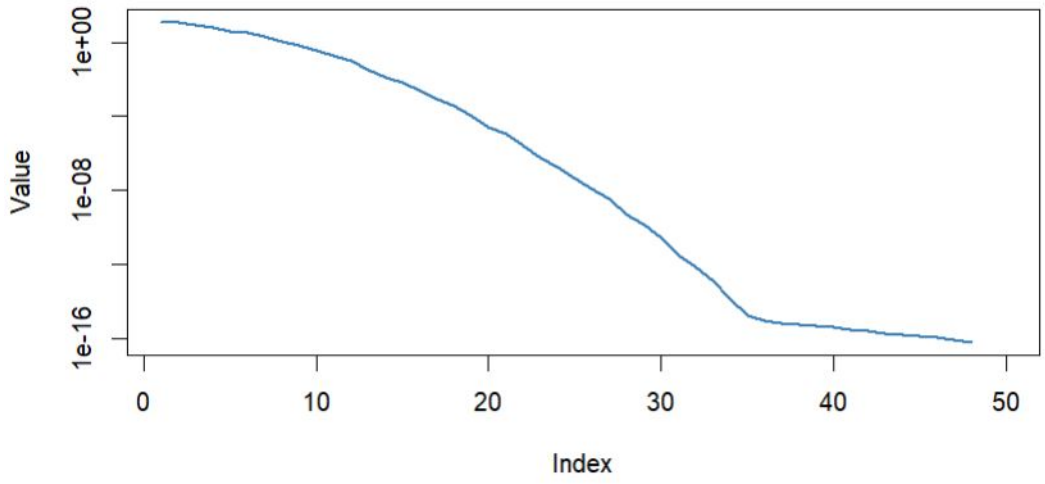

Figure 5.13: Logarithmic plot of the eigenvalues of the kernel matrix for the Gaussian kernel for length scale $l = 10$.

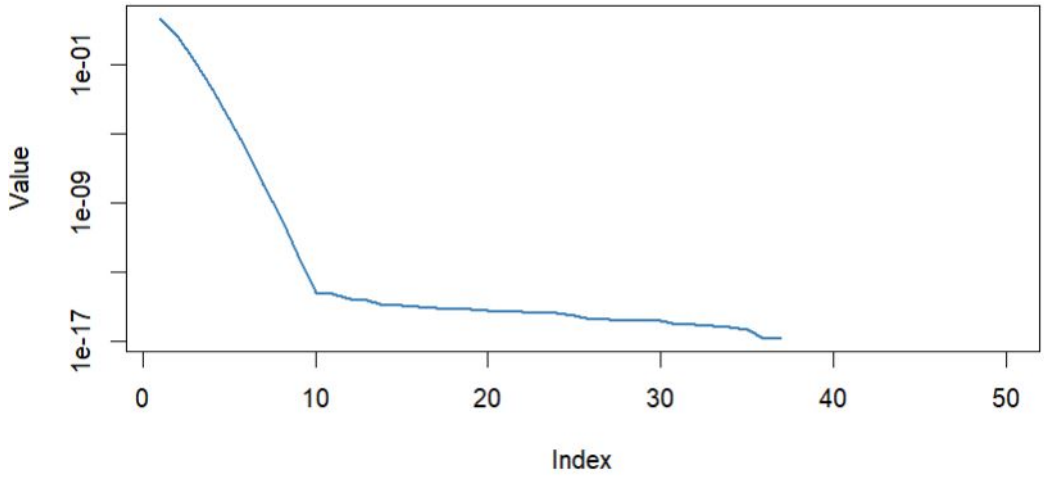

Figure 5.14: Logarithmic plot of the eigenvalues of the kernel matrix for the Gaussian kernel for length scale $l = 100$.

6 Fourier features

In the second setting of the previous chapter we have looked at the Gaussian kernel and Gaussian basis functions, which are of the form $\phi_k(x) = e^{-\gamma(x-v_k)^2}$, with $\gamma > 0$ and v_k the center of the basis function. Instead of Gaussian basis functions, we now look at another popular choice: Fourier basis functions, which are of the form $\phi_k(x) = e^{-i\omega_k x}$, with $\omega_k \in \mathbb{R}$ the frequency. It is known that, as the number of features diverges to infinity, the random Fourier features approach the Gaussian kernel, see e.g. [14]. The setting of a noise-free random Fourier features mdoel is described in Belkin et al. (2020) [4]. They derive an expression for the asymptotic test risk of $\hat{\theta}$, with $\frac{n}{d}$ and $\frac{p}{d}$ fixed as $p, n, d \to \infty$. We will check in this chapter whether beneficial overfitting occurs in this scenario and experimentally verify the result of [4].

6.1 Setting

Belkin et al. [4] consider a noise-free Fourier series model. They assume the following noise-free regression model

$$y = \Phi\theta$$

with $y \in \mathbb{C}^d$ the response vector, $\theta \in \mathbb{C}^d$ the true parameter vector and $\Phi \in \mathbb{C}^{d\times d}$ the $d \times d$ Discrete Fourier Transform matrix with entries

$$\Phi_{i,j} = \frac{1}{\sqrt{d}}\omega^{(i-1)(j-1)}$$

where $\omega := \exp(-2\pi i/d) \in \mathbb{C}$. If we write out the matrix-vector multiplication $y = \Phi\theta$, we get

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_d \end{pmatrix} = \frac{1}{\sqrt{d}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{d-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(d-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{d-1} & \omega^{2(d-1)} & \cdots & \omega^{(d-1)(d-1)} \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \\ \theta_d \end{pmatrix}$$

We can rewrite this as a system of equations

$$\begin{cases} y_1 = \frac{1}{\sqrt{d}}(\theta_1 + \theta_2 + \ldots + \theta_d) \\ y_2 = \frac{1}{\sqrt{d}}(\theta_1 + \theta_2\omega + \theta_3\omega^2 + \ldots + \theta_d\omega^{d-1}) \\ y_3 = \frac{1}{\sqrt{d}}(\theta_1 + \theta_2\omega^2 + \theta_3\omega^4 + \ldots + \theta_d\omega^{2(d-1)}) \\ \vdots \\ y_d = \frac{1}{\sqrt{d}}(\theta_1 + \theta_2\omega^{d-1} + \theta_3\omega^{2(d-1)} + \ldots + \theta_d\omega^{(d-1)(d-1)}) \end{cases}$$

Hence, we have that

$$y_j = \frac{1}{\sqrt{d}}\sum_{k=1}^{d}\theta_k\omega^{(k-1)(j-1)} = \sum_{k=1}^{d}\frac{\theta_k}{\sqrt{d}}e^{-2\pi i(k-1)(j-1)/d} = \sum_{k=1}^{d}\frac{\theta_k}{\sqrt{d}}\phi_k(x_j)$$

for $j = 1, \ldots, d$. Here the basis functions $\phi_k(x_j)$ are the Fourier basis functions

$$\phi_k(x_j) = e^{-2\pi i(k-1)x_j}$$

evaluated at points $x_j = (j-1)/d$. So we can indeed view this as a random features model such as in Chapter 5 for the Gaussian basis functions. We consider the asymptotic regime as $d \to \infty$, such that

$$\lim_{d\to\infty} \frac{p(d)}{d} = \rho, \qquad \lim_{d\to\infty} \frac{n(d)}{d} = \eta.$$

The design matrix Φ is constructed as follows. Let N, P be independent random subsets of $\{1, \dots, d\}$. Consider the random Bernoulli variables $B_i \sim \text{Ber}(\eta)$ and $C_i \sim \text{Ber}(\rho)$, for $i = 1, \dots, d$. Then $i \in N$ if $B_i = 1$ and $i \in P$ if $C_i = 1$. This results in a $|N| \times |P|$ design matrix $\Phi_{N,P}$ and a $|N|$-dimensional response vector y_N. Similar as in Chapter 2 with the isotropic Gaussian model, $\Phi_{N,P}$ denotes the sub-matrix of Φ with rows from N and columns from P. Note that the expected cardinality of $|P|$ and $|N|$ is equal to p resp. n. We briefly show the computation for $|P|$ (it is similar for $|N|$). We have

$$\mathbb{E}|P| = \mathbb{E}\sum_{i=1}^{d} \mathbb{1}\{C_i = 1\} = \sum_{i=1}^{d} \mathbb{P}(C_i = 1) = d\rho = p.$$

We estimate θ by using only a subset P out of the d features. Then the min-norm solution $\hat{\theta}$ is

$$\hat{\theta}_P := \Phi^{\dagger}_{N,P} y_N, \qquad \hat{\theta}_{P^c} := 0.$$

One of the properties of the discrete Fourier transform matrix (as mentioned in [4]) is that $\text{rank}(\Phi_{N,P}) = \min\{|N|, |P|\}$ and so the pseudo-inverse $\Phi^{\dagger}_{N,P}$ is given by

$$\Phi^{\dagger}_{N,P} = \begin{cases} \Phi^T_{N,P}(\Phi_{N,P}\Phi^T_{N,P})^{-1} & \text{if } |P| > |N| \\ (\Phi^T_{N,P}\Phi_{N,P})^{-1}\Phi^T_{N,P} & \text{if } |P| \leq |N| \end{cases}$$

We assume that θ is random (which is in contrast to the settings of most other papers) with covariance $\mathbb{E}(\theta\theta^T) = \frac{1}{d} I_d$. This random choice of θ is independent of N and P. We consider the setting where $\rho > \eta$, which is the over-parameterized setting where $p > n$. Notice that, if we define $D = \{1, \dots, d\}$, then

$$\Phi_{N,D}\hat{\theta} = (\Phi_{N,P}\Phi_{N,P^c})(\hat{\theta}_P, \hat{\theta}_{P^c})^T = \Phi_{N,P}\hat{\theta}_P + \Phi_{N,P^c}\hat{\theta}_{P^c} = \Phi_{N,P}\hat{\theta}_P$$

and

$$\begin{pmatrix} y_N \\ y_{N^c} \end{pmatrix} = \Phi\hat{\theta} = \begin{pmatrix} \Phi_{N,D} \\ \Phi_{N^c,D} \end{pmatrix} \hat{\theta} = \begin{pmatrix} \Phi_{N,D}\hat{\theta} \\ \Phi_{N^c,D}\hat{\theta} \end{pmatrix} = \begin{pmatrix} \Phi_{N,P}\hat{\theta}_P \\ \Phi_{N^c,P}\hat{\theta}_P \end{pmatrix}$$

Hence, we indeed have that $\hat{\theta}_P = \Phi^{\dagger}_{N,P} y_N$.

6.2 Main Result

The main result for the Fourier features model in [4] is as follows. Note that, since we assume a noise-free model, the risk of $\hat{\theta}$ is equal to the test risk.

Theorem 5 (Theorem 3 in [4])**.** *Consider the setting described above. Assume the true parameter vector θ is random with covariance $\mathbb{E}(\theta\theta^T) = \frac{1}{d} I_d$. Suppose $\eta = \lim_{d\to\infty} \frac{n(d)}{d}$ and $\rho = \lim_{d\to\infty} \frac{p(d)}{d}$. Assume $\rho \geq \eta$. Then the risk of $\hat{\theta}$ satisfies*

$$R_{over}(\hat{\theta}) = \mathbb{E}||\theta - \hat{\theta}||^2 \longrightarrow 1 - \eta\left(2 - \frac{\rho(1-\eta)}{\rho - \eta}\right)$$

as $d \to \infty$.

Proof. See the proof in [4].

6.3 Implications of Main Result

Theorem 5 provides an expression for the asymptotic test risk in the over-parameterized regime. This expression alone is not enough to determine whether beneficial overfitting will occur, as it only holds in the over-parameterized regime. However, we can easily find an expression for the under-parameterized regime in which $p = 0$. We do this as follows. In case $p = 0$, we have $\hat{\theta}_P = 0$. Since we define $\hat{\theta}_{P^c} = 0$, we thus have $\hat{\theta} = 0$. So $\mathbb{E}||\theta - \hat{\theta}||^2 = \mathbb{E}||\theta||^2 = 1$. Hence, the optimal under-parameterized risk, denoted by $R_{under}(\hat{\theta})$, is $R_{under}(\hat{\theta}) = 1$. Now for beneficial overfitting we need $R_{over}(\hat{\theta}) < R_{under}(\hat{\theta})$, that is

$$1 - \eta \left(2 - \frac{\rho(1-\eta)}{\rho - \eta}\right) < 1.$$

So for beneficial overfitting, ρ should satisfy the following condition

$$\rho > \frac{2\eta}{1+\eta}. \tag{6.1}$$

We will check this condition in some special cases and verify it experimentally in the next two sections.

6.4 Special cases

In this section we will look at some special cases of Theorem 5. We will consider the highly over-parameterized regime $\rho \to \infty$, and the case of no feature selection $\rho = 1$.

6.4.1 Highly over-parameterized limit

In the highly over-parameterized limit we have $\rho \to \infty$. Then the expression in Theorem 5 becomes

$$R_{over}(\hat{\theta}) = \lim_{\rho \to \infty} 1 - \eta \left(2 - \frac{\rho(1-\eta)}{\rho - \eta}\right) = 1 - \eta(2 - (1-\eta)) = 1 - \eta - 2\eta^2.$$

This is always less than $R_{under}(\hat{\theta}) = 1$ (since $\eta > 0$). This agrees with condition (6.1), since for $\rho \to \infty$ this condition is always satisfied. Hence, for the Fourier features model, if we keep increasing the number of parameters indefinitely, we will reach the point of beneficial overfitting.

6.4.2 No feature selection

In case there is no feature selection, we have $p = d$ and hence $\rho = 1$. Then the expression in Theorem 5 becomes

$$R_{over}(\hat{\theta}) = \lim_{\rho \to 1} 1 - \eta \left(2 - \frac{\rho(1-\eta)}{1 - \eta}\right) = 1 - \eta$$

This is again always less than $R_{under}(\hat{\theta}) = 1$. If $\rho = 1$, criterion (6.1) is satisfied if $1 > \frac{2\eta}{1+\eta}$. Hence we should have $\eta < 1$. This seems to imply a condition on η. However, since $\rho = 1$ and $\rho \geq \eta$, we must have that $\eta \leq 1$, so condition (6.1) is always satisfied.

6.5 Numerical experiments

In this section we will verify the result of Theorem 5 experimentally. In the numerical experiments, we will check two different scenarios, which consider different choices for N and P.

- Choosing N and P based on independent Bernoulli random variables with mean η resp. ρ. This is the setting described in [4] and in which Theorem 5 holds.
- Choosing N and P to be uniform subsets of D of cardinality n resp. p. This is the setting similar to the one discussed in Chapter 2 for random feature selection.

We consider the same setting as in the paper of Belkin et al. [4]. We take $d = 1024$, $n = 256$ and p ranging from 1 to d. We sample θ^* uniformly from the unit sphere in $\mathbb{R}^d$ by sampling $\theta^* \sim N(0, \frac{1}{d} I_d)$ and dividing by $||\theta^*||^2$ to normalize θ^*. Let $\Phi \in \mathbb{R}^{d \times d}$ be the discrete Fourier transform matrix. We have response vector $y = \Phi\theta^*$. We perform only $M = 10$ Monte Carlo iterations, to keep the computational time small. In each iteration:

- Sample $(B_i)_{i=1}^d$ and $(C_i)_{i=1}^d$ from Ber(η) resp. Ber(ρ).
- For $i, j \in D = \{1, \ldots, d\}$, let $i \in N$ and $j \in P$ if $B_i = 1$ resp. $C_j = 1$. To avoid breakdowns, if either P or N is empty, set it equal to $\{1\}$.
- Let $\Phi_{N,P} = \Phi[N, P]$ and calculate the min-norm solution

$$|P| \geq |N|, \qquad \hat{\theta}_P = \Phi_{N,P}^T (\Phi_{N,P} \Phi_{N,P}^T)^{-1} y_N$$

$$|P| < |N|, \qquad \hat{\theta}_P = (\Phi_{N,P} \Phi_{N,P})^{-1} \Phi_{N,P}^T y_N$$

  and $\hat{\theta}_{P^c} = 0$.
- Calculate MSE of $\hat{\theta}$

$$R = ||\theta^* - \hat{\theta}||^2.$$

The result of this simulation is a sequence $(R_i)_{i=1}^M$ of MSE values. Our estimate of the MSE is then the average over the $M = 100$ Monte Carlo runs

$$\hat{R} = \frac{1}{M} \sum_{i=1}^{M} R_i$$

For our choice of n and d, we have: $\eta = n/d = 256/1024 = 0.25$, so criterion (6.1) prescribes

$$\rho > \frac{2\eta}{1+\eta} = \frac{0.5}{1.25} = 0.4.$$

Hence, we take p ranging from 1 to d such that we expect beneficial overfitting, as $p > 0.4d$. We will check this both for the case where we use independent Bernoulli variables and where we use Uniform subsets. Furthermore, the value of the asymptotic test risk as given in Theorem 5 at $p = d$ (so $\rho = 1$) is

$$1 - \eta \left(2 - \frac{\rho(1-\eta)}{\rho - \eta}\right) = 1 - \eta = 1 - 0.25 = 0.75.$$

So we expect the test risk in the over-parameterized regime to converge to a value of 0.75.

Independent Bernoulli variables

The first case we consider is when N and P are constructed using independent Bernoulli random variables with mean η resp. ρ. In Figure 6.1 are the results of the Monte Carlo simulation. On the y-axis we use a logarithmic scale. The spikes we observe in Figure 6.1 are either due to the fact that we only average over $M = 10$ MSE value for each p or because the cardinality of N and P is random. This means that at the interpolation threshold, when $p = n$, we do not necessarily have $|P| = |N|$. Similarly, around the interpolation threshold, we can accidentally have that $|P| = |N|$, causing a spike in the test risk. As expected, we can see beneficial overfitting: the minimum test risk is achieved in the over-parameterized regime. Also notice that the test risk indeed converges to a value of 0.75 as predicted by Theorem 5.

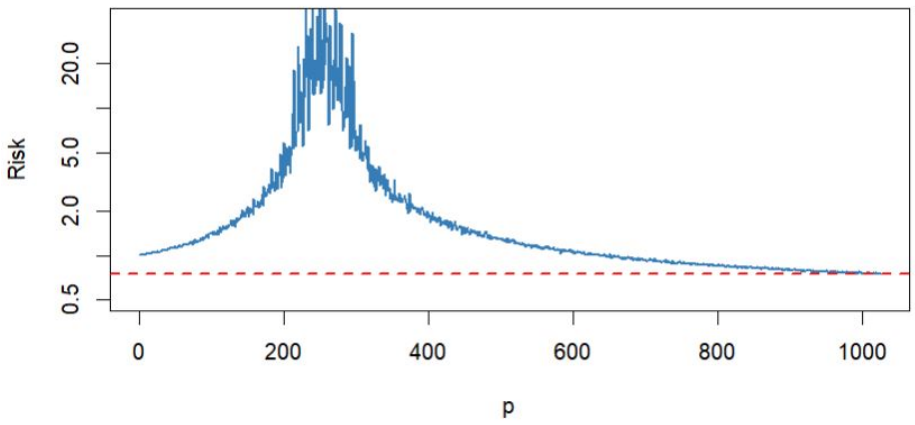

Figure 6.1: MSE $||\hat{\theta} - \theta^*||^2$ averaged over 10 runs, features are selected using independent Bernoulli variables and θ^* drawn uniformly from the unit sphere in $\mathbb{R}^d$. Blue line: test risk, red line: theoretical asymptotic test risk value of 0.75.

Uniform samples

Next, we consider the setting where N and P are Uniformly random subsets of $\{1, \ldots, d\}$ of cardinality n resp. p. A logarithmic plot of the average test risk is shown in Figure 6.2. We observe a much smoother curve than before, which is to be expected as the cardinality of N and P are now fixed. Therefore, the peak in test risk will now only occur exactly at the interpolation threshold. If we would keep increasing the number of Monte Carlo iterations M in the previous setting, then we expect to approach the curve of Figure 6.2. Indeed, after performing the Monte Carlo scheme, we observe $\{|P_1|, \ldots |P_M|\}$, which are the realizations for the cardinality of the set P (which is random), for a fixed value p. Then

$$\frac{1}{M} \sum_{j=1}^{M} |P_j| \longrightarrow \mathbb{E}|P| = p$$

as $M \to \infty$. Hence the average cardinality for P should become closer to its expected value p as $M \to \infty$ and Figure 6.1 should approach Figure 6.2. Finally, observe that again the test risk converges to a value of 0.75.

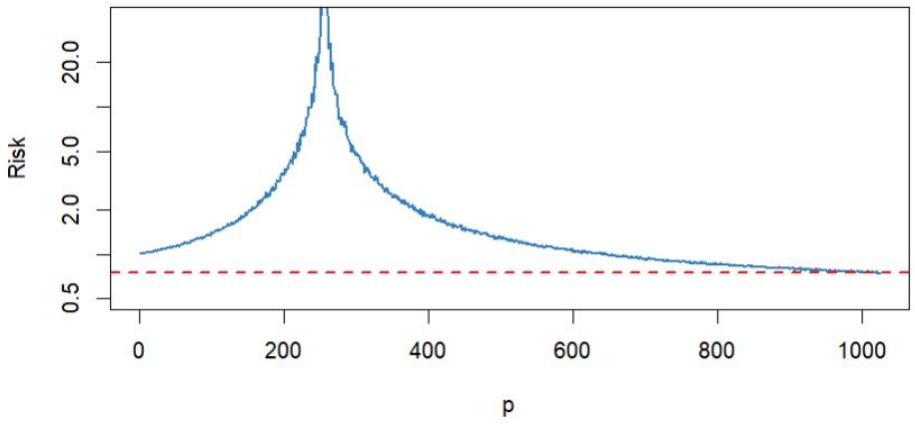

Figure 6.2: MSE $||\hat{\theta} - \theta^*||^2$ averaged over 10 runs, features are selected using Uniform samples and θ^* drawn uniformly from the unit sphere in $\mathbb{R}^d$. Blue line: test risk, red line: theoretical asymptotic test risk value of 0.75.

6.6 Comparison to previous results

In this section we briefly check whether the results for random Fourier features are consistent with previously discussed results. In [15] it is shown that the eigenvalues of the discrete Fourier transform matrix Φ are $\{-1, +1, -i, +i\}$. If we consider the kernel matrix $K = \Phi\Phi^*$, with * denoting the complex conjugate, then the eigenvalues of K are all equal to 1. This is similar to the identity covariance matrix in Chapter 2. There we saw that for identity covariance and random feature selection, beneficial overfitting occurs if the SNR $\frac{||\theta^*||^2}{\sigma^2}$ is large enough. In this chapter we assumed $\sigma = 0$, and hence SNR $= \infty$ and we would expect beneficial overfitting, which is indeed what we observe in Figure 6.2.

7 Classification model

Until now, we have only considered regression problems. However, Double Descent is not limited to regression. For instance, the example of Double Descent in machine learning models from Zhang et al. (2017) [1], the paper that was referenced in the Introduction, was in fact about classification problems. In this chapter we will look at one of the earliest examples of Double Descent in classification, described in Opper et al. (1990) [10]. They consider a simple classification model, with 3 different choices for the weight vector $\hat{\theta}$ and prove results for the learning and generalization ability. We will relate their result to the learning and test risk and try to verify it experimentally.

7.1 Setting

Opper et al. (1990) [10] consider the following classification model

$$y = f(x) = \text{sign}(x^T\theta), \tag{7.1}$$

with y the response variable, feature vector $x = (x_1, \ldots, x_p)^T \in \{+1, -1\}^p$ and parameter vector $\theta \in \mathbb{R}^p$. Notice that we assume noise-less labels y. We generate training data $(x_i, y_i)_{i=1}^n$, where we randomly choose the vectors $x_i \in \{+1, -1\}^p$, s.t. $\mathbb{P}(\{+1\}) = \mathbb{P}(\{-1\}) = \frac{1}{2}$, and where the y_i are generated according to equation (7.1). We assume a constant parameterization rate $\alpha := \frac{n}{p}$.

Opper et al. consider 3 ways to estimate θ, one of which is by using the pseudo-inverse. For $\alpha < 1$ (so $p > n$), the expression they give for $\hat{\theta}$ is

$$\hat{\theta}_j = \frac{1}{p}\sum_{i,k} y_i (C^{-1})_{i,k}(x_k)_j, \quad \text{with} \quad C_{i,k} = \frac{1}{p}\sum_j (x_i)_j (x_k)_j.$$

Here C is the correlation matrix and we denote by $(x_i)_j$ the j-th component of vector x_i. This expression for $\hat{\theta}$ is in fact the same as using the pseudo-inverse for $p > n$, as given in equation (1.3). Indeed, we have

$$(XX^T)_{i,k} = \langle X[i,:], X^T[:,k] \rangle = \sum_{j=1}^p X[i,j]X^T[j,k] = \sum_{j=1}^p (x_i)_j (x_k)_j = pC_{i,k}.$$

Here X is the design matrix with rows x_i^T, and we denote by $X[i,:]$ the i-th row of X. Hence $C = \frac{1}{p}XX^T$. Then for $\hat{\theta}$ we find

$$\hat{\theta}_j = \frac{1}{p}\sum_{i,k} y_i (C^{-1})_{i,k}(x_k)_j = \frac{1}{p}\sum_{i,k} (x_k)_j p (XX^T)^{-1}_{i,k} y_i$$

$$= \sum_i \sum_k X^T_{j,k} (XX^T)^{-1}_{k,i} y_i = \sum_i (X^T (XX^T)^{-1})_{j,i} y_i.$$

Hence $\hat{\theta} = X^T(XX^T)^{-1}y$, conform equation (1.3). For $\alpha > 1$ (so $p < n$), the least-squares solution $\hat{\theta} = X^T(XX^T)^{-1}y$ is used, which follows from minimizing $||y - X\theta||^2$. After training the model with data $(x_i, y_i)_{i=1}^n$, we obtain the estimate

$$\hat{y} = \hat{f}(x) = \text{sign}(x^T\hat{\theta})$$

7.2 Main Result

Opper et al. [10] consider the learning ability $L(\alpha)$ and generalization ability $G(\alpha)$, which they define as

- $L(\alpha)$: probability that $f(x_i) = \hat{f}(x_i)$ for data vector x_i,
- $G(\alpha)$: probability that $f(x_{new}) = \hat{f}(x_{new})$ for unseen data vector x_{new}.

We can relate these to the training and test risk as follows. Since we are working with a classification model, we use the 0-1 loss function

$$L(\hat{\theta}) = \mathbb{1}\{\text{sign}(x^T\theta) \neq \text{sign}(x^T\hat{\theta})\} = \mathbb{1}\{y \neq \hat{y}\}.$$

For data $(x_i, y_i)_{i=1}^n$, the training risk is

$$R_{train}(\hat{\theta}) = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{y_j \neq \hat{y}_j\}.$$

If we now consider the empirical probability distribution of the data, denoted by F_n, then

$$1 - L(\alpha) = \mathbb{P}(f(x_1) \neq \hat{f}(x_1)) = \mathbb{E}_{x_1 \sim F_n}(\mathbb{1}\{f(x_1) \neq \hat{f}(x_1)\})$$

$$= \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{f(x_j) \neq \hat{f}(x_j)\} = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{y_j \neq \hat{y}_j\}.$$

Hence $R_{train}(\hat{\theta}) = 1 - L(\alpha)$. Similarly, for the test risk we have

$$R_{test}(\hat{\theta}) = \mathbb{E}(\mathbb{1}\{y \neq \hat{y}\}) = \mathbb{P}(y \neq \hat{y}) = 1 - G(\alpha).$$

Hence $R_{test}(\hat{\theta}) = 1 - G(\alpha)$. We can now state the main result of Opper et al. [10] in terms of the test risk.

Theorem 6 (Main result in [10])**.** *For the classification problem in* (7.1)*, with* $||\theta|| = 1$*, and the choice of* $\hat{\theta}$ *where we use the pseudo-inverse, the test risk of* $\hat{\theta}$*, as a function of* $\alpha = \frac{n}{p}$*, is given by*

$$R_{test}(\alpha) = \begin{cases} \frac{1}{\pi}\cos^{-1}\sqrt{\frac{2\alpha(1-\alpha)}{\pi-2\alpha}} & \text{if } \alpha < 1 \\ \frac{1}{\pi}\cos^{-1}\sqrt{\frac{2(\alpha-1)}{\pi+2\alpha-4}} & \text{if } \alpha > 1 \end{cases}$$

Proof. See the proof in [10].

7.3 Numerical experiments

We will try to verify the formula in Theorem 6 through experiments in R. We fix the number of parameters $p = 100$ and vary the number of data points n from 1 to 400. This means that $\alpha = \frac{n}{p}$ will be between 0 and 4. The true parameter vector $\theta \in \mathbb{R}^p$ is a unit vector with components $\theta_j \propto 1$. We perform $M = 100$ Monte Carlo iterations according to the following scheme:

- Construct training and test data matrices, $X \in \mathbb{R}^{n\times p}$ and $X_{test} \in \mathbb{R}^{n\times p}$ with entries sampled uniformly from $\{+1, -1\}$.
- Calculate true labels
$$y = \text{sign}(X\theta^*), \qquad y_{test} = \text{sign}(X_{test}\theta^*).$$
- Calculate min-norm solution
$$p > n: \quad \hat{\theta} = X^T(XX^T)^{-1}y, \qquad p \leq n: \quad \hat{\theta} = (X^TX)^{-1}X^Ty.$$

- Calculate predictions

$$\hat{y} = \text{sign}(X\hat{\theta}), \qquad \hat{y}_{test} = \text{sign}(X_{test}\hat{\theta}).$$

- Calculate training and test risk

$$R_{train} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\hat{y} \neq y\}, \qquad R_{test} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\hat{y}_{test} \neq y_{test}\}.$$

We repeat above scheme $M = 100$ times and calculate the mean of the R_{train} and R_{test} values. The results are shown in Figure 7.1 below.

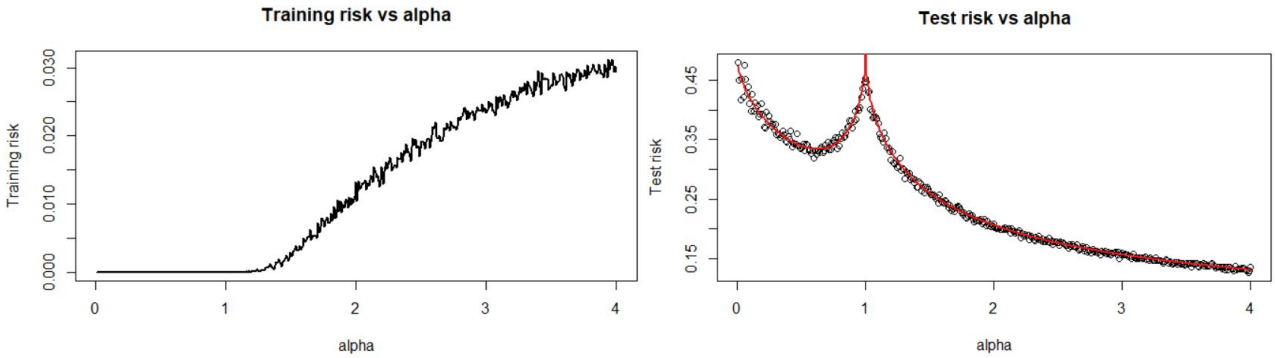

Figure 7.1: (left graph) Training risk plotted against α. (right graph) Experimental test risk (dots) and theoretical test risk (red line) plotted against α.

From the right graph in Figure 7.1 we can see that the agreement with the theoretical test risk is excellent. Furthermore, we observe the Double Descent behaviour for the test risk. In the under-parameterized regime ($\alpha > 1$), the test risk has a minimum at $\alpha = \infty$ and increases to the interpolation threshold ($\alpha = 1$), after which the test risk decreases again in the over-parameterized regime ($\alpha < 1$). The minimum test risk lies in the under-parameterized regime, so there is no beneficial overfitting.

7.4 Comparison to previous results

Previously, we have seen that beneficial overfitting depends on the covariance matrix of the data. Is this consistent with the classification setting from this chapter? We have selected data vectors x with components x_j, $j \in \{1, \ldots, p\}$ such that

$$\mathbb{P}(x_j = 1) = \mathbb{P}(x_j = -1) = \frac{1}{2}.$$

This implies for the mean and covariance of the data that

$$\mathbb{E}(x_j) = 1 \cdot \frac{1}{2} + (-1) \cdot \frac{1}{2} = 0$$

$$\text{Var}(x_j) = \mathbb{E}(x_j^2) = 1^2 \cdot \frac{1}{2} + (-1)^2 \cdot \frac{1}{2} = 1$$

Hence, the covariance matrix of the data is the identity matrix. In case of an identity covariance matrix, and no feature selection, we have seen that no beneficial overfitting was possible in the isotropic Gaussian case, see Chapter 2. This is consistent with Figure 7.1, in which the minimum test risk lies in the under-parameterized ($\alpha > 1$) regime.

8 Conclusion and Discussion

Conclusion

In this chapter we combine the previously discussed results and try to answer the main question: under what conditions does the optimal test risk lie in the over-parameterized regime? We have looked at several variations of the linear regression problem and considered kernel regression, random Fourier features and a classification model. Based on these models, we can draw the following conclusions.

First of all, we have seen that the Double Descent behaviour shows up in every model we considered in this thesis, regardless of the specific model (linear regression, kernel regression, random Fourier features, classification) or the way in which we select the features (deterministic or random). For beneficial overfitting however, the way in which we select the features is important as well. We distinguish between 3 options for feature selection: no feature selection, deterministic feature selection and random feature selection.

Without feature selection, beneficial overfitting is determined solely by the eigenvalues of the data covariance or kernel matrix. In Chapters 2, 4 and 7, we have seen instances where we do not select features and use the full vector of covariates. If the covariance matrix is equal to the identity matrix, or if the kernel matrix has all eigenvalues equal to 1, then no beneficial overfitting is possible. For both slowly and rapidly decaying eigenvalues of the covariance matrix, beneficial overfitting is possible, provided that we have a long tail of relatively small eigenvalues of the covariance matrix and a suitable number of parameters. Having a long tail of relatively small eigenvalues corresponds to having many low variance directions, which is the case when we are in a realistic over-parameterized regime where not all features have equal variance.

With deterministic feature selection and isotropic covariance, beneficial overfitting depends on the importance of features. In Chapters 2 and 5 we saw results for a deterministic feature selection with isotropic covariance matrix. This corresponds to a setting in which we select features using e.g. LASSO regression, before performing linear regression. In this setting, selecting the most important features (features with the highest weights as determined by the choice of parameter vector) does not lead to beneficial overfitting, whereas selecting features with the lowest weights does result in beneficial overfitting behaviour. This is consistent with selecting many low variance directions for beneficial overfitting if we would not perform feature selection.

With random feature selection and identity covariance, beneficial overfitting depends on the signal-to-noise ratio. In Chapters 2, 5 and 6 we looked at the random feature selection with all eigenvalues equal to 1 for the covariance matrix or kernel matrix. Beneficial overfitting occurs if the signal-to-noise ratio (SNR) is large enough. In machine learning, we often assume small label noise σ, so that the signal-to-noise ratio is large. Indeed, in the case of linear/quadratic kernel regression and random Fourier features, we assumed $\sigma = 0$ (so SNR $= \infty$) and we observed beneficial overfitting.

Linear regression results do not apply to Gaussian kernel regression. We have seen that the results in Chapter 5 for the linear and quadratic kernel, Chapter 6 and Chapter 7 seem to agree with the linear regression results for beneficial overfitting that were discussed in Chapters 2, 3 and 4: beneficial overfitting can be predicted by looking at the eigenvalues of the covariance matrix or kernel matrix, together with the way in which features are selected. However, in the case of Gaussian kernel regression from Chapter 5, this does not seem to be the case. Hence, to be able to predict whether beneficial overfitting will occur, more factors need to be considered.

Discussion

Next, we make some remarks about the results in this thesis and provide some directions for future research. First, it is important to note that the focus of this thesis is on Double Descent in the linear regression case with (sub-)Gaussian covariates. Hence, although these results seem to agree with the linear/quadratic kernel regression, random Fourier features and classification model from Chapters 5, 6 and 7, they do not hold in general, as we saw for the Gaussian kernel in Chapter 5. However, the results have made clear that the eigenvalues of the data covariance or kernel matrix and the way of selecting features are important to predict beneficial overfitting. Also, we believe it is still useful to compare simpler results to more general models, as it can sometimes be difficult to derive conditions for beneficial overfitting from more general and complicated results, which was the case when we considered the result in Chapter 3.

Furthermore, most of the Monte Carlo experiments we performed only contained a relatively small number of parameters and/or training data points, whereas in practice the number of parameters and data points may be much larger. However, the models we considered are relatively simple and hence the experiments and graphs are only meant as an illustrative example of Double Descent. More research is needed for Double Descent in deep neural networks, where the number of parameters is much larger and the model is more complex.

Moreover, we have not been able to fully explain the results for the Gaussian kernel from Chapter 5, based on the linear regression results. There are several interesting directions that may be able to explain this behaviour. Recall from the Introduction that Double Descent is related to a form of 'inductive bias' [3], where the bias is towards smoother solutions. This may explain the striking difference in test risk behaviour between Figures 5.5 and 5.6 for $l = 1$, and Figures 5.7 and 5.8 for $l = 10$, since for larger length scale l the Gaussian kernel estimator yields smoother solutions. Furthermore, it may be helpful to consider the eigenvalues of the matrix $K(X_P, X_P)$ instead of $K(X, X)$ in setting 1 of Chapter 5. In that case, we could compute characteristics of the eigenvalue sequence for each value of p, such as the smallest/largest eigenvalue of the kernel matrix K or the condition number of K, and look at a scatter plot of the test risk behaviour and this characteristic. Another direction is to look at the eigenvalues of a different matrix, such as in Liu et al. (2021) [16], in which it is shown that the shape of the Double Descent risk curve in case of kernel ridge regression depends on the eigenvalue behaviour of the matrix $\tilde{X} = \beta XX^T/d + \alpha 11^T$, with α, β constants depending on the choice of kernel function. Similar as what we found in this thesis, the eigenvalue behaviour of the covariance matrix and kernel matrix are important for beneficial overfitting. Another result from Liang et al. (2020) [17] considers the kernel ridgeless regression, where they find that the behaviour of the test risk for the minimum norm solution depends on the high dimensionality of the input data, the smoothness of the kernel function and the eigenvalue decay of the empirical covariance and kernel matrix, which confirms that we indeed have to consider more factors to fully explain the Double Descent behaviour.

Finally, in this thesis we have only considered one way of optimization, namely the least squares method for the under-parameterized regime and the minimum norm solution using the pseudo-inverse for the over-parameterized regime. However, as seen in e.g. [10], choosing a different optimization method can result in losing the Double Descent behaviour. Further literature about the effect of optimization on Double Descent can be found in [18], in which the test risk is decomposed in an optimization and a generalization term. Another way of selecting one of the infinitely many solutions in the over-parameterized regime is by using ensembles, which is used for example in random forests. In this setting, it seems that Double Descent behaviour is still possible, see e.g. [3].

Future directions

One important open question when it comes to Double Descent and beneficial overfitting is if and how the results for the simple (linear) case generalize to deep neural networks. A well-known connection is between the optimization algorithms for deep neural networks and linear regression. In deep neural networks, often the SGD algorithm is used for optimization. In e.g. [6] it is shown that its solution converges to the minimum norm least-squares solution if we initialize the SGD algorithm at $\theta = 0$. Another connection is that neural networks can also be described using kernel regression with the so-called neural tangent kernel, see e.g. Jacot et al. (2018) [19]. It turns out that very wide neural networks, trained with SGD with an appropriate initialization, can be well approximated by linear functions in a suitable kernel space, which corresponds to the neural tangent kernel. However this approximation depends on some strong assumptions, such as that the learnt coefficients θ_j cannot change too much from their initialization during training.

Another interesting question is whether beneficial overfitting is worth the extra computational cost that comes with it, as we are dealing with models that have a large amount of parameters, especially when the test risk only decreases slightly compared to the under-parameterized regime. In this thesis we have dealt with relatively simple models and so computational costs were not much of a concern, but already for the Fourier features model from Chapter 6 the computation of the test risk for p ranging from 1 to 1000 took a considerable amount of time, whereas the test risk only improved from a value of 1 to 0.75. Besides, we have not seen major improvements in test risk in the over-parameterized regime in this thesis, so for simple models beneficial overfitting may not be as interesting (but they can still be useful in understanding Double Descent for more complex models).

Furthermore, it turns out that, under certain conditions, optimization algorithms such as Stochastic Gradient Descent (SGD) have an exponential convergence rate in the over-parameterized regime. In Bassily et al. (2018) [20] and Liu et al. (2020) [21] it is shown that, if the loss function $\mathcal{L}$ satisfies the so-called PL* condition, then $\mathcal{L}$ has a global minimum and SGD converges exponentially to this minimum. Moreover, [21] show that over-parameterized systems, in particular sufficiently wide neural networks, satisfy the PL* condition around their initialization point, guaranteeing the exponential convergence of SGD.

Instead of looking at the performance of the test risk against the number of parameters, as we have done in this thesis, we can also look at the performance against the number of training data points. An extensive description and history of these so-called learning curves can be found in Viering et al. (2021) [22]. Two recent examples of Double Descent in learning curves are given in Nakkiran (2019) [23] and Nakkiran et al. (2019) [24]. In these papers it is shown that Double Descent can also occur if we look at the test risk against the number of training data points. Hence, this Double Descent can result in regimes (for $n < \infty$) where training with a larger number of samples hurts the performance. This is rather counter-intuitive and would imply that in some cases lowering the number of training samples can improve the model performance. Another option is to plot the test risk against the so-called effective degrees of freedom (see e.g. [8]), which is defined as $\mathrm{df} = \mathrm{tr}(H)$, where H is such that $\hat{y} = Hy$. In case of kernel regression, we have $H = K(K + \sigma^2 I)^{-1}$ and hence for $\sigma = 0$ the effective d.o.f. is equal to $\mathrm{df} = \mathrm{tr}(H) = \mathrm{tr}(I_n) = n$. This may be interesting to consider when the implicit feature space dimension of the kernel function is infinite dimensional (such as for the Gaussian kernel in Chapter 5).

Yet another direction lies in the use of Gaussian processes, which can be seen as the infinite-width limit of neural networks. In Harzli et al. (2021) [25] the Neural Network Gaussian Process is considered, which is the infinite-width limit of a particular neural network. They present an asymptotic expression for the test risk in the case that $\frac{n}{p}$ and $\frac{n}{d}$ remain fixed as $p, n, d \to \infty$, similar to the setting of [6] and [7]. Furthermore, they present conditions under which the Double Descent behaviour occurs. This seems like a particularly interesting direction, as we can draw connections between Gaussian processes and neural networks. This is also the direction that is argued for in Belkin et al. (2018) [26].

Final thoughts

All in all, we think there is still not a full understanding of the Double Descent phenomenon for deep neural networks. However, the large amount of literature that is emerging in recent years is a positive sign that we are getting closer. Double Descent in linear regression is now very well understood and also the understanding for kernel regression is increasing, although the analytical formulas for the test risk can quickly become rather complex. In this thesis we have seen that not all results can be applied or are easy to adapt to more complex models, as was the case when we looked at Gaussian kernel regression in Chapter 5. In order to fully understand Double Descent in deep neural networks, more research is needed on the connection between simpler models and deep networks, especially when more non-linearity is involved. An interesting recent work is from Frei et al. (2022) [27], in which classification in a two-layer neural network is considered where the model and learning dynamics are both non-linear.

Bibliography

[1] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires re-thinking generalization. *arXiv:1611.03530*, 2017. URL: https://arxiv.org/pdf/1611.03530.pdf.

[2] Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *arXiv:2102.06171*, 2021. URL: https://arxiv.org/pdf/2102.06171.pdf.

[3] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *PNAS*, 116(32):15849–15854, 2019. URL: https://www-pnas-org.tudelft.idm.oclc.org/content/pnas/116/32/15849.full.pdf.

[4] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2(4):1167–1180, 2020. URL: https://arxiv.org/pdf/1903.07571.pdf.

[5] Peter L. Bartlett, Philip M. Long, Gabor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *PNAS*, 117(48):30063–30070, 2020. URL: https://arxiv.org/pdf/1906.11300.pdf.

[6] T. Hastie, A. Montanari, S. Rosset, and R. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv:1903.08560*, 2020. URL: https://arxiv.org/pdf/1903.08560.pdf.

[7] Song Mei and Andrea Montanari. The generalization error of random features regression: precise asymptotics and double descent curve. *arXiv: 1908.05355*, 2020. URL: https://arxiv.org/pdf/1908.05355.pdf.

[8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2008. URL: https://hastie.su.domains/Papers/ESLII.pdf.

[9] Marco Loog, Tom Viering, Alexander Mey, Jesse H. Krijthe, and David M.J. Tax. A brief prehistory of double descent. *PNAS*, 117(20):10625–10626, 2020. URL: https://www-pnas-org.tudelft.idm.oclc.org/content/pnas/117/20/10625.full.pdf.

[10] M. Opper, W. Kinzel, J. Kleinz, and R. Nehl. On the ability of the optimal perceptron to generalise. *Journal of Physics A: Mathematical and General*, 23(11):581–586, 1990. URL: https://iopscience-iop-org.tudelft.idm.oclc.org/article/10.1088/0305-4470/23/11/012/pdf .

[11] Zeyu Deng, Abla Kammoun, and Christos Thrampoulidis. A model of double descent for high-dimensional binary linear classification. *arXiv:1911.05822*, 2020. URL: https://arxiv.org/pdf/1911.05822.pdf.

[12] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL: http://gaussianprocess.org/gpml/chapters/RW.pdf.

[13] C. Sammut and G.I. Webb. *Encyclopedia of Machine Learning*. Springer, Boston, MA, 2011. DOI: https://doi-org.tudelft.idm.oclc.org/10.1007/978-0-387-30164-8_324.

[14] Zhenyu Liao, Romain Couillet, and Michael W. Mahoney. A random matrix analysis of random fourier features: Beyond the gaussian kernel, a precise phase transition, and the corresponding double descent. *Advances in Neural Information Processing Systems*, 33:13939–13950, 2020. URL: https://arxiv.org/pdf/2006.05013v2.pdf.

[15] J. McClellan and T. Parks. Eigenvalue and eigenvector decomposition of the discrete fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 20(1):66–74, 1972. URL: https://ieeexplore-ieee-org.tudelft.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=1162342&tag=1.

[16] Fanghui Liu, Zhenyu Liao, and Johan Suykens. Kernel regression in high dimensions: Refined analysis beyond double descent. *PMLR*, 130:649–657, 2021. URL: http://proceedings.mlr.press/v130/liu21b/liu21b.pdf.

[17] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel ridgeless regression can generalize. *The Annals of Statistics*, 48:1329–2347, 2020. URL: https://arxiv.org/pdf/1808.00387.pdf.

[18] Ilja Kuzborskij, Csaba Szepesvari, Omar Rivasplata, Amal Rannen-Triki, and Razvan Pascanu. On the role of optimization in double descent: A least squares study. *Advances in Neural Information Processing Systems 34 pre-proceedings (NeurIPS 2021)*, 2021. URL: https://proceedings.neurips.cc/paper/2021/file/f754186469a933256d7d64095e963594-Paper.pdf.

[19] Arthur Jacot, Frank Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv:1806.07572*, 2018. URL: https://arxiv.org/pdf/1806.07572.pdf.

[20] Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of sgd in non-convex over-parametrized learning. *arXiv:1811.02564*, 2018. URL: https://arxiv.org/pdf/1811.02564.pdf.

[21] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parametrized non-linear systems and neural networks. *arXiv:2003.00307v2*, 2020. URL: https://arxiv.org/pdf/2003.00307.pdf.

[22] Tom Viering and Marco Loog. The shape of learning curves: a review. *arXiv:2103.10948*, 2021. URL: https://arxiv.org/pdf/2103.10948.pdf.

[23] Preetum Nakkiran. More data can hurt for linear regression: Sample-wise double descent. *arXiv:1912.07242*, 2019. URL: https://arxiv.org/pdf/1912.07242.pdf.

[24] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Hya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv:1912.02292*, 2019. URL: https://arxiv.org/pdf/1912.02292.pdf.

[25] Ouns El Harzli, Guillermo Valle-Perez, and Ard A. Louis. Double-descent curves in neural networks: a new perspective using gaussian processes. *arXiv:2102.07238*, 2021. URL: https://arxiv.org/pdf/2102.07238.pdf.

[26] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. *PMLR*, 80:541–549, 2018. URL: https://arxiv.org/pdf/1802.01396.pdf.

[27] Spencer Frei, Niladri S. Chatterji, and Peter L. Bartlett. Benign overfitting without linearity: Neural network classificers trained by gradient descent for noisy linear data. *arXiv:2202.05928*, 2022. URL: https://arxiv.org/pdf/2202.05928v2.pdf.

[28] L. Breiman and D. Freedman. How many variables should be entered in a regression equation? *Journal of the American Statistical Association*, 78(381):131–136, 1983. URL: https://www-tandfonline-com.tudelft.idm.oclc.org/doi/pdf/10.1080/01621459.1983.10477941.

[29] S. Page and S. Grunewalder. Ivanov-regularised least-squares estimators over large rkhss and their interpolation spaces. *arXiv:1706.03678*, 2017. URL: https://arxiv.org/pdf/1706.03678.pdf.

[30] Philippe Rigollet. *18.S997 High-Dimensional Statistics*. Massachusetts Institute of Technology: MIT OpenCourseWare, Spring 2015. URL: https://ocw.mit.edu.

[31] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018. DOI: 10.1017/9781108231596.

[32] Vladimir Koltchinskii and Karim Lounici. Concentration inequalities and moment bounds for sample covariance operators. *arXiv:1405.2468*, 2014. URL: https://arxiv.org/pdf/1405.2468.pdf.

9 Appendix

9.1 Proof of Theorem 1

We follow the proof as given in Belkin et al. (2020) [4], with some details added. We start with proving formula (2.1). We will distinguish between the under-parameterized ($p \leq n$) and over-parameterized ($p > n$) case.

Under-parameterized case ($p \leq n$)

In Breiman et al. (1983) [28] the situation for $p \leq n$ is discussed. They state the following Lemma.

Lemma 5 (Theorem 1.1 in [28])**.** *Define*

$$\sigma^2 = Var(\varepsilon), \qquad \sigma_p^2 = Var(\sum_{i=p+1}^{\infty} \theta_i x_i \mid x_1, \ldots, x_p), \qquad U_{n,p} := \mathbb{E}(\mathbb{E}((y - \hat{y}_{n+1})^2 \mid (y_i, x_j)_{i \leq n, j \leq n}))$$

with $\hat{y}_{n+1}$ the estimate for y after training with a training set of size n. If $p \leq n-2$, then

$$U_{n,p} = (\sigma^2 + \sigma_p^2)\left(1 + \frac{p}{n-p-1}\right).$$

For $n \geq p \geq n-1$ we have $U_{n,p} = \infty$.

We can apply Lemma 5 with some simplifications. Since $x \sim N(0, I_p)$, the components of x are IID $N(0,1)$, so $x_i \sim N(0,1)$. In this case, we have

$$\sigma_p^2 = \mathrm{Var}(\sum_{i=p+1}^{\infty} \theta_i x_i | x_1, \ldots, x_p) = \mathrm{Var}(\sum_{i \in P^c} \theta_i x_i) = \sum_{i \in P^c} \theta_i^2 \mathrm{Var}(x_i) = \sum_{i \in P^c} \theta_i^2 = ||\theta_{P^c}||^2.$$

Furthermore

$$U_{n,p} = \mathbb{E}(\mathbb{E}((y - \hat{y}_{n+1})^2 | (y_i, x_j)_{i \leq n, j \leq n})) = \mathbb{E}(y - \hat{y})^2 = \mathbb{E}(y - x^T\hat{\theta})^2 =: R(\hat{\theta}).$$

Hence we find

$$R(\hat{\theta}) = \begin{cases} (||\theta_{P^c}||^2 + \sigma^2)(1 + \frac{p}{n-p-1}) & \text{if } p \leq n-2 \\ \infty & \text{if } n-1 \leq p \leq n \end{cases}$$

Over-parameterized case ($p > n$)

Using that $x \sim N(0, I_d)$, we can rewrite the prediction risk of $\hat{\theta}$ as follows

$$\mathbb{E}(y - x^T\hat{\theta})^2 = \mathbb{E}(x^T(\theta - \hat{\theta}) + \varepsilon)^2 = \mathbb{E}(x^T(\theta - \hat{\theta}))^2 + 2\mathbb{E}(x^T(\theta - \hat{\theta})\varepsilon) + \mathbb{E}(\varepsilon^2)$$

$$= \mathbb{E}((\theta - \hat{\theta})^T x x^T (\theta - \hat{\theta})) + 0 + \sigma^2 = (\theta - \hat{\theta})^T I_d (\theta - \hat{\theta}) + \sigma^2 = ||\theta - \hat{\theta}||^2 + \sigma^2$$

$$= \sigma^2 + ||\theta_{P^c} - \hat{\theta}_{P^c}||^2 + ||\theta_P - \hat{\theta}_P||^2.$$

Furthermore, since $\hat{\theta}_{P^c} = 0$, we obtain

$$\mathbb{E}(y - x^T\hat{\theta})^2 = \sigma^2 + ||\theta_{P^c}||^2 + \mathbb{E}(||\theta_P - \hat{\theta}_P||^2).$$

For $p > n$ we know that the pseudo-inverse $X_P^\dagger$ is given by $X_P^\dagger = X_P^T(X_P X_P^T)^{-1}$. Set $\eta := y - X_P\theta_P$. Then

$$\theta_P - \hat{\theta}_P = \theta_P - X_P^T(X_P X_P^T)^{-1}y = \theta_P - X_P^T(X_P X_P^T)^{-1}(X_P\theta_P + \eta)$$
$$= (I - X_P^T(X_P X_P^T)^{-1}X_P)\theta_P - X_P^T(X_P X_P^T)^{-1}\eta.$$

Here the term $(I - X_P^T(X_P X_P^T)^{-1}X_P)\theta_P$ is an element of the null space of X_P, $N(X_P)$. Indeed, using that $v \in N(X_P)$ if and only if $X_P v = 0$, we find

$$X_P(I - X_P^T(X_P X_P^T)^{-1}X_P)\theta_P = X_P\theta_P - X_P X_P^T(X_P X_P^T)^{-1}X_P\theta_P = X_P\theta_P - X_P\theta_P = 0.$$

Furthermore, $-X_P^T(X_P X_P^T)^{-1}\eta$ is a vector in the row space of X_P, $R(X_P)$. Indeed, we know $v \in \mathrm{R}(X_P)$ if and only if there exists a vector w s.t.

$$X_P^T w = v = -X_P^T(X_P X_P^T)^{-1}\eta,$$

so we should take $w = -(X_P X_P^T)^{-1}\eta$. Since $R(X_P) \perp N(X_P)$, we have by the Pythagorean theorem

$$||\theta_P - \hat{\theta}_P||^2 = ||(I - X_P^T(X_P X_P^T)^{-1}X_P)\theta_P||^2 + ||X_P^T(X_P X_P^T)^{-1}\eta||^2.$$

First term: $||(I - X_P^T(X_P X_P^T)^{-1}X_P)\theta_P||^2$. Set $\Pi_P = X_P^T(X_P X_P^T)^{-1}X_P$. Notice that $\Pi_P := X_P^T(X_P X_P^T)^{-1}X_P$ is the orthogonal projection matrix for the row space of X_P. Indeed,

$$\Pi_P\Pi_P = X_P^T(X_P X_P^T)^\dagger X_P X_P^T(X_P X_P^T)^\dagger X_P = X_P^T(X_P X_P^T)^\dagger X_P = \Pi_P,$$

and

$$\Pi_P^T = (X_P^T(X_P X_P^T)^\dagger X_P)^T = X_P^T(X_P X_P^T)^\dagger X_P = \Pi_P.$$

So Π_P is an orthogonal projection matrix. Hence $\Pi_P\theta_P \perp \theta_P$, and by the Pythagorean theorem

$$||(I - \Pi_P)\theta_P||^2 = ||\theta_P||^2 + ||\Pi_P\theta_P||^2.$$

One of the properties of the multivariate standard normal distribution is the rotational symmetry, as its pdf only depends on the distance to the origin. Since Π_P is a projection matrix, we can write

$$\Pi_P = \sum_{i=1}^n v_i v_i^T,$$

with v_i the eigenvectors of Π_P. Here we assume that $\mathrm{rank}(\Pi_P) = n$, which is indeed the case as $\mathrm{rank}(X_P) = n$. Choose the eigenvectors such that $v_i = e_i \in \mathbb{R}^p$ (i-th unit vector). Because of the rotational symmetry, we can rotate the eigenvectors without changing the expected value. So setting $v_i = Re_i$, we get

$$\Pi_P = \sum_{i=1}^n (Re_i)(Re_i)^T = R\sum_{i=1}^n e_i e_i^T R^T.$$

Then

$$\mathbb{E}||\Pi_P\theta_P||^2 = \mathbb{E}\langle\theta, \Pi_P\theta_P\rangle = \mathbb{E}\theta_P^T\left[R(\sum_{i=1}^n e_i e_i^T)R^T\right]\theta_P = \mathbb{E}\sum_{i=1}^n (R^T\theta_P)^T e_i e_i^T (R^T\theta_P)$$

$$= \mathbb{E}\sum_{i=1}^n \langle e_i, R\theta_P\rangle^2 = \mathbb{E}\sum_{i=1}^n (R\theta_P)_i^2 = \mathbb{E}\frac{n}{p}\sum_{i=1}^p (R\theta_P)_i^2$$

where the last equality follows from symmetry, since we can choose any combination of n out of the p unit vectors. Now, since rotating a vector does not change its norm, we have

$$\mathbb{E}\frac{n}{p}\sum_{i=1}^p (R\theta_P)_i^2 = \frac{n}{p}||R\theta_P||^2 = \frac{n}{p}||\theta_P||^2.$$

Hence, it follows that

$$\mathbb{E}||\Pi_P\theta_P||^2 = ||\theta_P||^2\frac{n}{p}.$$

Therefore

$$\mathbb{E}||(I - X_P^T(X_PX_P^T)^{-1}X_P)\theta_P||^2 = ||\theta_P||^2(1-\frac{n}{p}).$$

Second term: $||X_P^T(X_PX_P^T)^{-1}\eta||^2$. We use the trace trick: $a^Tb = \mathrm{tr}(a^Tb)$. Then

$$\begin{aligned}||X_P^T(X_PX_P^T)^{-1}\eta||^2 &= (X_P^T(X_PX_P^T)^{-1}\eta)^T(X_P^T(X_PX_P^T)^{-1}\eta)\\ &= \mathrm{tr}(X_P^T(X_PX_P^T)^{-1}\eta)^TX_P^T(X_PX_P^T)^{-1}\eta)\\ &= \mathrm{tr}(\eta^T(X_PX_P^T)^{-1}X_PX_P^T(X_PX_P^T)^{-1}\eta)\\ &= \mathrm{tr}(\eta^T(X_PX_P^T)^{-1}\eta) = \mathrm{tr}((X_PX_P^T)^{-1}\eta\eta^T).\end{aligned}$$

Next, notice that η has i-th component $y_i - x_P^T\theta_P$. We can write

$$x^T\theta = x_P^T\theta_P + x_{P^c}^T\theta_{P^c},$$

$$y_i - x_P^T\theta_P = y - x^T\theta + x_{P^c}^T\theta_{P^c},$$

$$\eta_i = \varepsilon_i + (X_{P^c}\theta_{P^c})_i.$$

Hence η_i is independent of $x_P^T\theta_P$, so for the expectation we find

$$\mathbb{E}||X_P^T(X_PX_P^T)^{-1}\eta||^2 = \mathrm{tr}(\mathbb{E}(X_PX_P^T)^{-1}\mathbb{E}(\eta\eta^T))$$

Furthermore, we know that η_i is Gaussian with mean and covariance

$$\mathbb{E}(\eta_i) = \mathbb{E}(\varepsilon_i) + \mathbb{E}(x_{P^c}^T\theta_{P^c}) = 0 + 0 = 0,$$

$$\mathrm{Var}(\eta_i) = \mathrm{Var}(\varepsilon_i) + \mathrm{Var}(x_{P^c}^T\theta_{P^c}) = \sigma^2 + ||\theta_{P^c}||^2.$$

Hence η is Gaussian with mean 0 and covariance matrix $(||\theta_{P^c}||^2 + \sigma^2)I_n$. So

$$\mathbb{E}(\eta\eta^T) = (||\theta_{P^c}||^2 + \sigma^2)I_n.$$

Next, we know that $P := (X_PX_P^T)^{-1}$ has an inverse-Wishart distribution with scale matrix I_n and p degrees of freedom. Hence

$$\mathbb{E}(P) = \frac{1}{p-n-1}I_n$$

for $p \geq n+2$ and $\mathbb{E}(P) = \infty$ for $p = n, n+1$. So

$$\mathrm{tr}\mathbb{E}(X_PX_P^T)^{-1} = \mathrm{tr}\frac{1}{p-n-1}I_n = \frac{n}{p-n-1}$$

for $p \geq n+2$. So we can conclude

$$\mathbb{E}||X_P^T(X_PX_P^T)^{-1}\eta||^2 = \begin{cases}(||\theta_{P^c}||^2 + \sigma^2)\frac{n}{p-n-1} & \text{if } p \geq n+2\\ \infty & \text{if } p = n, n+1\end{cases}$$

Combining with the first term and the case $p \leq n$, we retrieve formula (2.1).

Proof of formula (2.2)

Since P is a uniformly random subset of $\{1,\dots,d\}$ with $|P|=p$, we have for all $i\in\{1,\dots,d\}$

$$\mathbb{P}(i\in P)=\frac{p}{d}.$$

Furthermore, we can write

$$(\theta^*_P)_j=\theta^*_j\mathbb{1}\{j\in P\}.$$

Hence, we have

$$\mathbb{E}(||\theta^*_P||^2)=\sum_j\mathbb{E}((\theta^*_P)^2_j)=\sum_j\mathbb{E}((\theta^*_j)^2\mathbb{1}\{j\in P\})=\sum_j(\theta^*_j)^2\mathbb{E}(\mathbb{1}\{j\in P\})$$

$$=\sum_j(\theta^*_j)^2\mathbb{P}(j\in P)=\frac{p}{d}\sum_j(\theta^*_j)^2=\frac{p}{d}||\theta^*||^2.$$

Similarly, we have

$$\mathbb{E}(||\theta^*_{P^c}||^2)=\left(1-\frac{p}{d}\right)||\theta^*||^2.$$

Now taking expectation with respect to P of formula (2.1), we find

$$R_{rand}(\hat\theta)=\mathbb{E}((y-x^T\hat\theta)^2)=\mathbb{E}_P\mathbb{E}_{X,\varepsilon}((y-x^T\hat\theta)^2).$$

For $p\le n-2$, this results in

$$(\mathbb{E}_P(||\theta^*_{P^c}||^2)+\sigma^2)(1+\frac{p}{n-p-1})=((1-\frac{p}{d})||\theta^*||^2+\sigma^2)(1+\frac{p}{n-p-1}).$$

For $p\ge n+2$, we find

$$\mathbb{E}_P||\theta^*_P||^2(1-\frac{n}{p})+(\mathbb{E}_P||\theta^*_{P^c}||^2+\sigma^2)(1+\frac{n}{p-n-1})=\frac{p}{d}||\theta^*||^2(1-\frac{n}{p})+((1-\frac{p}{d})||\theta^*||^2+\sigma^2)(1+\frac{n}{p-n-1})$$

$$=||\theta^*||^2(\frac{p}{d}(1-\frac{n}{p})+(1-\frac{p}{d})(1+\frac{n}{p-n-1}))+\sigma^2(1+\frac{n}{p-n-1})$$

$$=||\theta^*||^2(-\frac{n}{d}+1+\frac{n}{p-n-1}-\frac{pn}{d(p-n-1)})+\sigma^2(1+\frac{n}{p-n-1}),$$

where we can write

$$-\frac{n}{d}+1+\frac{n}{p-n-1}-\frac{pn}{d(p-n-1)}=-\frac{n(p-n-1)}{d(p-n-1)}+1+\frac{nd}{d(p-n-1)}-\frac{pn}{d(p-n-1)}$$

$$=1+\frac{-np+n^2+n+nd-np}{d(p-n-1)}=\frac{dp-dn-d-np+n^2+n+nd-np}{d(p-n-1)}$$

$$=\frac{dp-d-2np+n^2+n}{d(p-n-1)}.$$

On the other hand, in formula (2.2) we have

$$1-\frac{n}{d}(2-\frac{d-n-1}{p-n-1})=1-2\frac{n}{d}+\frac{n(d-n-1)}{d(p-n-1)}=\frac{d(p-n-1)-2n(p-n-1)+n(d-n-1)}{d(p-n-1)}$$

$$=\frac{dp-dn-d-2np+2n^2+2n+nd-n^2-n}{d(p-n-1)}=\frac{dp-d-2np+n^2+n}{d(p-n-1)}.$$

So we retrieve exactly the expressions from formula (2.2).

9.2 Proof of Lemma 1

The proof of Lemma 1 can also be found in Appendix H of [5]. If $\text{rank}(\Sigma) = p$, then we immediately have

$$\text{rank}(\Sigma)s(\Sigma) = p\frac{\frac{1}{p}\sum_{i=1}^{p}\lambda_i}{\lambda_1} = \frac{\sum_{i=1}^{p}\lambda_i}{\lambda_1} = r_0(\Sigma),$$

$$\text{rank}(\Sigma)S(\Sigma) = p\frac{(\frac{1}{p}\sum_{i=1}^{p}\lambda_i)^2}{\frac{1}{p}\sum_{i=1}^{p}\lambda_i^2} = \frac{(\sum_{i=1}^{p}\lambda_i)^2}{\sum_{i=1}^{p}\lambda_i^2} = R_0(\Sigma).$$

Furthermore, using that $\lambda_1 \geq \lambda_2 \geq \ldots, \lambda_p$, we have for $s(\Sigma)$

$$s(\Sigma) = \frac{\frac{1}{p}\sum_{i=1}^{p}\lambda_i}{\lambda_1} = \frac{1}{p}\sum_{i=1}^{p}\frac{\lambda_i}{\lambda_1} \leq \frac{1}{p}\sum_{i=1}^{p}1 = 1,$$

$$s(\Sigma) = \frac{\frac{1}{p}\sum_{i=1}^{p}\lambda_i}{\lambda_1} \geq \frac{\frac{1}{p}\lambda_1}{\lambda_1} = \frac{1}{p}.$$

Similar arguments can be used to show that $\frac{1}{p} \leq S(\Sigma) \leq 1$. What remains is to show that $1 \leq r_k(\Sigma) \leq R_k(\Sigma) \leq p$. The inequality $r_k(\Sigma) \geq 1$ follows immediately from the definition of $r_k(\Sigma)$

$$r_k(\Sigma) = \frac{\sum_{i=k+1}^{p}\lambda_i}{\lambda_{k+1}} = \sum_{i=k+1}^{p}\frac{\lambda_i}{\lambda_{k+1}} = 1 + \sum_{i=k+2}^{p}\frac{\lambda_i}{\lambda_{k+1}} \geq 1.$$

Furthermore, we have

$$r_k^2(\Sigma) = \left(\frac{\sum_{i=k+1}^{p}\lambda_i}{\lambda_{k+1}}\right)^2 = \frac{(\sum_{i=k+1}^{p}\lambda_i)^2}{\lambda_{k+1}^2} = \frac{\sum_{i=k+1}^{p}\lambda_i^2}{\lambda_{k+1}^2}\frac{(\sum_{i=k+1}^{p}\lambda_i)^2}{\sum_{i=k+1}^{p}\lambda_i^2} = r_k(\Sigma^2)R_k(\Sigma),$$

and

$$r_k(\Sigma^2) = \frac{\sum_{i=k+1}^{p}\lambda_i^2}{\lambda_{k+1}^2} \leq \frac{\lambda_{k+1}\sum_{i=k+1}^{p}\lambda_{k+1}}{\lambda_{k+1}^2} = r_k(\Sigma).$$

Combining these 2 results, we obtain

$$r_k^2(\Sigma) = r_k(\Sigma^2)R_k(\Sigma) \leq r_k(\Sigma)R_k(\Sigma)$$

From this it follows that $r_k(\Sigma) \leq R_k(\Sigma)$. Finally, we have that

$$R_k(\Sigma) = \frac{(\sum_{i=k+1}^{p}\lambda_i)^2}{\sum_{i=k+1}^{p}\lambda_i^2} \leq \frac{(q-k)\sum_{i=k+1}^{p}\lambda_i^2}{\sum_{i=k+1}^{p}\lambda_i^2} = p - k \leq p,$$

which concludes the proof.

9.3 Proof of Theorem 4

We will follow the proof as given in [5], with some steps added in between for clarity. The main steps in the proof are as follows:

1. Upper bound $R_{excess}(\hat{\theta})$ in terms of θ^*, a matrix B and the trace of a matrix C. With probability at least $1 - e^{-t}$, $t \geq 0$, we have

$$R_{excess}(\hat{\theta}) \leq 2(\theta^*)^T B\theta^* + 12t\sigma^2 \mathrm{tr}(C)$$

   with B and C defined as

$$B = (I - X^T(XX^T)^{-1}X)\Sigma(I - X^T(XX^T)^{-1}X),$$

$$C = (XX^T)^{-1}X\Sigma X^T(XX^T)^{-1}.$$

2. Express trace(C) in terms of independent subgaussian vectors

$$\mathrm{tr}(C) = \sum_i \frac{\lambda_i^2 z_i^T A_{-i}^2 z_i}{(1 + \lambda_i z_i^T A_{-i}^{-1} z_i)^2}$$

   with $z_i := Xv_i/\sqrt{\lambda_i}$ independent σ_x^2 (sub-)Gaussian vectors with unit variance and $A_{-i} := \sum_{j \neq i} \lambda_j z_j z_j^T$, where λ_j is the j-th eigenvalue of Σ.

3. Find upper and lower bounds on the eigenvalues of A_{-i}. There exists a universal constant c s.t. with prob. at least $1 - 2e^{-n/c}$

$$\frac{1}{c}\sum_{i=k+1} \lambda_i - c\lambda_{k+1}n \leq \mu_n(A_k) \leq \mu_1(A_k) \leq c\sum_{i=k+1} \lambda_i + c\lambda_{k+1}n.$$

4. Upper bound trace(C). There exist $b, c \geq 1$ s.t. if $0 \leq k \leq n/c$, $r_k(\Sigma) \geq bn$ and $l \leq k$, then with prob. at least $1 - 7e^{-n/c}$

$$\mathrm{tr}(C) \leq c\left(\frac{l}{n} + n\frac{\sum_{i>l}\lambda_i^2}{(\sum_{i>k}\lambda_i)^2}\right).$$

5. Make a convenient choice for the split l. Namely, take $l = k^*$.

6. Upper bound the B term: there exists a $C > 0$ s.t. for all $t \geq 1$, with probability at least $1 - e^{-t}$

$$(\theta^*)^T B\theta^* \leq C||\theta^*||^2||\Sigma|| \max\{\sqrt{r_0(\Sigma)/n}, r_0(\Sigma)/n, \sqrt{t/n}, t/n\}.$$

7. Combining the previous steps to find the statement of Theorem 4.

Step 1: Upper bound excess risk in terms of B and C

Recall that the excess risk, evaluated in the min-norm solution $\hat{\theta}$, is defined as

$$R_{excess}(\hat{\theta}) = \mathbb{E}_{x,y}(y - x^T\hat{\theta})^2 - \mathbb{E}_{x,y}(y - x^T\theta^*)^2.$$

Plugging in $y = x^T\theta^* + \varepsilon$

$$R_{excess}(\hat{\theta}) = \mathbb{E}_{x,y}(y - x^T\hat{\theta})^2 - \mathbb{E}_{x,y}(y - x^T\theta^*)^2 = \mathbb{E}(x^T\theta^* - \varepsilon - x^T\hat{\theta})^2 - \mathbb{E}(\varepsilon^2)$$

$$= \mathbb{E}(x^T(\hat{\theta} - \theta^*))^2 - 2\mathbb{E}(x^T(\hat{\theta} - \theta^*)\varepsilon) + \mathbb{E}(\varepsilon^2) - \mathbb{E}(\varepsilon^2) = \mathbb{E}(x^T(\hat{\theta} - \theta^*))^2.$$

Notice that, conditional on $\hat{\theta}$, we have the representation

$$R_{excess}(\hat{\theta}) = \mathbb{E}(x^T(\hat{\theta} - \theta^*))^2 = \mathbb{E}(x^T(\hat{\theta} - \theta^*)x^T(\hat{\theta} - \theta^*))$$

$$= \mathbb{E}((\hat{\theta} - \theta^*)^T xx^T(\hat{\theta} - \theta^*)) = (\hat{\theta} - \theta^*)^T\Sigma(\hat{\theta} - \theta^*).$$

We will use this representation in Section 4.3 to see how the eigenvalues of Σ influence the prediction accuracy.

Next, we plug in the estimator $\hat{\theta} = X^T(XX^T)^{-1}y = X^T(XX^T)^{-1}(X\theta^* + \varepsilon)$,

$$R_{excess}(\hat{\theta}) = \mathbb{E}(x^T(\hat{\theta} - \theta^*))^2 = \mathbb{E}(x^T(X^T(XX^T)^{-1}(X\theta^* + \varepsilon) - \theta^*))^2$$

$$= \mathbb{E}(x^T(X^T(XX^T)^{-1}X)\theta^* + x^T(X^T(XX^T)^{-1}\varepsilon) - x^T\theta^*)^2$$

$$= \mathbb{E}(x^T(I - X^T(XX^T)^{-1}X)\theta^* - x^TX^T(XX^T)^{-1}\varepsilon)^2$$

$$\leq 2\mathbb{E}(x^T(I - X^T(XX^T)^{-1}X)\theta^*)^2 + 2\mathbb{E}(x^TX^T(XX^T)^{-1}\varepsilon)^2,$$

where for the last inequality we use $(a-b)^2 \leq 2(a^2+b^2)$. Next, using that $\mathrm{Var}(x^Tb) = \mathrm{Var}(b^Tx) = b^T\mathrm{Var}(x)b$ with b constant, $\mathbb{E}(x) = \mathbb{E}(\varepsilon) = 0$ and the fact that $I - X^T(XX^T)^{-1}X$ is symmetric, we find

$$R_{excess}(\hat{\theta}) \leq 2\mathrm{Var}(x^T(I - X^T(XX^T)^{-1}X)\theta^*) + 2(\mathbb{E}(x^T(I - X^T(XX^T)^{-1}X)\theta^*))^2$$

$$+2\mathrm{Var}(x^TX^T(XX^T)^{-1}\varepsilon) + 2(\mathbb{E}(x^TX^T(XX^T)^{-1}\varepsilon))^2$$

$$= 2(\theta^*)^T(I - X^T(XX^T)^{-1}X)^T\mathrm{Var}(x)(I - X^T(XX^T)^{-1}X)\theta^*$$

$$+2\varepsilon^T(X^T(XX^T)^{-1})^T\mathrm{Var}(x)X^T(XX^T)^{-1}\varepsilon$$

$$= 2(\theta^*)^T(I - X^T(XX^T)^{-1}X)\Sigma(I - X^T(XX^T)^{-1}X)\theta^*$$

$$+2\varepsilon^T(X^T(XX^T)^{-1})^T\Sigma X^T(XX^T)^{-1}\varepsilon$$

$$= 2(\theta^*)^TB\theta^* + 2\varepsilon^TC\varepsilon,$$

with

$$B := (I - X^T(XX^T)^{-1}X)\Sigma(I - X^T(XX^T)^{-1}X),$$

$$C := (XX^T)^{-1}X\Sigma X^T(XX^T)^{-1}.$$

It remains to upper bound $\varepsilon^TC\varepsilon$ in terms of the trace of C. For this, Bartlett et al. (2020) [5] use Lemma 35 from Page and Grunewalder (2017) [29], which is stated in the following Lemma.

Lemma 6 (Lemma 35 from [29]). *Let ε_i be independent random variables s.t. for all $\lambda \in \mathbb{R}$*

$$\mathbb{E}(e^{\lambda\varepsilon_i}) \leq e^{\sigma^2\lambda^2/2}$$

(so ε_i is sub-Gaussian). Let $M \subset \mathbb{R}^{n\times n}$ be a positive semi-definite matrix and let $t \geq 0$. Then with probability at least $1 - e^{-t}$ we have

$$\varepsilon^TM\varepsilon \leq \sigma^2tr(M) + 2\sigma^2||M||t + 2\sigma^2\sqrt{||M||^2t^2 + tr(M^2)t}$$

Proof. See the proof of Lemma 35 in [29].

We will use Lemma 6 with $M = C$. Notice that C is indeed positive semi-definite and ε_i is a (sub-)Gaussian random variable. So by the lemma we have, with probability at least $1 - e^{-t}$

$$\varepsilon^T C\varepsilon \leq \sigma^2 \text{tr}(C) + 2\sigma^2 ||C|| t + 2\sigma^2 \sqrt{||C||^2 t^2 + \text{tr}(C^2)t}.$$

Next, using that $\text{tr}(C^2) \leq \text{tr}(C)^2$ since C is positive semi-definite (to see this, write $\text{tr}(C) = \sum_i \lambda_i(C)$ and use $\lambda_i \geq 0$), we have

$$||C||_F = \sqrt{\text{tr}(C^T C)} = \sqrt{\text{tr}(C^2)} \leq \sqrt{(\text{tr}(C))^2} = \text{tr}(C).$$

Hence, we find, with probability at least $1 - e^{-t}$

$$\varepsilon^T C\varepsilon \leq \sigma^2 \text{tr}(C) + 2\sigma^2 \text{tr}(C)t + 2\sigma^2 \sqrt{\text{tr}(C)^2 t^2 + \text{tr}(C)^2 t}$$

$$= (2t+1)\sigma^2 \text{tr}(C) + 2\sigma^2 (\sqrt{t^2+t})\text{tr}(C) \leq (4t+2)\sigma^2 \text{tr}(C),$$

using that $\sqrt{t^2+t} \leq \sqrt{t^2+t+1/4} = t + \frac{1}{2}$ for $t \geq 0$. Thus, with probability at least $1 - e^{-t}$ we have

$$R_{excess}(\hat{\theta}) \leq 2(\theta^*)^T B\theta^* + (8t+4)\sigma^2 \text{tr}(C).$$

Assuming that $t \geq 1$, we find, with probability at least $1 - e^{-t}$

$$R_{excess}(\hat{\theta}) \leq 2(\theta^*)^T B\theta^* + 12t\sigma^2 \text{tr}(C). \tag{9.1}$$

Step 2: Express trace in terms of independent (sub-)Gaussian vectors

We have assumed that we can write $x = V\Lambda^{1/2} z$ (with V, Λ s.t. $\Sigma = V\Lambda V^T$), with z having independent σ_x^2 (sub-)Gaussian components. Hence $z_i := Xv_i/\sqrt{\lambda_i}$ has components $x^T v_i/\sqrt{\lambda_i}$ that are independent σ_x^2 (sub-)Gaussian. Writing $Xv_i = \sqrt{\lambda_i} z_i$, we then have

$$v_i^T X^T = \sqrt{\lambda_i} z_i^T, \qquad XX^T = XVV^T X^T = \sum_i \lambda_i z_i z_i^T,$$

and for $X\Sigma X^T$

$$X\Sigma X^T = XV\Lambda V^T X^T = \sum_i \lambda_i^2 z_i z_i^T.$$

Then for the trace of C we have, using that $\text{tr}(ABC) = \text{tr}(BCA)$,

$$\text{tr}(C) = \text{tr}((XX^T)^{-1} X\Sigma X^T (XX^T)^{-1}) = \text{tr}(X\Sigma X^T (XX^T)^{-2}) = \text{tr}(\sum_i \lambda_i^2 z_i z_i^T (\sum_j \lambda_j z_j z_j^T)^{-2})$$

$$= \sum_i \lambda_i^2 \text{tr}(z_i z_i^T (\sum_j \lambda_j z_j z_j^T)^{-2}) = \sum_i \lambda_i^2 z_i^T (\sum_j \lambda_j z_j z_j^T)^{-2} z_i,$$

where for the last equality we use $\text{tr}(z_i z_i^T M) = \text{tr}(z_i^T M z_i)$. If we define $A_{-i} := \sum_{j \neq i} \lambda_j z_j z_j^T$, then

$$\sum_j \lambda_j z_j z_j^T = \lambda_i z_i z_i^T + A_{-i}.$$

Next, we will use Lemma 20 from Bartlett et al. (2020) [5].

Lemma 7 (Lemma 20 in [5]). *For $k < n$, $A \in \mathbb{R}^{n \times n}$ invertible, $Z \in \mathbb{R}^{n \times k}$ s.t. $ZZ^T + A$ invertible, we have*

$$Z^T (ZZ^T + A)^{-2} Z = (I + Z^T A^{-1} Z)^{-1} Z^T A^{-2} Z (I + Z^T A^{-1} Z)^{-1}.$$

Proof. The proof consists of using the Sherman-Morrison-Woodbury formula and some basic matrix manipulations. See the proof of Lemma 20 in [5].

We apply Lemma 7 with $k = 1$, $Z = \sqrt{\lambda_i} z_i$ and $A = A_{-i}$. Note that A_{-i} is invertible: since we have assumed that $\text{rank}(\Sigma) > n$, there are at least $n+1$ eigenvectors v_j with $\lambda_j > 0$, so the matrix $A_{-i} = \sum_{j \neq i} \lambda_j z_j z_j^T$ still has rank n and hence A_{-i} is invertible. Furthermore, since $z_i z_i^T$ has non-negative eigenvalues, the matrix $z_i z_i^T + A_{-i}$ has strictly positive eigenvalues and hence is invertible. So we can apply Lemma 7

$$\text{tr}(C) = \sum_i \lambda_i^2 z_i^T (\lambda_i z_i z_i^T + A_{-i})^{-2} z_i = \sum_i \frac{\lambda_i^2 z_i^T A_{-i}^2 z_i}{(1 + \lambda_i z_i^T A_{-i}^{-1} z_i)^2}, \tag{9.2}$$

with $z_i := X v_i / \sqrt{\lambda_i}$ independent σ_x^2 (sub-)Gaussian vectors with mean 0 and unit variance.

Step 3: Lower and upper bound eigenvalues

First, define

$$A = \sum_i \lambda_i z_i z_i^T, \qquad A_{-i} = \sum_{j \neq i} \lambda_j z_j z_j^T, \qquad A_k = \sum_{i > k} \lambda_i z_i z_i^T,$$

with the $z_i \in \mathbb{R}^n$ independent σ_x^2 (sub-)Gaussian with unit variance.

Recall that $z_i = X v_i / \sqrt{\lambda_i}$ is a random vector with independent σ_x^2 (sub-)Gaussian and unit variance components. Then for a constant vector $v \in \mathbb{R}^n$, we have that $v^T z_i$ is $||v||^2 \sigma_x^2$ (sub-)Gaussian, which easily follows using the independence of the components of z_i.

Next, we will apply Lemma 1.12 from the lecture notes by Rigollet [30].

Lemma 8 (Lemma 1.12 in [30]). *If X is σ^2 sub-Gaussian, then $Z := X^2 - \mathbb{E}(X^2)$ is $16\sigma^2$ sub-Exponential, that is*

$$\mathbb{E}(e^{sZ}) \leq e^{(16\sigma^2)^2 s^2/2} = e^{128\sigma^4 s^2}$$

for all $|s| \leq \frac{1}{\lambda}$.

Proof. See the proof of Lemma 1.12 in [30].

Notice that $\mathbb{E}((v^T z_i)^2) = \text{Var}(v^T z_i) = \sum_j v_j^2 \text{Var}((z_i)_j) = ||v||_2^2$. So Lemma 8 implies that the random variable $(v^T z_i)^2 - ||v||_2^2$ is sub-Exponential with constant $16\sigma_x^2 ||v||_2^2$. For a unit vector v, this means that: $(v^T z_i)^2 - 1$ is $16\sigma_x^2$ sub-Exponential. Furthermore

$$v^T A v = \sum_i \lambda_i v^T z_i z_i^T v = \sum_i \lambda_i (v^T z_i)^2.$$

Then

$$|v^T A v - \sum_i \lambda_i| = |\sum_i \lambda_i (v^T z_i)^2 - \sum_i \lambda_i| = |\sum_i \lambda_i ((v^T z_i)^2 - 1)| = |\sum_i \lambda_i X_i|,$$

with $X_i := (v^T z_i)^2 - 1$, sub-Exponential$(16\sigma_x^2)$. Next, we apply Theorem 2.8.2 from [31].

Theorem 7 (Theorem 2.8.2 in [31]). *Let $X_1, \ldots, X_n$ be independent, mean-0, sub-exponential(K) random variables and let $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$. Then*

$$\mathbb{P}\left(|\sum_{i=1}^n a_i X_i| \geq t \right) \leq 2 \exp\left(-c \min\left(\frac{t^2}{K^2 ||a||_2^2}, \frac{t}{K ||a||_\infty} \right) \right)$$

where $c > 0$ is a universal constant and $K = \max_i ||X_i||_{\Psi_1}$.

Proof. See the proof in [31].

Notice that, in case $a = (\lambda_1, \ldots, \lambda_n)$, we have that $||a||_2^2 = \sum_i \lambda_i^2$ and $||a||_\infty = \lambda_1$. So by Theorem 7 for $X_i := (v^T z_i)^2 - 1$ sub-Exponential(K) we have

$$\mathbb{P}\left(|\sum_{i=1}^n \lambda_i X_i| \geq t \right) \leq 2\exp\left(-c\min(\frac{t^2}{K^2 \sum_i \lambda_i^2}, \frac{t}{K\lambda_1}) \right).$$

We can rewrite this as

$$\mathbb{P}\left(|\sum_i \lambda_i X_i| \leq t \right) \geq 1 - 2\exp(-x),$$

where we choose x such that

$$x = c\min\left(\frac{t^2}{K^2 \sum_i \lambda_i^2}, \frac{t}{K\lambda_1^2} \right).$$

If this minimum is equal to $\frac{t^2}{K^2 \sum_i \lambda_i^2}$, then

$$t = \sqrt{c} K \sqrt{x \sum_i \lambda_i^2}.$$

If this minimum is equal to $\frac{t}{K\lambda_1^2}$, then

$$t = cKx\lambda_1.$$

Hence, if we take

$$t = (\sqrt{c} + c)K \max\left(\sqrt{x \sum_i \lambda_i^2}, x\lambda_1 \right) = c_1 K \max(\sqrt{x}||\lambda||_2, x||\lambda||_\infty),$$

then there exists a constant $C > 0$ s.t. with probability at least $1 - 2e^{-t}$

$$|\sum_i \lambda_i X_i| \leq CK \max(\sqrt{t}||\lambda||_2, t||\lambda||_\infty). \tag{9.3}$$

Hence, with probability at least $1 - 2e^{-t}$

$$|v^T A v - \sum_i \lambda_i| \leq c\sigma_x^2 \max\left(\sqrt{t}\sqrt{\sum_i \lambda_i^2}, t\lambda_1 \right), \tag{9.4}$$

where we have chosen $X_i = (v^T z_i)^2 - 1$, which is sub-Exponential $(16\sigma_x^2)$. Next, we use an ε-net argument. Consider the following lemma.

Lemma 9 (Lemma 25 in [5]). *Suppose $A \in \mathbb{R}^{n \times n}$ is symmetric and N_ε is an ε-net on the unit sphere S^{n-1}, with $\varepsilon < 1/2$, then*

$$||A|| \leq (1 - \varepsilon)^{-2} \max_{x \in N_\varepsilon} |x^T A x|.$$

Proof. See the proof in [5].

We restrict ourselves to an ε-net on the unit sphere S^{n-1} s.t. $|N| \leq 9^n$. We can achieve this by choosing $\varepsilon = \frac{1}{4}$. This follows from Corollary 4.2.13 from Vershynin (2018) [31].

Corollary 1 (Corollary 4.2.13 in [31]). *The covering numbers of the unit Euclidean ball B_2^n satisfy for any $\varepsilon > 0$*

$$(1/\varepsilon)^n \leq N(B_2^n, \varepsilon) \leq (2/\varepsilon + 1)^n.$$

Here $N(B_2^n, \varepsilon)$ is the smallest possible cardinality of an ε-net of B_2^n. The same upper bound holds for the unit Euclidean sphere S^{n-1}.

Proof. See the proof in [31].

If we choose $\varepsilon = 1/4$, then by Corollary 1, we can cover the unit sphere S^{n-1} using an ε-net with cardinality at most 9^n. Next, we use this ε-net combined with a union bound argument to find an upper bound on $||A - I_n \sum_i \lambda_i||$.

Let $v_j \in N_\varepsilon$ and define the event A_j as

$$A_j := \left\{ v_j^T A v_j \geq c \max\left(\lambda_1 t, \sqrt{t \sum_i \lambda_i^2} \right) \right\}.$$

Notice that $\mathbb{P}(A_j) \leq 2e^{-t}$ by equation (9.4). We have the following union bound

$$P\left(\bigcup_{j:v_j \in N} A_j \right) \leq \sum_{j:v_j \in N} \mathbb{P}(A_j) \leq 9^n \max_j \mathbb{P}(A_j) \leq 2e^{n\log(9)} e^{-t} = 2e^{-(t - n\log(9))} = 2e^{-\tilde{t}},$$

where $\tilde{t} = t - n\log(9)$ (so $t = \tilde{t} + n\log(9)$). Hence, for all $v \in N_\varepsilon$, with probability at least $1 - 2e^{-t}$, we have

$$|v^T A v - \sum_i \lambda_i| \leq c\sigma_x^2 \max\left(\lambda_1(t + n\log(9)), \sqrt{(t + n\log(9)) \sum_i \lambda_i^2} \right) \leq c\sigma_x^2 \max\left(\lambda_1 n, \sqrt{n \sum_i \lambda_i^2} \right),$$

where we assume that $t \leq n/c_0$ for some $c_0 \geq 1$ such that $t + n\log(9) \leq (1/c_0 + \log(9))n = \tilde{c}n$. Using the ε-net, we then have, with probability at least $1 - 2e^{-t}$

$$||A - I_n \sum_i \lambda_i|| \leq (1 - \varepsilon)^{-2} \max_{v \in N_\varepsilon} |v^T A v - \sum_i \lambda_i (v^T v)| \leq c \max_{v \in N_\varepsilon} |v^T A v - \sum_i \lambda_i|$$

$$\leq c\sigma_x^2 \max\left(\sqrt{n} \sqrt{\sum_i \lambda_i^2}, n\lambda_1 \right) \qquad \text{(by equation (9.4))}$$

$$\leq c\sigma_x^2 \left(\sqrt{n\lambda_1 \sum_i \lambda_i} \right) + c\sigma_x^2 n\lambda_1.$$

Recall the AM-GM inequality

$$(x_1 \cdot x_2 \cdots x_n)^{1/n} \leq 1/n(x_1 + x_2 + \ldots + x_n).$$

Applying this with $n = 2$ and $x_1 = n$, $x_2 = \sum_i \lambda_i$, we get

$$||A - I_n \sum_i \lambda_i|| \leq c\sigma_x^2(\lambda_1 n + \sum_i \lambda_i) + c\sigma_x^2 n\lambda_1 \leq c\sigma_x^2 \lambda_1 n + c\sigma_x^2 \sum_i \lambda_1.$$

So, choosing $C = c\sigma_x^2$, with probability $1 - 2e^{-t}$ we have

$$||A - I_n \sum_i \lambda_i|| \leq C(\lambda_1 n + \sum_i \lambda_i).$$

So for the j-th eigenvalue of A, $\mu_j(A)$, we have

$$|\mu_j(A) - \sum_i \lambda_i| \leq C(\lambda_1 n + \sum_i \lambda_i).$$

Using the definition of the absolute value, this implies

$$-C(\lambda_1 n + \sum_i \lambda_i) \leq \mu_j(A) - \sum_i \lambda_i \leq C(\lambda_1 n + \sum_i \lambda_i),$$

$$-C\lambda_1 n - C\sum_i \lambda_i + \sum_i \lambda_i \leq \mu_j(A) \leq C\lambda_1 n + C\sum_i \lambda_i + \sum_i \lambda_i,$$
$$(1-C)\sum_i \lambda_i - C\lambda_1 \leq \mu_n(A) \leq \mu_1(A) \leq (1+C)\sum_i \lambda_i + C\lambda_1 n$$

Now take c_2 as

$$c_2 = \max\{1/(1-C), 1+C\}.$$

Then $1+C \leq c_2$, $C < c_2$, $-C > -c_2$ and $1-C \geq 1/c_2$. Hence, there exists a constant c_2 s.t. with probability at least $1-2e^{-t}$ (for $t < n/c_2$)

$$\frac{1}{c}\sum_i \lambda_i - c\lambda_1 n \leq \mu_n(A) \leq \mu_1(A) \leq c\left(\sum_i \lambda_i + \lambda_1 n\right).$$

We can repeat the arguments above but now for the matrix $A_k = \sum_{i=k+1} \lambda_i z_i z_i^T$ by removing the first k eigenvalues. We can state this result in the following lemma.

Lemma 10 (Lemma 9 in [5])**.** *If $\mu_1(A_k) \geq \ldots \geq \mu_n(A_k)$ are the eigenvalues of A_k, then there exists a constant $c > 1$ such that with probability at least $1-2e^{-n/c}$ we have*

$$\frac{1}{c}\sum_{i=k+1} \lambda_i - c\lambda_{k+1} n \leq \mu_n(A_k) \leq \mu_1(A_k) \leq c\sum_{i=k+1} \lambda_i + c\lambda_{k+1} n.$$

Using this result, we can prove the following lemma.

Lemma 11 (Lemma 10 in [5])**.** *There exist constants $b, c \geq 1$ s.t. with prob. at least $1-2e^{-n/c}$ we have:*
(1) for all $i \geq 1$

$$\mu_{k+1}(A_{-i}) \leq \mu_{k+1}(A) \leq \mu_1(A_k) \leq c\sum_{j=k+1} \lambda_j + c\lambda_{k+1} n,$$

(2) for all $1 \leq i \leq k$

$$\mu_n(A) \geq \mu_n(A_{-i}) \geq \mu_n(A_k) \geq \frac{1}{c}\sum_{j=k+1} \lambda_j - c\lambda_{k+1} n,$$

(3) if $r_k(\Sigma) \geq bn$, then

$$\frac{1}{c}\lambda_{k+1} r_k(\Sigma) \leq \mu_n(A_k) \leq \mu_1(A_k) \leq c\lambda_{k+1} r_k(\Sigma).$$

Proof. We will prove statements (1), (2) and (3) separately.

Proof of (1).
By Lemma 10 we immediately have

$$\mu_1(A_k) \leq c\sum_{j=k+1} \lambda_j + c\lambda_{k+1} n$$

So it remains to show that $\mu_{k+1}(A_{-i}) \leq \mu_{k+1}(A) \leq \mu_1(A_k)$. The matrix $A - A_k$ has rank at most k, so for all v in the null space of $A - A_k$

$$v^T A v = v^T(A_k + (A - A_k))v = v^T A_k v + v^T(A - A_k)v = v^T A_k v \leq \mu_1(A_k)||v||^2.$$

Since the eigenvector v_{k+1} corresponding to eigenvalue $\mu_{k+1}(A)$ is in the null space of $A - A_k$, we have that

$$\mu_{k+1}(A)||v_{k+1}||^2 = v_{k+1}^T \mu_{k+1}(A) v_{k+1} = v_{k+1}^T A v_{k+1} \leq \mu_1(A_k)||v_{k+1}||^2.$$

Hence $\mu_{k+1}(A) \leq \mu_1(A_k)$. Furthermore, we have

$$\mu_{k+1}(A) = \mu_{k+1}(A_{-i} + \lambda_i z_i z_i^T) = \mu_{k+1}(A_{-i}) + \mu_{k+1}(\lambda_i z_i z_i^T) \geq \mu_{k+1}(A_{-i}).$$

Proof of (2).
Similar argument as above, write

$$\mu_n(A) = \mu_n(A_{-i} + \lambda_i z_i z_i^T) = \mu_n(A_{-i}) + \mu_n(\lambda_i z_i z_i^T) \geq \mu_n(A_{-i})$$

and for $i \leq k$

$$\mu_n(A_{-i}) = \mu_n(A_k) + (A_{-i} - A_k)) = \mu_n(A_k) + \mu_n(A_{-i} - A_k) \geq \mu_n(A_k)$$

using that $A_{-i} - A_k$ is positive (semi-) definite.

Proof of (3)
By Lemma 10 we have

$$\frac{1}{c}\sum_{i=k+1} \lambda_i - c\lambda_{k+1}n \leq \mu_n(A_k) \leq \mu_1(A_k) \leq c\sum_{i=k+1} \lambda_i + c\lambda_{k+1}n$$

Recall that

$$r_k(\Sigma) = \frac{\sum_{i=k+1} \lambda_i}{\lambda_{k+1}}, \qquad \text{or} \qquad \sum_{i=k+1} \lambda_i = \lambda_{k+1} r_k(\Sigma).$$

Hence

$$\frac{1}{c}\lambda_{k+1} r_k(\Sigma) - c\lambda_{k+1}n \leq \mu_n(A_k) \leq \mu_1(A_k) \leq c\lambda_{k+1} r_k(\Sigma) + c\lambda_{k+1}n,$$

$$\lambda_{k+1}(\frac{1}{c} r_k(\Sigma) - cn) \leq \mu_n(A_k) \leq \mu_1(A_k) \leq \lambda_{k+1}(c r_k(\Sigma) + cn).$$

For the upper bound we have, using that $r_k(\Sigma) \geq bn$ (so $n \leq r_k(\Sigma)/b$),

$$\lambda_{k+1}(c r_k(\Sigma) + cn) \leq \lambda_{k+1}(c r_k(\Sigma) + c r_k(\Sigma)/b) = \lambda_{k+1} r_k(\Sigma) c(1 + 1/b).$$

For the lower bound we find

$$\lambda_{k+1}(r_k(\Sigma)/c - cn) \geq \lambda_{k+1}(r_k(\Sigma)/c - c r_k(\Sigma)/b) = \lambda_{k+1} r_k(\Sigma)(1/c - c/b).$$

We want $1/c - c/b$ to be larger than 0, so we should take $b > c^2$. Choose $c_1 = \max\{(1+1/b)c, 1/(1/c - c/b)\}$. Then

$$1/c_1 \lambda_{k+1} r_k(\Sigma) \leq \mu_n(A_k) \leq \mu_1(A_k) \leq c_1 \lambda_{k+1} r_k(\Sigma).$$

Step 4: Upper bound on the trace of C

Using the results of Lemma 11, we will show that there exist constants $b, c \geq 1$ s.t. if $0 \leq k \leq n/c$, $r_k(\Sigma) \geq bn$ and $l \leq k$, then with probability at least $1 - 7e^{-n/c}$, we have

$$\mathrm{tr}(C) \leq c\left(\frac{l}{n} + n\frac{\sum_{i=l+1} \lambda_i^2}{(\sum_{i=k+1} \lambda_i)^2}\right).$$

Recall from equation (9.2) that

$$\mathrm{tr}(C) = \sum_{i=1}^{p} \frac{\lambda_i^2 z_i^T A_{-i}^{-2} z_i}{(1 + \lambda_i z_i^T A_{-i}^{-1} z_i)^2}.$$

We can split this sum into 2 terms as

$$\mathrm{tr}(C) = \sum_{i=1}^{l} \frac{\lambda_i^2 z_i^T A_{-i}^{-2} z_i}{(1 + \lambda_i z_i^T A_{-i}^{-1} z_i)^2} + \sum_{i>l} \lambda_i^2 z_i^T A^{-2} z_i, \tag{9.5}$$

where $l \leq k$ and k s.t. $r_k(\Sigma) \geq bn$. Fix b to be the same value as in Lemma 11.

First term. Consider the sum up to l in equation (9.5). By Lemma 11, if $r_k(\Sigma) \geq bn$, then with probability at least $1 - 2e^{-n/c_1}$, for al $i \leq k$

$$\mu_n(A_{-i}) \geq \mu_n(A_k) \geq 1/c_1\lambda_{k+1}r_k(\Sigma).$$

This implies that

$$z_i^T A_{-i}^{-2} z_i \leq z_i^T \mu_1(A_{-i}^{-2}) z_i = \frac{z_i^T z_i}{\mu_n(A_{-i})^2} \leq \frac{c_1^2 ||z_i||^2}{\lambda_{k+1}^2 r_k(\Sigma)^2}$$

Also, by Lemma 11, with probability at least $1 - 2e^{-n/c_1}$, for all i

$$\mu_{k+1}(A_{-i}) \leq \mu_1(A_k) \leq c_1\lambda_{k+1}r_k(\Sigma).$$

This implies

$$z_i^T A_{-i}^{-1} z_i \geq (\Pi_{S_i} z_i)^T A_{-i}^{-1} \Pi_{S_i} z_i \geq (\Pi_{S_i} z_i)^T \mu_{n-k}(A_{-i}^{-1}) \Pi_{S_i} z_i = \frac{(\Pi_{S_i} z_i)^T \Pi_{S_i} z_i}{\mu_{k+1}(A_{-i})} \geq \frac{||\Pi_{S_i} z_i||^2}{c_1\lambda_{k+1}r_k(\Sigma)},$$

where S_i is the span of the $n-k$ eigenvectors of A_{-i} corresponding to its $n-k$ smallest eigenvalues, so that $\Pi_{S_i} z = 0$ for all eigenvectors z corresponding to one of $\{\mu_1(A_{-i}), \ldots, \mu_k(A_{-i}\}$. Putting these results together, we have, for $i \leq l$, with probability at least $1 - 2e^{-t}$ $(t \leq n/c_1)$

$$\frac{\lambda_i^2 z_i^T A_{-i}^{-2} z_i}{(1 + \lambda_i z_i^T A_{-i}^{-1} z_i)^2} \leq \frac{z_i^T A_{-i}^{-2} z_i}{(z_i^T A_{-i}^{-1} z_i)^2} \leq c_1^4 \frac{||z_i||^2}{||\Pi_{S_i} z_i||^4}.$$

We will now use the following Corollary.

Corollary 2 (Corollary 13 in [5])**.** *For z centered r.v. with independent σ^2-subgaussian coordinates with unit variances, S is a random subspace of $\mathbb{R}^n$ of co-dimension k and S independent of z. Then there exists a universal constant a s.t., with prob. at least $1 - 3e^{-t}$*

$$||z||^2 \leq n + a\sigma^2(t + \sqrt{nt}), \qquad ||P_S z||^2 \geq n - a\sigma^2(k + t + \sqrt{nt}),$$

where P_S is the orthogonal projection on S.

Proof. See the proof in [5].

Applying Corollary 2, we find, making use of union bounds, with probability at least $1 - 3e^{-t}$

$$\begin{aligned} ||z_i||^2 &\leq n + a\sigma^2 \left(t + \log(k) + \sqrt{n(t + \log(k))}\right) \\ &\leq n + a\sigma^2(n/C + n/C + \sqrt{nt} + \sqrt{n\log(k)}) \\ &\leq n + a\sigma^2(n/C + n/C + \sqrt{n^2/C} + \sqrt{n^2/C}) \\ &\leq (1 + 2a\sigma^2(1/C + 1/\sqrt{C}))n = c_2 n, \end{aligned}$$

assuming that $t < n/C$ and also $k < n/C$, and with probability at least $1 - 3e^{-t}$

$$\begin{aligned} ||\Pi_{S_i} z_i||^2 &\geq n - a\sigma^2(k + t + \log(k) + \sqrt{n(t + \log(k))}) \\ &\geq n - a\sigma^2(n/C + n/C + n/C + \sqrt{n^2/C} + \sqrt{n^2/C}) \\ &\geq n - a\sigma^2(3n/C + 2n/\sqrt{C}) \\ &\geq (1 - a\sigma^2(3/C + 2/\sqrt{C}))n = c_3 n, \end{aligned}$$

assuming $k < n/C$ and where we take $C > 1$ large enough such that $a\sigma^2 + 3/C + 2/\sqrt{C} \leq 1$. This means that we should take C such that

$$C \geq \left(\frac{5}{1 - a\sigma^2}\right)^2.$$

So we now have, with probability at least $1-2e^{-t}$ the event E_1 happens, where

$$E_1 = \left\{ \frac{\lambda_i^2 z_i^T A_{-i}^{-2} z_i}{(1+\lambda_i z_i^T A_{-i}^{-1} z_i)^2} \leq c_1^4 \frac{||z_i||^2}{||\Pi_{S_i} z_i||^4} \right\},$$

and with probability at least $1-3e^{-t}$ the event E_2 happens, where

$$E_2 = \left\{ ||z_i||^2 \leq c_2 n, ||\Pi_{S_i} z_i||^2 \geq c_3 n \right\}.$$

Then, using basic probability calculations

$$\mathbb{P}(E_1^c) \leq 2e^{-t}, \qquad \mathbb{P}(E_2^c) \leq 3e^{-t},$$

$$\mathbb{P}(E_1^c \cup E_2^c) \leq \mathbb{P}(E_1^c) + \mathbb{P}(E_2^c) = 5e^{-t},$$

$$\mathbb{P}(E_1 \cap E_2) = \mathbb{P}((E_1^c \cup E_2^c)^c) = 1 - \mathbb{P}(E_1^c \cup E_2^c) = 1 - 5e^{-t}.$$

Therefore, with probability at least $1-5e^{-t}$, we have

$$\frac{\lambda_i^2 z_i^T A_{-i}^{-2} z_i}{(1+\lambda_i z_i^T A_{-i}^{-1} z_i)^2} \leq c_1^4 \frac{c_2 n}{(c_3 n)^2} = \frac{c_1^4 c_2}{c_3} \frac{1}{n} =: c_4 \frac{1}{n},$$

where $t < \min(n/c_1, n/C) = n/C$ for C large enough as defined before. Hence, with probability at least $1-5e^{-n/C}$ we have

$$\sum_{i=1}^{l} \frac{\lambda_i^2 z_i^T A_{-i}^{-2} z_i}{(1+\lambda_i z_i^T A_{-i}^{-1} z_i)^2} \leq c_1^4 \frac{c_2 n}{(c_3 n)^2} \leq c_4 \frac{l}{n}.$$

Second term. Next, we look at the second term $\sum_{i>l} \lambda_i^2 z_i^T A^{-2} z_i$ from equation (9.5). Similar as with the first term, by Lemma 11, we have with probability at least $1-2e^{-n/c_1}$ that $\mu_n(A) \geq \mu_n(A_{-i}) \geq \mu_n(A_k) \geq \lambda_{k+1} r_k(\Sigma)/c_1$, provided that $r_k(\Sigma) \geq bn$. Therefore, using similar arguments as before, with probability at least $1-2e^{-t}$ ($t < n/c_1$)

$$z_i^T A^{-2} z_i \leq \frac{c_1^2 ||z_i||^2}{(\lambda_{k+1} r_k(\Sigma))^2}, \qquad \text{and so} \qquad \sum_{i>l} \lambda_i^2 z_i^T A^{-2} z_i \leq \frac{c_1^2 \sum_{i>l} \lambda_i^2 ||z_i||^2}{(\lambda_{k+1} r_k(\Sigma))^2}$$

Since z_i has independent sub-Gaussian components, we know that $\sum_{i>l} \lambda_i^2 ||z_i||^2$ is a sum of σ^2 sub-exponential random variables with weights λ_i^2. Then by equation (9.3), with prob. at least $1-2e^{-t}$

$$\begin{aligned}
\sum_{i>l} \lambda_i^2 ||z_i||^2 &\leq n \sum_{i>l} \lambda_i^2 + a\sigma^2 \max\left(\lambda_{l+1}^2 t, \sqrt{tn \sum_{i>l} \lambda_i^4} \right) \\
&\leq n \sum_{i>l} \lambda_i^2 + a\sigma^2 \max(t \sum_{i>l} \lambda_i^2, \sqrt{tn} \sum_{i>l} \lambda_i^2) \\
&\leq (n + a\sigma^2(t + \sqrt{tn})) \sum_{i>l} \lambda_i^2 \\
&\leq (n + a\sigma^2(n/C + n/\sqrt{C})) \sum_{i>l} \lambda_i^2 \\
&\leq (1 + a\sigma^2(1/C + 1/\sqrt{C}))n \sum_{i>l} \lambda_i^2 = c_5 n \sum_{i>l} \lambda_i^2,
\end{aligned}$$

using that $t < n/C$. So we find, with probability at least $1-2e^{-t}$

$$\sum_{i>l} \lambda_i^2 z_i^T A^{-2} z_i \leq c_1^2 c_5 n \frac{\sum_{i>l} \lambda_i^2}{(\lambda_{k+1} r_k(\Sigma))^2} = c_6 n \frac{\sum_{i>l} \lambda_i^2}{(\lambda_{k+1} r_k(\Sigma))^2}.$$

So combining the results for the first and second term of (9.5), we finally have, with probability at least $1-(2e^{-t}+5e^{-t})=1-7e^{-t}$

$$\mathrm{tr}(C)=\sum_{i=1}^{l}\frac{\lambda_i^2 z_i^T A_{-i}^{-2} z_i}{(1+\lambda_i z_i^T A_{-i}^{-1} z_i)^2}+\sum_{i>l}\lambda_i^2 z_i^T A^{-2} z_i \le c_4\frac{l}{n}+c_6 n\frac{\sum_{i>l}\lambda_i^2}{(\lambda_{k+1}r_k(\Sigma))^2}=c_7\left(\frac{l}{n}+n\frac{\sum_{i>l}\lambda_i^2}{(\lambda_{k+1}r_k(\Sigma))^2}\right),$$

where $c_7=c_4+c_6$. Taking $c=\max(c_7,C)$, we conclude that there exist $b,c\ge 1$ s.t. if $0\le k\le n/c$, $r_k(\Sigma)\ge bn$ and $l\le k$, then with prob. at least $1-7e^{-n/c}$

$$\mathrm{tr}(C)\le c\left(\frac{l}{n}+n\frac{\sum_{i>l}\lambda_i^2}{(\sum_{i>k}\lambda_i)^2}\right). \tag{9.6}$$

Step 5: Convenient choice of split l

The choice of the split l in 9.6 is arbitrary (as long as $l\le k$). Hence, to obtain the sharpest bound, we take the minimum over all $l\le k$

$$\mathrm{tr}(C)\le c\min_{l\le k}\left(\frac{l}{n}+n\frac{\sum_{i>l}\lambda_i^2}{(\lambda_{k+1}r_k(\Sigma))^2}\right),$$

for $k\le n/c$ s.t. $r_k(\Sigma)\ge bn$. We will prove the following Lemma.

Lemma 12 (Lemma 17 in [5])**.** *For any $b\ge 1$ and $k^*=\min\{k: r_k(\Sigma)\ge bn\}$. If $k^*<\infty$, we have*

$$\min_{l\le k^*}\left(\frac{l}{bn}+\frac{bn\sum_{i>l}\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}\right)=\frac{k^*}{bn}+\frac{bn\sum_{i>k^*}\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}=\frac{k^*}{bn}+\frac{bn}{R_{k^*}(\Sigma)}.$$

Proof.
We can write (using $l\le k^*$)

$$\frac{l}{bn}+\frac{bn\sum_{i>l}\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}=\sum_{i=1}^{l}\frac{1}{bn}+\sum_{i>l}\frac{bn\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}$$

$$\ge\sum_{i=1}^{k^*}\min(1/bn,\frac{bn\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2})+\sum_{i>k^*}\frac{bn\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}$$

$$=\sum_{i=1}^{l^*}\frac{1}{bn}+\sum_{i>l^*}\frac{bn\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2},$$

where l^* is defined as

$$l^*:=\max_{i\le k^*}\left(i:\frac{1}{bn}\le\frac{bn\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}\right).$$

The inequality inside the maximum holds if

$$\lambda_i\ge\frac{\lambda_{k^*+1}r_{k^*}(\Sigma)}{bn},\qquad\text{or}\qquad r_{k^*}(\Sigma)\le\frac{bn\lambda_i}{\lambda_{k^*+1}}.$$

By definition, $r_{k^*-1}(\Sigma)<bn$. Hence

$$r_{k^*}(\Sigma)=\frac{\sum_{i>k^*}\lambda_i}{\lambda_{k^*+1}}=\frac{\sum_{i>k^*-1}\lambda_i-\lambda_{k^*}}{\lambda_{k^*+1}}=\frac{\sum_{i>k^*-1}\lambda_i-\lambda_{k^*}}{\lambda_{k^*+1}}=\frac{\lambda_{k^*}}{\lambda_{k^*+1}}(r_{k^*-1}(\Sigma)-1)$$

$$<\frac{\lambda_{k^*}}{\lambda_{k^*+1}}(bn-1)\le\frac{bn\lambda_{k^*}}{\lambda_{k^*+1}}.$$

Therefore, $\lambda^*=k^*$. Now since for all l

$$\sum_{i=1}^{l}\frac{1}{bn}+\sum_{i>l}\frac{bn\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}\ge\sum_{i=1}^{l^*}\frac{1}{bn}+\sum_{i>l^*}\frac{bn\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2},$$

we have that $l=l^*=k^*$ is the minimizer of the left hand side. Hence

$$\min_{l\le k^*}\left(\frac{l}{bn}+\frac{bn\sum_{i>l}\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}\right)=\frac{k^*}{bn}+\frac{bn\sum_{i>k^*}\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2}=\frac{k^*}{bn}+\frac{bn}{R_{k^*}(\Sigma)}.$$

Step 6: Upper bound on the B term

It remains to find an upper bound for the term $(\theta^*)^T B\theta^*$. Recall

$$B = (I - X^T(XX^T)^{-1}X)\Sigma(I - X^T(XX^T)^{-1}X).$$

First notice that

$$(I - X^T(XX^T)^{-1}X)(X^TX) = X^TX - X^T(XX^T)^{-1}(XX^T)X = X^TX - X^TX = 0.$$

So we can write

$$\begin{aligned}(\theta^*)^T B\theta^* &= (\theta^*)^T(I - X^T(XX^T)^{-1}X)\Sigma(I - X^T(XX^T)^{-1}X)\theta^* \\ &= (\theta^*)^T(I - X^T(XX^T)^{-1}X)(\Sigma - \hat{\Sigma})(I - X^T(XX^T)^{-1}X)\theta^*,\end{aligned}$$

with $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^n x_i x_i^T = \frac{1}{n}X^TX$. Taking norms, we find

$$(\theta^*)^T B\theta^* \leq ||\theta^*||^2||I - X^T(XX^T)^{-1}X||^2||\Sigma - \hat{\Sigma}||.$$

This will allow us to use results on the sample covariance matrix.

Next, take $v \in R(A)$ with $A = I - X^T(XX^T)^{-1}X$. Then there exists a vector w such that

$$v = Aw = w - X^T(XX^T)^{-1}Xw.$$

Then

$$\begin{aligned}Av = A^2w &= (I - X^T(XX^T)^{-1}X)(I - X^T(XX^T)^{-1}X)w \\ &= w - 2X^T(XX^T)^{-1}Xw + X^T(XX^T)^{-1}Xw \\ &= (I - X^T(XX^T)^{-1}X)w = Aw = v.\end{aligned}$$

So for $v \in R(A)$ we have $Av = v$. Furthermore, for $v \in N(A)$, we have $Av = 0$. Since $\mathbb{R}^p = N(A) \perp R(A)$, we have for all $v \in \mathbb{R}^p$

$$||Av|| \leq ||v||.$$

Therefore

$$||I - X^T(XX^T)^{-1}X|| = ||A|| = \max_{v:||v||=1} ||Av|| \leq \max_{v:||v||=1} ||v|| = 1.$$

A second option to show that $||A|| \leq 1$ is by showing that A is a projection matrix. Indeed, as we have seen above $A^2 = A$ and it is also easy to see that $A^T = A$. Hence, A is an orthogonal projection matrix and $||A|| \leq 1$. So we now have

$$(\theta^*)^T B\theta^* \leq ||\theta^*||^2||\Sigma - \hat{\Sigma}||.$$

To upper bound $||\Sigma - \hat{\Sigma}||$, we use Theorem 9 from Koltchinskii et al. (2014) [32]:

Theorem 8 (Theorem 9 in [32])**.** *Let $x, x_1, \ldots, x_n$ be centered subgaussian random variables with covariance Σ. Then there exists a constant $C > 0$ s.t. for all $t \geq 1$, with probability at least $1 - e^{-t}$*

$$||\Sigma - \hat{\Sigma}|| \leq C||\Sigma|| \max\left\{\sqrt{r(\Sigma)/n}, r(\Sigma)/n, \sqrt{t/n}, t/n\right\},$$

where

$$r(\Sigma) = \frac{(\mathbb{E}||x||)^2}{||\Sigma||}, \qquad \hat{\Sigma} = \frac{1}{n}\sum_{j=1}^n x_j x_j^T = \frac{1}{n}X^TX.$$

Proof. See the proof in [32].

Notice first that

$$r(\Sigma) = \frac{(\mathbb{E}||x||)^2}{||\Sigma||} \leq \frac{\mathbb{E}(||x||^2)}{||\Sigma||} = \frac{\mathrm{tr}(\Sigma)}{||\Sigma||} = \frac{\sum_{i=1}^p \lambda_i}{\lambda_1} = r_0(\Sigma).$$

So according to Theorem 8 we have that there exists a constant C s.t. for all $t \geq 1$, with probability at least $1 - e^{-t}$

$$||\Sigma - \hat{\Sigma}|| \leq C||\Sigma|| \max\{\sqrt{r_0(\Sigma)/n}, r_0(\Sigma)/n, \sqrt{t/n}, t/n\}.$$

Therefore there exists a constant C s.t. for all $t \geq 1$ (so $\sqrt{t/n} \geq t/n$), with probability at least $1 - e^{-t}$

$$(\theta^*)^T B\theta^* \leq C||\theta^*||^2||\Sigma|| \max\{\sqrt{r_0(\Sigma)/n}, r_0(\Sigma)/n, \sqrt{t/n}\}.$$

Step 7: Combining the results

From equation (9.6) we know that there exist $b, c \geq 1$ s.t. if $0 \leq k \leq n/c$, $r_k(\Sigma) \geq bn$ and $l \leq k$, then with prob. at least $1 - 7e^{-n/c}$ we have

$$\mathrm{tr}(C) \leq c\left(\frac{l}{n} + n\frac{\sum_{i>l}\lambda_i^2}{(\sum_{i>l}\lambda_i)^2}\right).$$

From Step 5 we know that the minimum over all $l \leq k$ is attained at $l = k$. Now setting k^* to be the minimum value of k for which $r_k(\Sigma) \geq bn$, the sharpest upper bound is found for choosing $l = k^*$, so that with probability at least $1 - 7e^{-n/c}$ we have

$$\mathrm{tr}(C) \leq c\min_{l \leq k^*}\left(\frac{l}{n} + \frac{n\sum_{i>l}\lambda_i^2}{(\lambda_{k^*+1}r_{k^*}(\Sigma))^2} = c(\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}\right).$$

Furthermore, by Step 6 we have, with probability at least $1 - e^{-t}$ $(t \geq 1)$

$$(\theta^*)^T B\theta^* \leq c_2||\theta^*||^2||\Sigma||\max\{\sqrt{r_0(\Sigma)/n}, r_0(\Sigma)/n, \sqrt{t/n}\}.$$

By Step 1 we have, with prob. at least $1 - e^{-t}$ $(t \geq 1)$

$$R_{excess}(\hat{\theta}) \leq 2(\theta^*)^T B\theta^* + 12t\sigma^2\mathrm{tr}(C).$$

Combining these results, we find that, with probability at least $1 - 7e^{-t}$, for $1 \leq t \leq n/c$, that

$$R_{excess}(\hat{\theta}) \leq 2c_2||\theta^*||^2||\Sigma||\max\{\sqrt{r_0(\Sigma)/n}, r_0(\Sigma)/n, \sqrt{t/n}\} + 12t\sigma^2 c\left(\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}\right).$$

Now choose $c_0 = \max\{12c, 2c_2\}$. Then with prob. at least $1 - e^{-t}$, for $1 \leq t \leq n/c_0$, we have

$$R_{excess}(\hat{\theta}) \leq c_0||\theta^*||^2||\Sigma||\max\{\sqrt{r_0(\Sigma)/n}, r_0(\Sigma)/n, \sqrt{t/n}\} + c_0t\sigma^2\left(\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}\right).$$

Or by setting $t = \log(1/\delta)$, we have the following. There exist $b, c \geq 1$ such that, with probability at least $1 - \delta$ (where $e^{-n/c} \leq \delta \leq e^{-1}$), we have

$$R_{excess}(\hat{\theta}) \leq c||\theta^*||^2||\Sigma||\max\left\{\sqrt{\frac{r_0(\Sigma)}{n}}, \frac{r_0(\Sigma)}{n}, \sqrt{\frac{\log(1/\delta)}{n}}\right\} + c\sigma^2\log(1/\delta)\left(\frac{k^*}{n} + \frac{n}{R_{k^*}(\Sigma)}\right),$$

with $k^* := \min\{k \geq 0 : r_k(\Sigma) \geq bn\}$. This is exactly the statement in Theorem 4.

9.4 R code

Code for Chapter 2

```
## Libraries
library(MASS)
library(RColorBrewer)
library(purrr)

## Functions
risk <- function(X,X_p,eps, sigma,Sigma,theta_star,P){
  n = dim(X)[1]
  p = dim(X_p)[2]

  ## generate the true labels
  y = X %*% theta_star + eps

  ## calculate least-squares / min-norm solution
  theta_hat = numeric(d)
  if (p>= n){
    theta_hat[P] = t(X_p) %*% solve( (X_p %*% t(X_p) ) ) %*% y
  }
  else{
    theta_hat[P] = solve( t(X_p)%*%X_p) %*% t(X_p) %*% y
  }
  theta_hat[-P] = 0

  ## compute the training/test risk
  R = 0
  for (i in 1:n){
    R = R + 1/n*(y[i]-X[i,]%*% theta_hat )^2
  }
  R_test = sigma^2 + t(theta_hat - theta_star) %*% Sigma %*% (theta_hat - theta_star)
  theta_norm = norm(theta_hat, type = '2') # L_2 norm of parameter vector
  return(c(R,R_test,theta_norm) )
}

## theoretical risk
theo_det <- function(theta_star, sigma, p,n, Sigma){
  P = 1:p
  if (p<= n-2){
    R = (norm( Sigma[-P,-P]^(1/2) %*% theta_star[-P] , type='2')^2
         + sigma^2)*(1+p/(n-p-1) )
  }
  if(n-1 <= p & p <= n+1){
    R = 1000      # infinity
  }
  if(p>= n+2 & p < d){
    R = norm(theta_star[P] ,type ='2')^2*(1-n/p) +
      ( norm(theta_star[-P],type ='2')^2 + sigma^2)*(1+n/(p-n-1) )
  }
  if(p == d){
    R = norm(theta_star[P], type = '2')^2*(1-n/p) + sigma^2*(1+n/(p-n-1) )
  }
  return(R)
}

theo_rand <- function(theta_star, sigma, p, n,d, Sigma){
  if (p <= n-2){
    R = ((1-p/d)*norm(Sigma^(1/2)%*%theta_star,'2')^2 + sigma^2)*(1+p/(n-p-1) )
    }
  if(n-1 <= p & p<= n+1){
    R = 1000     # infinity
  }
  if(p>=n+2){
    R = norm(theta_star,'2')^2*(1-n/d*(2-(d-n-1)/(p-n-1) )) + sigma^2*(1+n/(p-n-1) )
```

```
    }
  return(R)
}

## Setting up parameters
n = 40               # number of training data points
d = 100              # dimension
M = 100               # number of Monte Carlo iterations
sigma = 1/5       # sd of the label noise
Sigma = diag(1, nrow = d, ncol = d)
theta_star = 1/(1:d)
theta_star = theta_star / norm(theta_star, '2')
rnd = 1       # random choice (rnd=1) or deterministic choice (rnd=0)

## Monte Carlo simulation
R = matrix(0,d,M)
R_mean = numeric(d)
R_test = matrix(0,d,M)
R_test_mean = numeric(d)
theta = matrix(0,d,M)
theta_mean = numeric(d)

for (p in 1:d){
  for(i in 1:M){
    ## sample new training data
    X = matrix(mvrnorm(n, numeric(d), Sigma), n,d)     # training data matrix
    X = matrix(X,n,d)                      # convert to proper matrix
    if (rnd==1){
      P = sample(1:d,p)
    }
    if (rnd==0){
      P = 1:p
    }

    X_p = matrix(X[1:n,P],n,p)
    ## sample new label noise
    eps = rnorm(n, mean = 0, sd = sigma)          # noise in training labels
    result = risk(X,X_p,eps,sigma,Sigma,theta_star,P)
    R[p,i] = result[1]
    R_test[p,i] = result[2]
    theta[p,i] = result[3]
  }
  R_mean[p] = mean(R[p,])
  R_test_mean[p] = mean(R_test[p,])
  theta_mean[p] = mean(theta[p,])

  ## show progression for p
  if (p%%10 == 0){
    cat('Loading ', p/d*100, '%', '\n')
  }
}

## plot results
cols = brewer.pal(9,'Set1')
plot(1:d, R_test_mean, xlab = 'Number of parameters', ylab = 'Risk', ylim = c(0,10), col = cols[2] )
abline(v=n, col = cols[1], lwd = 2)
lines(1:d, R_mean, col = cols[3], lwd = 2 )
```

Code for Chapter 4

```
## Libraries
library(MASS)
library(RColorBrewer)

## Functions
excess_risk <- function(X,eps, Sigma,theta_star){
```

```
  n = dim(X)[1]
  p = dim(X)[2]

  ## generate the true labels
  y = X %*% theta_star + eps

  ## calculate least-squares / min-norm solution
  if (p>= n){
    theta_hat = t(X) %*% solve( (X %*% t(X) ) ) %*% y
  }
  else{
    theta_hat = solve( (t(X) %*% X)) %*% t(X) %*% y
  }

  ## compute the excess training/test risk
  R = 0
  for (i in 1:n){
    R = R + 1/n*(y[i]-X[i,]%*% theta_hat )^2 - 1/n*(y[i] - X[i,] %*% theta_star)^2
  }
  R_test = t(theta_hat - theta_star) %*% Sigma %*% (theta_hat - theta_star)
  theta_norm = norm(theta_hat, type = '2') # L_2 norm of parameter vector

  return(c(R,R_test,theta_norm) )
}

## Monte Carlo simulation
MC_sim <- function(n,d,M,sigma,Sigma){
  R = matrix(0,d,M)
  R_mean = numeric(d)
  R_test = matrix(0,d,M)
  R_test_mean = numeric(d)
  theta = matrix(0,d,M)
  theta_mean = numeric(d)

  for (p in 1:d){
    theta_star = matrix(1,p,1)
    theta_star = theta_star / norm(theta_star, '2')

    for(i in 1:M){
      ## sample new training data
      X = matrix(mvrnorm(n, numeric(p), Sigma[1:p,1:p]),n,p)

      ## sample new label noise
      eps = rnorm(n, mean = 0, sd = sigma)          # noise in training labels

      result = excess_risk(X,eps,Sigma[1:p,1:p],theta_star)
      R[p,i] = result[1]
      R_test[p,i] = result[2]
      theta[p,i] = result[3]
    }

    R_mean[p] = mean(R[p,])
    R_test_mean[p] = mean(R_test[p,])
    theta_mean[p] = mean(theta[p,])

    ## show progression for p
    if (p%%10 == 0){
      cat('Loading ', p/d*100, '%', '\n')
    }

  }

  S = matrix(0,3,p)
  S[1,] = R_mean
  S[2,] = R_test_mean
  S[3,] = theta_mean
```

```
  return(S)

}

## Variables

n = 5              # number of training data points
d = 100            # maximum number of parameters
M = 100              # number of Monte Carlo iterations
sigma = 1          # sd of the label noise

Delta = 1/n^3
Sigma = diag(exp(-(1:d))+Delta, nrow = d, ncol = d)
# alpha = 2
# Sigma = diag((1:d)^(-alpha), nrow = d, ncol = d)
# Sigma = diag(1, nrow = d, ncol = d)

S = MC_sim(n,d,M,sigma,Sigma)
R_mean = S[1,]
R_test_mean = S[2,]

## plot results
cols = brewer.pal(9,'Set1')
plot(1:d, R_test_mean, type = 'l', col = cols[2], lwd = 2,
    xlab = 'Number of parameters', ylab = 'Excess test risk', log = 'y', ylim = c(0.1, 50),
    main = 'Logarithm of test risk against p')
abline(v=n, col=cols[1], lwd = 2)
#lines(1:d, R_mean, type = 'l', col = cols[3], lwd = 2)
#abline(h = R_test_mean[1], col = cols[1], lty = 2, lwd = 2)
#legend('topright',
#       inset = 0.05,
#       legend = c('Training risk', 'Test risk', 'Test risk at p=0'),
#       col = c(cols[3],cols[2], cols[1]), lwd = 2,
#       cex = 0.8 )
```

Code for Chapter 5, setting 1

```
library(MASS)
library(RColorBrewer)
library(purrr)
library(pracma)

## functions
f_true = function(x){
  d = length(x)
  theta = 1/(1:d)
  theta = theta/norm(theta, '2')
  # sine function
  y_sin = sin(t(theta)%*%as.matrix(x))
  # linear regression
  y_lin = t(theta)%*%as.matrix(x)
  return(y_lin)
}

kernel <- function(s,t,l){
  s = as.matrix(s)
  t = as.matrix(t)
  # Gaussian kernel
  K_gauss = exp(-1/(2*l^2)*norm(s-t,'2')^2 )
  # Linear kernel
  K_linear = t(s)%*%t
  # Polynomial kernel
  q = 2
  K_pol = (1+t(s)%*%t)^q
  return(K_pol)
```

```
}

# calculate kernel matrix
K_mat <- function(A,B,l){
  A = as.matrix(A)
  B = as.matrix(B)
  nrow = dim(A)[1]
  ncol = dim(B)[1]
  Sigma = matrix(NA,nrow, ncol)
  for (i in 1:nrow){
    for (j in 1:ncol){
      Sigma[i,j] = kernel(A[i,],B[j,],l)
    }
  }
  return(Sigma)
}

## perform Monte Carlo simulation
MC_sim <- function(M,d,n,n_test,sigma,l,rnd){
  R_train = matrix(NA,M,d)
  R_test = matrix(NA,M,d)
  R_train_mean = matrix(NA,d,1)
  R_test_mean = matrix(NA,d,1)

  for (p in 1:d){
    for (m in 1:M){
      Sigma = diag(1, d,d)
      xtrain = matrix(mvrnorm(n,numeric(d),Sigma),n,d)
      xtest = matrix(mvrnorm(n_test,numeric(d),Sigma), n_test,d)

      if (rnd==0){
        P = 1:p
      }
      if (rnd == 1){
        P = sample(1:d,p)
      }

      x_p = matrix(xtrain[1:n,P],n,p)
      ytrain = numeric(n)
      for(i in 1:n){
        ytrain[i] = f_true(xtrain[i,])+sigma*rnorm(1,0,1)
      }

      K = K_mat(x_p, x_p,l)
      A = K+sigma^2*diag(1,n,n)
      alpha = pinv(A)%*%ytrain

      ## calculate f_hat
      f_hat = function(xtrain, x){
        n = dim(xtrain)[1]
        fhat = 0
        for (i in 1:n){
          fhat = fhat + alpha[i]*kernel(xtrain[i,],x,l)
        }
        return(fhat)
      }

      R_train[m,p] = 0
      for (i in 1:n){
        R_train[m,p] = R_train[m,p]+1/n*(f_true(xtrain[i,] )- f_hat(x_p, xtrain[i,P] ) )^2
      }

      R_test[m,p] = 0
      for (j in 1:n_test){
        R_test[m,p] = R_test[m,p]+1/n_test*(f_true(xtest[j,] )-f_hat(x_p, xtest[j,P] ) )^2
      }
```

```
    }

    R_train_mean[p] = mean(R_train[,p] )
    R_test_mean[p] = mean(R_test[,p] )

    if (p%%(d/100)==0){
      cat('Loading...', p/d*100, '%', '\n')
    }
  }

  S = matrix(NA,2,p)
  S[1,] = R_train_mean
  S[2,] = R_test_mean
  return(S)
}

## set parameters
n = 45
n_test = 100
sigma = 0
l = 1
rnd = 1

## perform MC simulation
M = 10
d = 100
S = MC_sim(M,d,n,n_test,sigma,l,rnd)
R_train_mean = S[1,]
R_test_mean = S[2,]

## plot training / test risk
cols = brewer.pal(9,'Set1')
plot(1:d, R_test_mean, lwd = 2, col = cols[2], type = 'l',
     xlab = 'p', ylab = 'Test risk')
#lines(1:d, R_train_mean, lwd = 2, col = cols[3] )
abline(h = R_test_mean[1], lwd = 2, col = cols[1], lty = 2 )
```

Code for Chapter 5, setting 2

```
library(MASS)
library(RColorBrewer)
library(purrr)
library(pracma)

## functions
f_true = function(x,l){
  n = length(x)
  Sigma = matrix(NA,n,n)
  for (i in 1:n){
    for (j in 1:n){
      Sigma[i,j] = exp(-(x[i] - x[j] )^2 / (4*l^2 ) ) # Gaussian kernel
      #Sigma[i,j] = x[i]*x[j]
    }
  }
  set.seed(0)
  y = mvrnorm(1, numeric(n), Sigma)
  return(y)
}

Phi_mat = function(X,l,p){
  n = length(X)
  Phi = matrix(NA,n,p)

  for (j in 1:n){
    for (k in 1:p){
      c = runif(1,min=1,max=3*n)
```

```
      Phi[j,k] = exp( - (X[j] - c)^2 / (2*l^2) ) # Gaussian basis functions
      #Phi[j,k] = X[j]*X[k]
    }
  }
  return(Phi)
}

## parameters
sigma = 0
n = 50
n_test = 2*n
p_max = 500
l = 100
M = 10

## Monte Carlo simulation
R_train = matrix(NA,M,p_max)
R_test = matrix(NA,M,p_max)
R_train_mean = numeric(p_max)
R_test_mean = numeric(p_max)

for (p in 1:p_max){
  for (m in 1:M){
    X_tot = seq(1,n+n_test,length.out = n+n_test)

    train_ind = sort(sample(1:(n+n_test), n) )
    test_ind = sort( setdiff(1:(n+n_test), train_ind) )

    X = X_tot[train_ind]
    Xtest = X_tot[test_ind]
    ftrue = f_true(X_tot, l)

    y = ftrue[train_ind] + sigma*rnorm(n,0,1)
    ytest = ftrue[test_ind] + sigma*rnorm(n_test,0,1)

    Phi_train = Phi_mat(X,l,p)
    Phi_test = Phi_mat(Xtest, l, p)

    theta_hat = pinv(Phi_train)%*%y

    yhat = Phi_train%*%theta_hat
    yhat_test = Phi_test%*%theta_hat

    R_train[m,p] = 1/n*norm(y - yhat, '2')^2
    R_test[m,p] = 1/n_test*norm(ytest - yhat_test, '2')^2
  }

  R_train_mean[p] = mean(R_train[,p])
  R_test_mean[p] = mean(R_test[,p] )

  # show progress
  cat('Loading...', p/p_max*100, '%', '\n')

}

## plots
cols = brewer.pal(9,'Set1')
plot(1:p_max, R_test_mean, type = 'l', main = paste('Risk curves for l =', l, 'and sigma =', sigma),
     xlab = 'Number of parameters', ylab = 'Risk', col = cols[2], lwd=2,
     log = 'y')
#lines(1:p_max, R_train_mean, col = cols[3],lwd=2)
abline(h = min(R_test_mean), col = cols[1], lwd = 2, lty = 2)
```

Code for Chapter 6

```
## Libraries
library(MASS)
library(RColorBrewer)
library(purrr)

## Set parameters
n = 256
d = 1024
M = 10
step = 1

## Construct discrete Fourier transform
F_d = matrix(0,d,d)
omega = exp(-2i*pi/d)
for (i in 1:d){
  for (j in 1:d){
    F_d[i,j] = 1/sqrt(d)*omega^( (i-1)*(j-1) )
  }
}

# sample true value of theta
Sigma = diag(1, nrow = d, ncol = d)
theta = mvrnorm(1, numeric(d), 1/d*Sigma)
theta = theta/ norm(theta,'2')

# response vector
y = F_d %*% theta

## simulations
MSE = matrix(0,d,M)
MSE_mean = numeric(d)
D = 1:d
for (p in 1:d){
  for (m in 1:M){

    if (p%%step==0){
      # N = D[ rbernoulli(d,n/d) ]
      # P = D[ rbernoulli(d,p/d) ]
      N = sample(1:d,n)
      P = sample(1:d,p)

      if(is_empty(P)){
        P = 1
      }
      if(is_empty(N)){
        N = 1
      }

      F_np = matrix(F_d[N,P], length(N), length(P) )
      if (length(P) >= length(N) ){
        F_dagger = Conj(t(F_np))%*%solve(F_np%*%Conj(t(F_np)) )
      } else {
        F_dagger = solve(Conj(t(F_np))%*%F_np)%*%Conj(t(F_np))
      }

      theta_hat = numeric(d)
      theta_hat[P] = F_dagger%*%y[N]
      y_hat = F_d%*%theta_hat
      MSE[p,m] = norm(theta-theta_hat,'2')^2
    }
  }
  MSE_mean[p] = mean(MSE[p,])

  # show progress
```

```
  if (p%%100==0){
    cat('Loading...', p/d*100,'%', '\n')
  }
}

## make plots
cols = brewer.pal(9,'Set1')
plot(seq(step,d,step), MSE_mean[which(MSE_mean!=0)], type = 'l', lwd = 2, col = cols[2], log = 'y', ylim = c(0.5, 4
abline(h = 0.75, lty = 2, lwd = 2, col = cols[1])
```

Code for Chapter 7

```
library(MASS)
library(RColorBrewer)

R_opper <- function(alpha){

  if(alpha<=1){
    R = 1/pi*acos( sqrt( 2*alpha*(1-alpha) / (pi-2*alpha) ) )
  }
  else{
    R = 1/pi*acos(sqrt(2*(alpha-1)/(pi+2*alpha-4)))
  }

  return(R)
}

## Monte Carlo simulation

p = 100
d = 400  # max number of parameters
M = 100

MSE_train = matrix(0,d, M)
MSE_train_mean = numeric(d)
MSE_test = matrix(0,d,M)
MSE_test_mean = numeric(d)

theta_star = 1/sqrt(p)*matrix(1,p,1)

for (n in 1:d){

  for(m in 1:M){
    data = sample(c(-1,1), n*p, replace = TRUE)
    X = matrix(data, n, p)
    y_true = sign(X%*%theta_star)

    n_test = n
    data_test = sample(c(-1,1), n_test*p, replace = TRUE)
    X_test = matrix(data_test,n_test,p)
    y_test = sign(X_test%*%theta_star)

    if (p> n){
      theta_hat = t(X) %*% solve( (X %*% t(X) ) ) %*% y_true
    }
    if(p<=n){
      theta_hat = solve( (t(X)%*%X) ) %*% t(X) %*% y_true
    }

    y_pred = sign(X%*%theta_hat)
    y_pred_test = sign(X_test%*%theta_hat)
    MSE_train[n,m] = sum(y_true!=y_pred)/n
    MSE_test[n,m] = sum(y_test!=y_pred_test)/n_test
  }

  MSE_train_mean[n] = mean(MSE_train[n,])
```

```
  MSE_test_mean[n] = mean(MSE_test[n,])

  if (n%%10 == 0){
    cat('Loading ', n/d*100, '%', '\n')

  }
}

## plot training risk
alpha = (1:d)/p
plot(alpha, MSE_train_mean, type = 'l', lwd = 2,
     main = 'Training risk vs alpha', xlab = 'alpha',
     ylab = 'Training risk')

## plot test risk and theoretical bound from Opper
cols = brewer.pal(9,'Set1')
plot(alpha, MSE_test_mean, main = 'Test risk vs gamma',
     xlab = 'gamma', ylab = 'Test risk')
R_opp = numeric(d)
for(i in 1:d){
  R_opp[i] = R_opper(alpha[i])
}
lines(alpha, R_opp, type = 'l', lwd = 2, col = 'red')
abline(v=1, lty = 2)
```