Heteroskedasticity in online linear regression: An application of weighted regression

Mereu, N.

Heteroskedasticity in Online Linear Regression

An application of Weighted Regression.

Niek Mereu (s1234269)

Thesis advisors: Dr. T.A.L van Erven
Msc. D. van der Hoeven

MASTER THESIS

Defended on Month Day, Year

Specialization: Data Science

STATISTICAL SCIENCE
FOR THE LIFE AND BEHAVIOURAL
SCIENCES

Abstract

Online Linear Regression is a sequential variant of regression in which the data points arrive one by one. It is normally studied in the game-theoretic framework of Online Convex Optimization, which models the data as being generated by an adversary. In this framework, the standard statistical procedure of Online Ridge Regression is known to be essentially optimal.

In Statistics, there is an improvement for Ridge Regression when the noise is not constant. This improvement is Weighted Ridge Regression, which relies on weighting the data by their variances. In this thesis, we will employ weighting in Online Ridge Regression to show that an improvement over Online Ridge Regression can be made.

We furthermore explored the situation where weighting is disadvantageous, mathematically and experimentally using simulations. Finally we applied Online Weighted Ridge Regression to different real-world datasets and found that we also can improve Online Ridge Regression in practical situations.

Contents

Chapter 1

General introduction

Prediction, in the Online Convex Optimization (OCO) context, deals with the progression of events observed in nature. The learner must predict the next element in a potentially infinite sequence of elements, given past information. After the learner's prediction, the true answer (the signal) is revealed and the learner pays the loss between the chosen answer and the signal. OCO is a game-theoretic framework, where statistical assumptions are almost always absent.

The OCO framework is useful in different scenarios. Two examples of scenarios are the situation where a dataset is too large to fit in memory and the situation where data is constantly being generated. Illustrations that fit the second scenario include e-mail spam filtering, credit card fraud detection and equity market portfolio selection. In the context of e-mail spam filtering, the system ought to classify mails as spam or valid. The system has to cope with a constant in-flow of adversarially generated data, as the spam generator is deliberately trying to fool the system. Adversarial and varying data requires a dynamic system, which is a hallmark of OCO (Hazan, 2015).

In Online Linear Regression (OLR), a special case of OCO, the learner is asked for a reaction on feature $\mathbf{x}_t \subseteq \mathbb{R}^d$. The learner then predicts vector $\mathbf{w}_t$, the signal y_t is subsequently revealed and the learner pays $(y_t - \mathbf{x}_t^\top \mathbf{w}_t)^2$: the squared loss, (Shalev-Shwartz, 2012). Online Ridge Regression (ORR), an algorithm for OLR, is known to be essentially optimal (Cesa-Bianchi and Lugosi, 2006).

We change the framework from OCO to Statistics, where we define the following regression model:

$$y_t = \mathbf{x}_t^\top \mathbf{u}^* + \epsilon_t \quad \forall t \tag{1.1}$$

with error term ϵ_t and true parameters $\mathbf{u}^*$. When the variance of this error term is not constant, we speak of heteroskedasticity. This arises, for example, in the situation where the size of the observed units differs drastically. When our data contains firms with one employee, as well as firms with a thousand employees, we can expect the large firms to have higher values on the signal, on the feature(s) and on the unobserved variables that are collected in the error term. The variance of large firms tends to be larger than the variance of small firms (Verbeek, 2012)

If the variance of the error term is homoskedastic (constant), the standard method for obtaining an estimator of $\mathbf{u}^*$ is, much like in OLR, Ridge Regression. When the noise is heteroskedastic, however, an improvement over Ridge Regression can be made: Weighted Ridge Regression (Askin and Montgomery, 1980). In essence, this method comes down to weighting all the data by their individual variances. In this thesis, we will apply weighting to ORR to take advantage of heteroskedasticity and potentially make better predictions relative to the unweighted estimator. From the preceding paragraphs we deduce the following central research question:

"Can we, under heteroskedasticity, improve Online Ridge Regression predictions by weighting the data by their variances?"

As far as we know, heteroskedasticity in OLR has been named in the literature once (Anava and Mannor, 2016). In their research paper, the authors introduce a framework that is a hybrid between OLR and Statistics. They assume a conditional variance for y_t given $\mathbf{x}_t$, which they subsequently want to learn online. They claim that "traditional modeling assumptions on the signal generation can be substantially relaxed while still maintaining the ability to solve the problem". Unlike our research, Anava and Mannor (2016) do not try to exploit heteroskedasticity for better predictions. The similarity of this thesis to Anava and Mannor (2016) lies in the framework, which is a hybrid: incorporating elements from OCO as well as from Statistics. Like Anava and Mannor (2016) we assume a conditional variance of the signal given the feature(s). A part of the significance of our work is bridging the chasm between Statistics, with restrictive assumptions on the error term, and OCO, where the error term along with the statistical assumptions are absent.

Outline. The remainder of this thesis is organised as follows. The introduction consists of Chapter 2 and Chapter 3. Chapter 2 introduces the OCO framework with an elaboration on OLR, as a special case of OCO. Two algorithms for OLR: Online Least Squares (OnLS) and Online Ridge Regression are introduced in this chapter. All tools needed from Statistics are then introduced in Chapter 3, where we explain how weighting is used in Statistics to deal with heteroskedasticty. We then bridge to OCO and describe how we incorporate weights in the OnLS and ORR algorithms. In Chapter 4 we analyze the expected loss of the Online Weighted Ridge Regression (OWRR) and the ORR algorithms to further examine whether OWRR is always better than ORR. We find that this is true for the one-dimensional case, but not for higher dimensions. We then generalize this theoretical insight in simulation studies. For different experimental set-ups, we search for situations where OWRR is worse than ORR. In Chapter 5 we compare all four algorithms (OnLS, OWLS, ORR, OWRR) on three real-world, heteroskedastic datasets. The first dataset is about the number of cellphones in countries; the second dataset involves economic output data from Belgian firms; the third dataset is about housing prices in Boston. We show that, even when having to learn the variance function on the fly, the weighted algorithms can outperform the unweighted ones. The discussion and recommendations for future research are found in Chapter 6.

Chapter 2

Online Convex Optimization

This chaper contains the first half of the introduction. In this chapter we introduce the Online Convex Optimization (OCO) framework. We commence broadly by introducing the general outline of OCO and its performance measure, Regret, in Section 2.1. We subsequently introduce Online Linear Regression (OLR) as a special case of OCO in Section 2.2. We also introduce Online Least Squares (OnLS) as an algorithm for OLR and we elaborate on the Regret bound of this algorithm. In Section 2.3 we introduce Online Ridge Regression (ORR) which has a better Regret bound than OnLS.

2.1 The OCO framework

OCO is the process of answering a sequence of questions, given knowledge of the correct answers to previous questions and additional information if accessible. The learning is not executed instantly, but takes place in consecutive rounds $t = 1, 2, ..., T$. The learner predicts vector $\mathbf{w}_t \subseteq \mathbb{R}^d$ in round t. Consequently, the environment reveals a convex loss function $\ell_t : \mathbb{R}^d \to \mathbb{R}$.The learner then suffers $\ell_t(\mathbf{w}_t) = \ell(\mathbf{w}, \mathbf{x}_t)$, a convex loss function (Shalev-Shwartz, 2012). A summary of the OCO framework is shown in Protocol 2.1.

By the very nature of the framework, OCO is a repeated game played by a player versus an opponent, hereafter referred to as the environment. The behaviour of this environment can be deterministic (e.g. adversarial) or stochastic. A direct consequence is that the learner has to prepare for a worst-case scenario, namely that the environment is adversarially adaptive to the learner's own behaviour (Shalev-Shwartz, 2012).

Protocol 2.1 Online Convex Optimization

```
for $t = 1, 2..., T$ do
    predict $\mathbf{w}_t \subseteq \mathbb{R}^d$
    receive convex loss function $\ell_t : \mathbb{R}^d \to \mathbb{R}$
    pay loss $\ell_t(\mathbf{w}_t)$
end for
```

The learner is required to compete with an hypothesis $\mathbf{u}$. The learner's performance is measured in Regret, which is defined as the difference between the cumulative loss of the learner and the cumulative loss of the optimal hypothesis:

$$\text{Regret}_T(\mathbf{u}) = \sum_{t=1}^{T} \ell_t(\mathbf{w}_t) - \sum_{t=1}^{T} \ell_t(\mathbf{u}) \tag{2.1}$$

which is most commonly evaluated with respect to $\mathbf{u} = \mathbf{u}^*$, where comparator $\mathbf{u}^*$ is the minimizer of cumulative loss: $\mathbf{u}^* = \text{argmin}_{\mathbf{w}} \sum_{t=1}^{T} \ell_t(\mathbf{w}_t)$.

The objective of the learner is to have Regret that grows at most sublinearly with T for any $\mathbf{u}$ and in particular for $\mathbf{u}^*$, without making any statistical assumption on the losses. Intuitively, Regret measures how sorry the learner is for not having used the optimal hypothesis $\mathbf{u}^*$ in retrospect (Salev-Shwartz, 2007).

An example of an algorithm that is capable of achieving Regret that grows sublinear with T is Online Gradient Descent. With initialization of $\mathbf{w}_1 = \vec{0}$ we update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \ell_t(\mathbf{w}_t)$ with step size η and $\nabla \ell_t(\mathbf{w}_t)$ the gradient of ℓ_t specifically at $\mathbf{w}_t$. The Regret is shown to be bounded by $BL\sqrt{2T}$ when $||\mathbf{u}^*||_2 \leq B$, $||\nabla \ell_t(\mathbf{w}_t)||_2 \leq L$ and $\eta = \frac{B}{L\sqrt{2T}}$ with $\|\cdot\|_2$ being the Euclidean-norm (Shalev-Shwartz, 2012).

2.2 Online Linear Regression and Online Least Squares

Online Linear Regression (OLR) is a special case of OCO. In each round, the learner is asked for a reaction on $\mathbf{x}_t \subseteq \mathbb{R}^d$, where d is the number of features. The learner then predicts vector $\mathbf{w}_t$ which should be competitive with reference vector $\mathbf{u}^*$. The loss function in the OLR framework is the squared loss $\ell_t(\mathbf{w}_t) = (\mathbf{x}_t^\top \mathbf{w} - y_t)^2$, which is convex.

A basic algorithm for OLR is Online Least Squares (OnLS). By minimizing the sum of the loss functions, we can find an expression for $\mathbf{w}_t$ that mirrors the Maximum Likelihood Estimator derived in Statistics:

$$\begin{aligned} \mathbf{w}_t^L &= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{s=1}^{t-1} \ell_t(\mathbf{w}_t) \\ &= (\mathbf{A}_t)^{-1}\mathbf{b}_t \end{aligned} \tag{2.2}$$

where $\mathbf{A}_t = \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\top$ and $\mathbf{b}_t = \sum_{s=1}^{t-1} \mathbf{x}_s y_s$. Note that we take the outer product in the definition of $\mathbf{A}_t$. This way, $\mathbf{A}_{t-1}$ (as well as vector $\mathbf{b}_t$) can be updated incrementally with the information of round t.

No satisfying Regret bound for the algorithm is achieved. Firstly, as $\mathbf{A}_t$ is not invertible for the first $t < d$ rounds, the Regret would not even be defined for these rounds. Secondly, an adversary could just wait until the learner predicts $\mathbf{w}_t$ and could then choose an y_t that would maximize $\ell_t(\mathbf{w}_t)$. The predicted vector $\mathbf{w}_t$ is not stable and would change a lot from round to round. A better Regret bound is derived for Online Ridge Regression, with a regularization term to stablize the predictions.

2.3 Online Ridge Regression

The preceding algorithm is easily altered to give rise to an algorithm that does have a Regret bound that grows sublinear with T: Online Ridge Regression (ORR). We now want to minimize the ridge loss function, which is the squared loss with an additional penalty term: $\lambda\|\mathbf{w}\|_2^2$ with penalty parameter $\lambda > 0$. We find an expression for $\mathbf{w}_t^R$ in a similar way as for OnLS:

$$\begin{aligned} \mathbf{w}_t^R &= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{s=1}^{t-1} \ell_t(\mathbf{w}_t) + \lambda\|\mathbf{w}\|_2^2 \\ &= (\mathbf{A}_t^R)^{-1}\mathbf{b}_t \end{aligned} \tag{2.3}$$

where $\mathbf{A}_t^R = \lambda \boldsymbol{I} + \sum_{s=1}^{t-1} \mathbf{x}_s \mathbf{x}_s^\top$ and $\mathbf{b}_t = \sum_{s=1}^{t-1} \mathbf{x}_s y_s$ (Cesa-Bianchi and Lugosi, 2006).

With application of the penalty term (regularization) on the weights $\mathbf{w}$, we can do better than the OLR algorithm described in Section 2.2 (Cesa-Bianchi and Lugosi, 2006). The reason is that regularization stabilizes $\mathbf{w}_t^R$ from round

$t-1$ to t (Shalev-Shwartz, 2012). Consequently it is shown that for $\lambda = 1$ the Regret is bounded:

$$\text{Regret}_T(\mathbf{u}) \leq ||\mathbf{u}||_2^2 + \left(\sum_{i=1}^{d} \ln(1+\rho_i) \right) \max_{t=1,\ldots,T} \ell_t(\mathbf{w}_t) \tag{2.4}$$

where $\rho_1, ..., \rho_d$ are the eigenvalues of matrix $\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^\top$. With $\|\mathbf{u}\|$, $\|\mathbf{x}_t\|$ and $|y_t|$ bounded by constants the Regret bound is $O(\ln T)$. The Regret grows at most logarithmic with T. No such bound is available for OnLS (Cesa-Bianchi and Lugosi, 2006).

Chapter 3

Heteroskedasticity and Weighted Regression

This chapter contains the second half of our introduction. In this chapter we build a bridge to Statistics and elaborate on the Statistical concepts that we need in order to apply weighting in OLR. In Section 3.1 we introduce the homoskedastic model. We explain how heteroskedasticity impairs the efficiency of the Least Squares estimator. In Section 3.2 we proceed with the heteroskedastic model and a way to solve heteroskedasticity: Weighted Least Squares. In Section 3.3 we introduce Weighted Ridge Regression; a combination of weighting and penalization. Finally, in Section 3.4 we move to a more practical perspective and we introduce Iterative Reweighted Ridge Regression (IRRR), used to find weights for Online Weighted Least Squares (OWLS) and Online Weighted Ridge Regression (OWRR).

3.1 The homoskedastic model and heteroskedasticity

For each round t the model is $y_t = \mathbf{x}_t^\top \mathbf{u}^* + \epsilon_t$ with indepdent ϵ_t, $\mathbb{E}\left[\epsilon_t\right] = 0$ and conditional variance $\mathrm{Var}\left[y_t|\mathbf{x}_t\right] = \sigma_t^2$. When the expression $\sigma_t^2 = \sigma^2$ is true, we speak of homoskedasticity. To find $\mathbf{w}^L$, the Least Squares estimator, we want to minimize the squared loss; this minimization problem is found in Equation 2.2.

The Gauss-Markov theorem states that $\mathbf{w}^L$ is the Best Linear Unbiased Estimator (BLUE) under homoskedasticity (Clapham and Nicholson, 2009). The theorem states that, out of all unbiased, linear estimators, the Least Squares estimator has the lowest variance.

Under heteroskedasticity, the conditional variance becomes: $\mathrm{Var}\left[y_t|\mathbf{x}_t\right] = \sigma_t^2 = \sigma^2(\mathbf{x}_t)$ with variance function $\sigma^2(\mathbf{x})$. For now we assume that $\sigma^2(\mathbf{x}_t)$ is known; later on we will estimate it from the data. The residuals are still assumed to be independent of each other, but they are no longer identically distributed. The Least Squares estimator remains unbiased, but its efficiency is impaired and the usual formulae for standard errors are inaccurate. The severity of this problem depends on the degree to which the conditional variances differ and the sample size of the data. Under heteroskedasticity, the Least Squares estimator is no longer BLUE; there exists an unbiased linear estimator with lower variance (Fox, 2008).

3.2 Weighted Linear Regression

Weighted Linear Regression solves the problem of heteroskedasticity. As was described towards the end of Section 3.1 we model $y_t = \mathbf{x}_t^\top \mathbf{u}^* + \epsilon_t$ with $\mathrm{Var}\left[y_t|\mathbf{x}_t\right] = \sigma_t^2 = \sigma^2(\mathbf{x}_t)$ where $\sigma^2(\mathbf{x}_t)$ is a known variance function, yielding the variance for round t.

We can transform our model back to a homoskedastic model by weighting the model by the square root of the variance function. We define the weight for round t as the inverse of the variance function: $\omega_t = \sigma^2(\mathbf{x}_t)^{-1}$. We now weight the heteroskedastic model and we obtain:

$$\begin{aligned} y_t\sqrt{\omega_t} &= \mathbf{x}_t^\top \sqrt{\omega_t}\mathbf{u}^* + \epsilon_t\sqrt{\omega_t} \\ y_t^\star &= (\mathbf{x}_t^\star)^\top \mathbf{u}^* + \epsilon_t^\star \end{aligned} \tag{3.1}$$

where $y_t^\star = y_t\sqrt{\omega_t}$, $\mathbf{x}_t^\star = \mathbf{x}_t\sqrt{\omega_t}$ and $\epsilon_t^\star = \epsilon_t\sqrt{\omega_t}$. If $y_t^\star$ is regressed on $\mathbf{x}_t^\star$ using Least Squares, we speak of Weighted Least Squares.

The conditional variance in round t $\mathrm{Var}\left[y_t|\mathbf{x}_t\right] = \sigma_t^2 = \sigma^2(\mathbf{x}_t)$ is made constant by weighting the conditional variance by the square root of the variance function:

$$\begin{aligned} \mathrm{Var}\left[\frac{\mathbf{y}_t}{\sigma(\mathbf{x}_t)}\middle|\mathbf{x}_t\right] &= \frac{1}{\sigma^2(\mathbf{x}_t)}\mathrm{Var}\left[y_t|\mathbf{x}_t\right] \\ &= \frac{1}{\sigma^2(\mathbf{x}_t)}\sigma^2(\mathbf{x}_t). \\ &= 1 \end{aligned} \tag{3.2}$$

The conditional variance in round t no longer depends on $\sigma^2(\mathbf{x}_t)$. We define the corresponding minimization problem:

$$\begin{aligned} \mathbf{w}_t^{WL} &= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{s=1}^{t-1} \omega_t \ell_t(\mathbf{w}_t). \\ &= (\mathbf{A}_t^{WL})^{-1} \mathbf{b}_t^W \end{aligned} \tag{3.3}$$

where $\mathbf{A}_t^{WL} = \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s \mathbf{x}_s^T$ and $\mathbf{b}_t^W = \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s y_s$.

The OWLS algorithm is found in Algorithm 3.1.

Algorithm 3.1 Online Weighted Least Squares (OWLS)

```
for $t = 1, 2, ..., T$ do
    $\mathbf{A}_t^{WL} = \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s \mathbf{x}_s^T$
    $\mathbf{b}_t^W = \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s y_s$
    $\mathbf{w}_t^{WL} = (\mathbf{A}_t^{WL})^{-1} \mathbf{b}_t^W$
    $\ell_t(\mathbf{w}_t) = (y_t - \mathbf{x}_t^\top \mathbf{w}_t^{WL})^2$
end for
```

3.3 Weighted Ridge Regression

In Section 2.3, we introduced ORR as an algorithm that has a better Regret bound than the OnLS algorithm. In Statistics, the use of Ridge Regression is often motivated in terms of its Mean Squared Error (MSE). For MSE improvement, bias is introduced in the model in exchange for variance; the quality of an estimator is determined by its bias and its variance. It is shown there exists a λ for which the Ridge estimator has a lower MSE than the Least Squares estimator (van Wieringen, 2015). The minimization problem for Ridge regression is found in Equation 2.3.

Weighting and regularization are combined in Weighted Ridge Regression. Just as there is a λ for which the Ridge estimator has a lower MSE than the Least Squares estimator, there is also a λ for which the Weighted Ridge estimator has lower MSE than the Weighted Least Squares estimator (Askin and Montgomery, 1980). We can find an expression for the Weighted Ridge estimator as follows:

$$\begin{aligned} \mathbf{w}_t^{WR} &= \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{s=1}^{t-1} \omega_t \ell_t(\mathbf{w}_t) + \lambda \|\mathbf{w}\|_2^2 \\ &= (\mathbf{A}_t^{WR})^{-1} \mathbf{b}_t^W \end{aligned} \tag{3.4}$$

where $\mathbf{A}_t^{WR} = \lambda \boldsymbol{I} + \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s \mathbf{x}_s^T$ and $\mathbf{b}_t^W = \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s y_s$.

The OWRR algorithm is found in Algorithm 3.2.

Algorithm 3.2 Online Weighted Ridge Regression (OWRR)

```
for $t = 1, 2, ..., T$ do
    $\mathbf{A}_t^{WR} = \lambda \boldsymbol{I} + \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s \mathbf{x}_s^T$
    $\mathbf{b}_t^W = \sum_{s=1}^{t-1} \omega_s \mathbf{x}_s y_s$
    $\mathbf{w}_t^{WR} = (\mathbf{A}_t^{WR})^{-1} \mathbf{b}_t^W$
    $\ell_t(\mathbf{w}_t) = (y_t - \mathbf{x}_t^\top \mathbf{w}_t^{WR})^2$
end for
```

3.4 Iterative Reweighted Ridge Regression

In Section 3.2 we assumed that variance function $\sigma^2(\mathbf{x})$ was known. As this variance function is often not known *a priori*, we would have to approximate it from the data. A way to do so is by application of Iterative Reweighted Ridge Regression (IRRR), which is Iterative Reweighted Least Squares (Carroll and Ruppert, 1988) with the Ridge loss function instead of the squared loss. This will further be explained in the next paragraph. As implied by the name, this algorithm is iterative with the iteration denoted by j. We define the estimator of the variance function $\hat{\sigma}^2(\mathbf{x})$ and we define $\hat{\omega}_t = \hat{\sigma}^2(\mathbf{x}_t)^{-1}$. We assume a linear variance function so that $\hat{\sigma}^2(\mathbf{x}_t) = |\mathbf{x}_t^\top \hat{\boldsymbol{\nu}}|$ where $\hat{\sigma}^2 : \mathbf{x} \to \mathbb{R}$ and where the absolute value is taken to ensure that the variance is positive.

IRRR proceeds as follows. All weights are initialized to one. The Weighted Ridge estimator is used to estimate $\hat{\epsilon}_t = (y_t - \mathbf{x}_t^\top \mathbf{w}^{WR})$. We find $\hat{\boldsymbol{\nu}}$ by application of Ridge Regression of $f(\hat{\epsilon}_t)$ on $\mathbf{x}_t$. We then use $\hat{\sigma}^2(\mathbf{x}_t) = |\mathbf{x}_t^\top \hat{\boldsymbol{\nu}}|$ to update the weights and the process is repeated until the algorithm converges, meaning that the difference between the sum of the weights of iteration $j + 1$ and that of iteration j changes with δ or less. A common option for f is $f(\hat{\epsilon}_t) = \hat{\epsilon}_t^2$, as the expectation of the squared residuals approximately corresponds to the variance. If regressing the squared residuals on $\mathbf{x}_t$ does not seem to reduce the heteroskedasticity, we find an alternative in $f(\hat{\epsilon}_t) = \sqrt{|\hat{\epsilon}_t|}$. Although the square root of the absolute values of the residuals does not model the variance directly, it models something that correlates with the variance. The square root of the absolute values of the residuals is less susceptible to outliers, as are the squared residuals (Carroll and Ruppert, 1988).

There are two reasons for using IRRR instead of Iterative Reweighted Least Squares. The first reason is that, in contrast with $\mathbf{w}^{WL}$, $\mathbf{w}^{WR}$ is always defined; even when $t < d$. We can thus learn the variance function from the first round onward. Secondly, there is a λ for which the MSE of estimator $\mathbf{w}^{WR}$ is lower than the MSE of $\mathbf{w}^{WL}$ (Askin and Montgomery, 1980). Exchanging variance for bias might work especially well for low t, as the variance of the estimator tends to be large when little data are available.

Chapter 4

Is weighting always better?

Until now we have laid out the framework for weighting in the OCO context. We hypothesize that we can improve the predictions of Online Least Squares (OnLS) and Online Ridge Regression (ORR) by weighting, if heteroskedasticity is present. This chapter is only dedicated to the comparison of ORR and OWRR. We already know that $\mathbf{w}^{WL}$ is BLUE in the presence of heteroskedasticity (Clapham and Nicholson, 2009). In this chapter we are making statistical assumptions that are unusual in OLR, as OLR is usually desired to be as free from assumptions as possible. The first assumption is that we model $y_t = \mathbf{x}_t^\top \mathbf{u}^* + \epsilon_t$. The second assumption is $\text{Var}\left[y_t|\mathbf{x}_t\right] = \sigma_t^2 = \sigma^2(\mathbf{x}_t)$. All expectations in this chapter are taken over y_t. It is important to note that the derivations in this chapter hold for vector $\mathbf{u}^*$ and not for any $\mathbf{u} \subseteq \mathbb{R}^d$.

We do not know how to optimally determine the value of penalization factor λ. A simple solution would be setting $\lambda = 1$, which gives the Regret bound found in Equation 2.4 (Cesa-Bianchi and Lugosi, 2006). In Ridge Regression, we put a constraint on the squared Euclidean norm of $\mathbf{w}$. In Ridge Regression, penalization is not applied equally over the elements of $\mathbf{w}$; larger regression coefficients are penalized more than smaller ones. This implies that, unlike the Least Squares estimator, the Ridge estimator is not invariant to the scaling of $\mathbf{x}_t$. The choice for λ thus depends on the scaling of $\mathbf{x}_t$. In Statistics, we normally solve this issue by scaling the predictors to have a mean of 0 and unit variance. In ORR and OWRR we instead scale λ with optimal parameters $\mathbf{u}^*$. We hereto define $\lambda = 1/\|\mathbf{u}^*\|_2^2$.

The first element we need is the expected Regret for OWRR and ORR, which are derived in Section 4.1. In Section 4.2 we then show that, when employing a single feature, OWRR is always at least as good as ORR. Then, in Section 4.3 we show that this is no longer true when two features are employed. In the two-feature setting, situations exist where ORR is better than OWRR. In Section 4.4 we back the theoretical finding of Section 4.3 up with data simulations. We will demonstrate that there are many situations where ORR results in lower expected Regret than OWRR with two features.

4.1 The expectation of the Regret

In this section we derive the expectations for OWRR and ORR. It is not common to calculate the expectation of the Regret in OLR. We can only do so by putting assumptions on the generation of y_t (see introduction of this chapter), that are not common in OCO.

We define $\overline{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{t-1})^\top$, $\mathbf{\Omega} = diag(\omega_1, ..., \omega_{t-1})$ and $\overline{\mathbf{y}} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{t-1})^\top$. We remind the reader that $\text{Regret}_T(\mathbf{u}^*) = \sum_{t=1}^T \ell_t(\mathbf{w}_t) - \sum_{t=1}^T \ell_t(\mathbf{u}^*)$, that $\ell_t(\mathbf{w}_t) = (\mathbf{x}_t^\top \mathbf{w}_t - y_t)^2$ and $\lambda = 1/\|\mathbf{u}^*\|_2^2$. We calculate the expected Regret, which is formulated as $\mathbb{E}\left[\text{Regret}_T(\mathbf{u}^*)\right] = \mathbb{E}\left[\sum_{t=1}^T \ell_t(\mathbf{w}_t) - \sum_{t=1}^T \ell_t(\mathbf{u}^*)\right]$. We drop the summation in the definition of $\mathbb{E}\left[\text{Regret}_T(\mathbf{u}^*)\right]$ and we continue:

$$\begin{aligned}
\mathbb{E}\left[\ell_t(\mathbf{w}_t^{WR}) - \ell_t(\mathbf{u}^*)\right] &= \mathbb{E}\left[(\mathbf{x}_t^\top \mathbf{w}_t^{WR} - y_t)^2 - (\mathbf{x}_t^\top \mathbf{u}^* - y_t)^2\right] \\
&= \mathbb{E}\left[(\mathbf{x}_t^\top \mathbf{w}_t^{WR})^2 - 2\mathbf{x}_t^\top \mathbf{w}_t^{WR} y_t + y_t^2\right] \\
&- \mathbb{E}\left[(\mathbf{x}_t^\top \mathbf{u}^*)^2 - 2\mathbf{x}_t^\top \mathbf{u}^* y_t + y_t^2\right] \\
&= \mathbf{x}_t^\top \mathbb{E}\left[\mathbf{w}_t^{WR}(\mathbf{w}_t^{WR})^\top\right]\mathbf{x}_t - 2\mathbf{x}_t^\top \mathbb{E}\left[y_t \mathbf{w}_t^{WR}\right] + \mathbb{E}\left[y_t^2\right] \\
&- \mathbf{x}_t^\top \mathbf{u}^*(\mathbf{u}^*)^\top \mathbf{x}_t - 2\,\mathbb{E}\left[y_t\right]\mathbf{x}_t^\top \mathbf{u}^* - \mathbb{E}\left[y_t^2\right] \\
&= \mathbf{x}_t^\top (\text{Var}\left[\mathbf{w}_t^{WR}\right] + \mathbb{E}\left[\mathbf{w}_t^{WR}\right]\mathbb{E}\left[\mathbf{w}_t^{WR}\right]^\top)\mathbf{x}_t - 2\mathbf{x}_t^\top \mathbf{u}^*\,\mathbb{E}\left[\mathbf{w}_t^{WR}\right]^\top \mathbf{x}_t \\
&- (\mathbf{x}_t^\top \mathbf{u}^*(\mathbf{u}^*)^\top \mathbf{x}_t - 2\mathbf{x}_t^\top \mathbf{u}^*(\mathbf{u}^*)^\top \mathbf{x}_t) \\
&= \mathbf{x}_t^\top (\text{Var}\left[\mathbf{w}_t^{WR}\right] + \mathbb{E}\left[\mathbf{w}_t^{WR}\right]\mathbb{E}\left[\mathbf{w}_t^{WR}\right]^\top \\
&- 2\mathbf{u}^*\,\mathbb{E}\left[\mathbf{w}_t^{WR}\right]^\top + \mathbf{u}^*(\mathbf{u}^*)^\top)\mathbf{x}_t.
\end{aligned} \tag{4.1}$$

We furthermore know that $\text{Var}\left[\mathbf{w}_t^{WR}\right] = (\mathbf{A}_t^{WR})^{-1}\overline{\mathbf{x}}^\top \mathbf{\Omega}\,\text{Var}\left[\overline{\mathbf{y}}\right]\mathbf{\Omega}\overline{\mathbf{x}}(\mathbf{A}_t^{WR})^{-1}$ and $\mathbb{E}\left[(\mathbf{w}_t^{WR})\right] = (\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \mathbf{\Omega}\,\overline{\mathbf{x}}\,\mathbf{u}^*)$ with $\mathbf{A}^{WR} = \overline{\mathbf{x}}^\top \mathbf{\Omega}\,\overline{\mathbf{x}} + \lambda \boldsymbol{I}$ (van Wieringen, 2015). Filling in the mean and the variance of $\mathbf{w}_t^{WR}$ and using that

$\mathrm{Var}\,[\overline{\mathbf{y}}] = \boldsymbol{\Omega}^{-1}$ we obtain:

$$\begin{aligned}
\mathbb{E}\,[\ell_t(\mathbf{w}_t^{WR}) - \ell_t(\mathbf{u}^*)] &= \mathbf{x}_t^\top \Big(\mathrm{Var}\,[\mathbf{w}_t^{WR}] + \mathbb{E}\,[\mathbf{w}_t^{WR}]\,\mathbb{E}\,[\mathbf{w}_t^{WR}]^\top \\
&\quad - 2\mathbf{u}^*\,\mathbb{E}\,[\mathbf{w}_t^{WR}]^\top + \mathbf{u}^*(\mathbf{u}^*)^\top \Big)\mathbf{x}_t \\
&= \mathbf{x}_t^\top \Big((\mathbf{A}_t^{WR})^{-1}\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\mathrm{Var}\,[\overline{\mathbf{y}}]\boldsymbol{\Omega}\overline{\mathbf{x}}(\mathbf{A}_t^{WR})^{-1}) \\
&\quad + \big((\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\,\mathbf{u}^*)\big)\big((\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\,\mathbf{u}^*)\big)^\top \\
&\quad - 2\mathbf{u}^*\big((\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\,\mathbf{u}^*)\big)^\top + \mathbf{u}^*(\mathbf{u}^*)^\top \Big)\mathbf{x}_t \\
&= \mathbf{x}_t^\top \Big((\mathbf{A}_t^{WR})^{-1}\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\overline{\mathbf{x}}(\mathbf{A}_t^{WR})^{-1} \\
&\quad + \big((\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\,\mathbf{u}^*)\big)\big((\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\,\mathbf{u}^*)\big)^\top \\
&\quad - 2\mathbf{u}^*\big((\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\,\mathbf{u}^*)\big)^\top + \mathbf{u}^*(\mathbf{u}^*)^\top \Big)\mathbf{x}_t
\end{aligned} \tag{4.2}$$

We now introduce the Woodbury Matrix Identity (Golub and Van Loan, 1996) of $(\mathbf{A}^{WR})^{-1}$, given as $(\mathbf{A}^{WR})^{-1} = \frac{1}{\lambda}\boldsymbol{I} - \frac{1}{\lambda}\boldsymbol{I}\,\overline{\mathbf{x}}\,(\boldsymbol{\Omega}^{-1} + \frac{1}{\lambda}\,\overline{\mathbf{x}}\,\overline{\mathbf{x}}^\top)\,\overline{\mathbf{x}}\,\frac{1}{\lambda}\boldsymbol{I}$. We can now write the expectation as:

$$\mathbb{E}\,[\ell_t(\mathbf{w}_t^{WR}) - \ell_t(\mathbf{u}^*)] = \mathbf{x}_t^\top(\mathbf{A}_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}} + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top)(\mathbf{A}_t^{WR})^{-1}\mathbf{x}_t \tag{4.3}$$

The expected Regret for ORR is derived in a similar way. The difference is that $\mathbb{E}\,[\mathbf{w}_t^R]$ does not include $\boldsymbol{\Omega}$, so that $\mathrm{Var}\,[\overline{\mathbf{y}}]$ does not cancel as happens in Equation 4.2. We write the expectation for ORR as :

$$\begin{aligned}
\mathbb{E}\,[\ell_t(\mathbf{w}_t^R) - \ell_t(\mathbf{u}^*)] &= \mathbf{x}_t^\top(\mathbf{A}_t^R)^{-1}(\overline{\mathbf{x}}^\top\,\mathrm{Var}\,[\overline{\mathbf{y}}]\,\overline{\mathbf{x}} + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top)(\mathbf{A}_t^R)^{-1}\mathbf{x}_t \\
&= \mathbf{x}_t^\top(\mathbf{A}_t^R)^{-1}(\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\,\overline{\mathbf{x}} + \lambda^2(\mathbf{u}^*)^\top\mathbf{u}^*)(\mathbf{A}_t^R)^{-1}\mathbf{x}_t
\end{aligned} \tag{4.4}$$

4.2 One feature

We show that OWRR has a lower expected loss than ORR when employing one feature. In order to do so, we need to show that this is true for all $x_1, x_2, ..., x_t$.

Theorem 1. *Suppose* $\mathrm{Var}\,[y_t|\boldsymbol{x}_t] = \sigma^2(\boldsymbol{x}_t)$*, then for* $\lambda = u^{-2}$ *and* $d = 1$ *we have* $\mathbb{E}\,[\ell_t(w_t^{WR}] \leq \mathbb{E}\,[\ell_t(w_t^R)] \quad \forall t$ *.*

Proof. We start with simplification of $\mathbb{E}\left[\ell_t(w_t^{WR})\right]$:

$$\begin{aligned}\mathbb{E}\left[\ell_t(w_t^{WR})\right] &= x_t\big((A_t^{WR})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\, + \lambda^2 u^2)(A_t^{WR})^{-1}\big)x_t + \mathbb{E}\left[y_t^2\right] \\ &= x_t^2(A_t^{WR})^{-2}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}\,\overline{\mathbf{x}}\, + \lambda^2 u^2) + \mathbb{E}\left[y_t^2\right] \\ &= (A_t^{WR})^{-1}x_t^2 + \mathbb{E}\left[y_t^2\right].\end{aligned} \tag{4.5}$$

Then we simplify $\mathbb{E}\left[\ell_t(w^R)\right]$:

$$\begin{aligned}\mathbb{E}\left[\ell_t(w_t^{R})\right] &= x_t\big((A_t^{R})^{-1}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda^2 u^2)(A_t^{R})^{-1}\big)x_t + \mathbb{E}\left[y_t^2\right] \\ &= x_t^2(A_t^{R})^{-2}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda) + \mathbb{E}\left[y_t^2\right]\end{aligned} \tag{4.6}$$

We compare both expectations and we obtain:

$$\begin{aligned}\mathbb{E}\left[\ell_t(w_t^{WR})\right] - \mathbb{E}\left[\ell_t(w_t^{R})\right] &= x_t^2(A_t^{WR})^{-1} + \mathbb{E}\left[y_t^2\right] \\ &\quad - \big(x_t^2(A_t^{R})^{-2}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda) + \mathbb{E}\left[y_t^2\right]\big) \\ &= x_t^2(A_t^{WR})^{-1} - x_t^2(A_t^{R})^{-2}(\overline{\mathbf{x}}^\top \boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda)\end{aligned} \tag{4.7}$$

For completion we have to verify if the expression in Equation 4.7 is always negative:

$$\begin{aligned}x_t^2(A_t^{WR})^{-1} - x_t^2(A_t^{R})^{-2}(\overline{\mathbf{x}}\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}}^\top + \lambda) &\leq 0 \\ (A_t^{WR})^{-1} &\leq (A_t^{R})^{-2}(\overline{\mathbf{x}}\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}}^\top + \lambda) \\ (A_t^{R})^{2} &\leq A_t^{WR}(\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda) \\ (\overline{\mathbf{x}}^\top\overline{\mathbf{x}})^2 + 2\lambda\overline{\mathbf{x}}^\top\overline{\mathbf{x}} + \lambda^2 &\leq \overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}}\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda\overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}} + \lambda\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda^2 \\ (\overline{\mathbf{x}}^\top\overline{\mathbf{x}})^2 + 2\lambda\overline{\mathbf{x}}^\top\overline{\mathbf{x}} &\leq \overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}}\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} + \lambda\overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}} + \lambda\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}}\end{aligned} \tag{4.8}$$

We split the last inequality in Equation 4.8 in two parts. First:

$$(\overline{\mathbf{x}}^\top\overline{\mathbf{x}})^2 \leq \overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}}\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}}. \tag{4.9}$$

We define the norm $\|\overline{\mathbf{x}}\,\| = \sqrt{\overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}}}$, its dual norm is $\|\overline{\mathbf{x}}\,\|_* = \sqrt{\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}}}$

and it then follows from Hölder's inequality (Kuptsov, 2001) that $|\overline{\mathbf{x}}^\top \overline{\mathbf{x}}| \leq \|\overline{\mathbf{x}}\|\|\overline{\mathbf{x}}\|_*$, which implies Inequality 4.9.

We write the second part of the last inequality of Equation 4.8:

$$\begin{aligned} 2\lambda\overline{\mathbf{x}}^\top\overline{\mathbf{x}} &\leq \lambda\overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}} + \lambda\overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} \\ \iff 2\overline{\mathbf{x}}^\top\overline{\mathbf{x}} &\leq \overline{\mathbf{x}}^\top\boldsymbol{\Omega}\overline{\mathbf{x}} + \overline{\mathbf{x}}^\top\boldsymbol{\Omega}^{-1}\overline{\mathbf{x}} \\ \iff \overline{\mathbf{x}}^\top\mathbf{Q}\overline{\mathbf{x}} &\leq 0 \end{aligned} \tag{4.10}$$

with $\mathbf{Q} = (2\boldsymbol{I} - \boldsymbol{\Omega} - \boldsymbol{\Omega}^{-1})$.

In order for the inequality in Equation 4.10 to hold, matrix $\mathbf{Q}$ needs to be negative semi-definitive. As $\mathbf{Q}$ is a diagonal matrix, it suffices to show that all elements $\mathbf{Q}_{i,i} \leq 0$. Reformulate this inequality as $2 - \omega_i - \frac{1}{\omega_i} \leq 0$. This is a concave function with maximum of 0 attained at 1. It is thus shown that the last inequality in Equation 4.8 holds and this concludes the proof. □

4.3 Two features

In this section we will search for a simple situation where OWRR has a higher expected Regret than ORR. We will refer to this situation as a counter-example. We will calculate the losses for two rounds and we assume that $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}$ are the same. As we calculate the losses for two rounds, the only weight we have to incorporate is the weight of the first round, which is defined as $\omega_1 = \frac{1}{\sigma_1^2}$. Using identical $\mathbf{x}_1$ and $\mathbf{x}_2$ makes the setting in fact homoskedastic. We will generalize the counter-example to heteroskedasticity in Section 4.4 using data simulations.

Theorem 2. *Let $\mathbb{E}\left[\ell_2(\boldsymbol{w}^{WR})\right]$ and $\mathbb{E}\left[\ell_2(\boldsymbol{w}^{R})\right]$ be the expectations of the loss in the second round of the loss for the Online Weighted Ridge and Online Ridge algorithms respectively. Then for $\lambda = 1/\|\boldsymbol{u}^*\|_2^2$, $\boldsymbol{x}^\top\boldsymbol{u}^* = 0$ and $d = 2$ we have $\mathbb{E}\left[\ell_2(\boldsymbol{w}^{WR}\right] \leq \mathbb{E}\left[\ell_2(\boldsymbol{w}^{R})\right]$ if and only if:*

$$\left(\frac{1}{\sigma_1^2} - \sigma_1^2\right)\frac{1}{\lambda^2} + 2\left(\gamma^R\sigma_1^2 - \frac{\gamma^{WR}}{\sigma_1^2}\right)\frac{1}{\lambda}(\boldsymbol{x}^\top\boldsymbol{x}) + \left(\frac{1}{\sigma_1^2}\gamma^{2,WR} - \sigma_1^2\gamma_R^2\right)(\boldsymbol{x}^\top\boldsymbol{x})^2 \leq 0 \tag{4.11}$$

with $\gamma^{WR} = \frac{1}{\sigma_1^2\lambda^2(1+\frac{1}{\sigma_1^2\lambda}\mathbf{x}^\top\mathbf{x})}$ and $\gamma^R = \frac{1}{\lambda^2(1+\frac{1}{\lambda}\mathbf{x}^\top\mathbf{x})}$.

We fill in $\mathbf{x} = (1,2)^\top, \mathbf{u}^* = (1,-1/2)^\top$ and $\sigma_1^2 = 0.5$. We obtain a positive outcome of 0.006.

Proof. We again start by deriving $\mathbb{E}\left[\ell_2(\mathbf{w}_t^{WR})\right]$:

$$\begin{aligned}
\mathbb{E}\left[\ell_2(\mathbf{w}_t^{WR})\right] &= \mathbb{E}\left[(\mathbf{x}^\top \mathbf{w}_2^{WR} - y_2)^2\right] \\
&= \mathbb{E}\left[(\mathbf{x}^\top \mathbf{w}_2^{WR})^2 - 2y_2\mathbf{x}^\top \mathbf{w}_2^{WR} + y_2^2)\right] \\
&= \mathbf{x}^\top\, \mathbb{E}\left[\mathbf{w}_2^{WR}\mathbf{w}_2^\top\right]\mathbf{x} - 2y_2\mathbf{x}^\top\, \mathbb{E}\left[\mathbf{w}_2^{WR}\right] + \mathbb{E}\left[y_2\right] \\
&= \mathbf{x}^\top(\mathrm{Var}\left[\mathbf{w}_2^{WR}\right] + \mathbb{E}\left[\mathbf{w}_2^{WR}\right]\mathbb{E}\left[\mathbf{w}_2^{WR}\right]^\top)\mathbf{x} - 2y_2\mathbf{x}^\top\, \mathbb{E}\left[\mathbf{w}_2^{WR}\right] \\
&= \mathbf{x}^\top(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda\boldsymbol{I})^{-1}(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top)(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda\boldsymbol{I})^{-1}\mathbf{x}.
\end{aligned} \tag{4.12}$$

The above definition includes two identical matrix inverses. As both matrices are two-dimensional (square), we can calculate their inverses efficiently. We do so by application of the Woodbury Matrix Identity (Golub and Van Loan, 1996):

$$\begin{aligned}
(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda\boldsymbol{I})^{-1} &= \frac{1}{\lambda}\boldsymbol{I} - \frac{\mathbf{x}\mathbf{x}^\top}{\sigma_1^2\lambda^2(1 + \frac{1}{\sigma_1^2\lambda}\mathbf{x}\mathbf{x}^\top)} \\
&= \frac{1}{\lambda}\boldsymbol{I} - \gamma^{WR}\mathbf{x}\mathbf{x}^\top.
\end{aligned} \tag{4.13}$$

We reformulate the expression for $\mathbb{E}\left[\ell_2(\mathbf{w}_t^{WR}\right]$:

$$\begin{aligned}
\mathbb{E}\left[\ell_2(\mathbf{w}_t^{WR}\right] =& \mathbf{x}^\top\left(\left(\frac{1}{\lambda}\boldsymbol{I} - \gamma^{WR}\mathbf{x}\mathbf{x}^\top\right)\left(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top\right)\right. \\
&\left.\left(\frac{1}{\lambda}\boldsymbol{I} - \gamma^{WR}\mathbf{x}\mathbf{x}^\top\right)\right)\mathbf{x}.
\end{aligned} \tag{4.14}$$

We expand the expression within the inner brackets. Hereto we define the following variables:

$$\begin{aligned}
A &= \frac{1}{\lambda} \\
B &= \frac{\mathbf{x}\mathbf{x}^\top}{\sigma_1^2\lambda^2(1 + \frac{1}{\sigma_1^2\lambda}\mathbf{x}^\top\mathbf{x})} \\
&= \gamma^{WR}\mathbf{x}\mathbf{x}^\top \\
C &= (\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top).
\end{aligned} \tag{4.15}$$

Replacing the different parts in Equation 4.14, we obtain the following:

$$\begin{aligned}\mathbb{E}\left[\ell_2(\mathbf{w}_t^{WR})\right] &= \mathbf{x}^\top(A-B)C(A-B)\mathbf{x} \\ &= \mathbf{x}^\top(ACA - ACB - BCA + BCB)\mathbf{x} \\ &= \mathbf{x}^\top(ACA - 2ACB + BCB)\mathbf{x}.\end{aligned} \tag{4.16}$$

We derive ACA, ACB and BCB seperately.

$$\begin{aligned}\mathbf{x}^\top ACA\mathbf{x} &= \mathbf{x}_2^\top\left(\frac{1}{\lambda}\left(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}_2^\top + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top\right)\frac{1}{\lambda}\right)\mathbf{x} \\ &= \frac{1}{\lambda^2}\mathbf{x}^\top\left(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top\right)\mathbf{x} \\ &= \frac{1}{\lambda^2\sigma_1^2}(\mathbf{x}^\top\mathbf{x})^2 + (\mathbf{x}^\top\mathbf{u}^*)^2\end{aligned} \tag{4.17}$$

$$\begin{aligned}\mathbf{x}_2^\top ACB\mathbf{x} &= \mathbf{x}^\top\left(\frac{1}{\lambda}\gamma_{WR}\mathbf{x}\mathbf{x}^\top(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top)\right)\mathbf{x} \\ &= \mathbf{x}^\top\left(\frac{1}{\lambda\sigma_1^2}\gamma_{WR}(\mathbf{x}\mathbf{x}^\top\mathbf{x}\mathbf{x}^\top) + \frac{1}{\lambda}\gamma_{WR}\lambda^2\mathbf{x}\mathbf{x}^\top\mathbf{u}^*(\mathbf{u}^*)^\top\right)\mathbf{x} \\ &= \frac{1}{\lambda\sigma_1^2}\gamma_{WR}(\mathbf{x}^\top\mathbf{x})^3 + \frac{1}{\lambda}\gamma_{WR}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2\mathbf{x}^\top\mathbf{x}\end{aligned} \tag{4.18}$$

$$\begin{aligned}\mathbf{x}^\top BCB\mathbf{x} &= \mathbf{x}^\top\left(\gamma_{WR}\mathbf{x}\mathbf{x}^\top(\frac{1}{\sigma_1^2}\mathbf{x}\mathbf{x}^\top + \lambda^2\mathbf{u}^*(\mathbf{u}^*)^\top)\gamma^{WR}\mathbf{x}\mathbf{x}^\top\right)\mathbf{x} \\ &= \mathbf{x}^\top\left(\frac{1}{\sigma_1^2}\gamma^{2,WR}(\mathbf{x}\mathbf{x}^\top)^3 + \gamma^{2,WR}\lambda^2(\mathbf{x}\mathbf{x}^\top)^2\mathbf{u}^*(\mathbf{u}^*)^\top\right)\mathbf{x} \\ &= \frac{1}{\sigma_1^2}\gamma^{2,WR}(\mathbf{x}^\top\mathbf{x}_2)^4 + \gamma^{2,WR}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2(\mathbf{x}^\top\mathbf{x})^2\end{aligned} \tag{4.19}$$

We substitute the derivations for ACA, ACB and BCB in Equation 4.16:

$$\begin{aligned}\mathbb{E}\left[\ell_2(\mathbf{w}_t^{WR})\right] = &\frac{1}{\lambda^2\sigma_1^2}(\mathbf{x}^\top\mathbf{x})^2 + (\mathbf{x}^\top\mathbf{u}^*)^2 \\ &- 2\bigg(\frac{1}{\lambda\sigma_1^2}\gamma^{WR}(\mathbf{x}^\top\mathbf{x})^3 + \frac{1}{\lambda}\gamma^{WR}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2\mathbf{x}^\top\mathbf{x}\bigg) \\ &+ \frac{1}{\sigma_1^2}\gamma^{2,WR}(\mathbf{x}^\top\mathbf{x})^4 + \gamma^{2,WR}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2(\mathbf{x}^\top\mathbf{x})^2.\end{aligned} \tag{4.20}$$

Next, we compare the expectation above to the expectation of the loss of unweighted Online Ridge Regression: $\mathbb{E}\left[\ell_2(\mathbf{w}_t^R)\right]$. This expectation is found by replacing $1/\sigma^2$ by σ^2. In addition, we replace γ^{WR} by γ^R (see Theorem 2). We now formulate the expectation of the loss for ORR:

$$\begin{aligned}\mathbb{E}\left[\ell_2(\mathbf{w}_t^R)\right] = \frac{\sigma_1^2}{\lambda^2}(\mathbf{x}^\top\mathbf{x})^2 - 2\bigg(\frac{\sigma_1^2}{\lambda}\gamma^R(\mathbf{x}^\top\mathbf{x}_2)^3 + \frac{1}{\lambda}\gamma^R\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2\mathbf{x}^\top\mathbf{x}\bigg) \\ + \sigma_1^2\gamma^{2,R}(\mathbf{x}^\top\mathbf{x})^4 + \gamma^{2,R}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2(\mathbf{x}^\top\mathbf{x})^2.\end{aligned} \tag{4.21}$$

Like with a single feature, we can compare both expectations:

$$\begin{aligned}&\mathbb{E}\left[\ell_2(\mathbf{w}_t^{WR})\right] - \mathbb{E}\left[\ell_2(\mathbf{w}_t^R)\right] = \frac{1}{\lambda^2\sigma_1^2}(\mathbf{x}^\top\mathbf{x})^2 \\ &- 2\bigg(\frac{1}{\lambda\sigma_1^2}\gamma^{WR}(\mathbf{x}^\top\mathbf{x})^3 + \frac{1}{\lambda}\gamma^{WR}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2\mathbf{x}^\top\mathbf{x}\bigg) \\ &+ \frac{1}{\sigma_1^2}\gamma^{2,WR}(\mathbf{x}^\top\mathbf{x})^4 + \gamma^{2,WR}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2(\mathbf{x}^\top\mathbf{x})^2 \\ &- \bigg(\frac{\sigma_1^2}{\lambda^2}(\mathbf{x}^\top\mathbf{x})^2 - 2\bigg(\frac{\sigma_1^2}{\lambda}\gamma^R(\mathbf{x}^\top\mathbf{x})^3 + \frac{1}{\lambda}\gamma^R\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2\mathbf{x}^\top\mathbf{x}\bigg) \\ &+ \sigma_1^2\gamma^{2,R}(\mathbf{x}^\top\mathbf{x})^4 + \gamma^{2,R}\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2(\mathbf{x}_2^\top\mathbf{x})^2\bigg) \\ &= \bigg(\frac{1}{\sigma_1^2} - \sigma_1^2\bigg)\frac{1}{\lambda^2}(\mathbf{x}^\top\mathbf{x})^2 \\ &+ 2\bigg(\bigg(\gamma_R\sigma_1^2 - \frac{\gamma^{WR}}{\sigma_1^2}\bigg)\frac{1}{\lambda}(\mathbf{x}^\top\mathbf{x})^3 + (\gamma^R - \gamma^{WR})\lambda(\mathbf{x}^\top\mathbf{u}^*)^2(\mathbf{x}^\top\mathbf{x})\bigg) \\ &+ \bigg(\frac{1}{\sigma_1^2}\gamma^{2,WR} - \sigma_1^2\gamma^{2,R}\bigg)(\mathbf{x}^\top\mathbf{x})^4 + (\gamma^{2,WR} - \gamma^{2,R})\lambda^2(\mathbf{x}^\top\mathbf{u}^*)^2(\mathbf{x}^\top\mathbf{x})^2.\end{aligned} \tag{4.22}$$

Assuming orthogonality between $\mathbf{x}$ and $\mathbf{u}^*$ we can reduce Equation 4.22 to:

$$\begin{aligned}\mathbb{E}\left[\ell_2(\mathbf{w}^{WR})\right] - \mathbb{E}\left[\ell_2(\mathbf{w}_t^{R})\right] &= \left(\frac{1}{\sigma_1^2} - \sigma_1^2\right)\frac{1}{\lambda^2}(\mathbf{x}^\top\mathbf{x})^2 \\ &\quad + 2\left(\gamma_R\sigma_1^2 - \frac{\gamma^{WR}}{\sigma_1^2}\right)\frac{1}{\lambda}(\mathbf{x}^\top\mathbf{x})^3 \qquad (4.23)\\ &\quad + \left(\frac{1}{\sigma_1^2}\gamma^{2,WR} - \sigma_1^2\gamma_R^2\right)(\mathbf{x}^\top\mathbf{x})^4\end{aligned}$$

from which the result follows. □

4.4 Data simulations

With one counter-example known, as was shown in Section 4.3, we enhanced our conceptual knowledge of counter-examples by performing data simulations. We simulated data for varying σ^2, $\mathbf{u}^*$ and $\mathbf{x}_t$, we executed ORR and OWRR and we searched for commonalities in the areas where OWRR performed worse than ORR.

For all experiments we sampled three $x_{t,2} \sim \mathcal{N}(\tilde{x}, s)$, where $\mathcal{N}$ is a Gaussian distribution with mean $\tilde{x}$ and variance $s = 1$ or $s = 0.01$, so that $\mathbf{x}_t = (x_{t,1}, x_{t,2})^\top$ where $x_{t,1} = 1$. Furthermore we defined $\sigma^2(\mathbf{x}_t) = x_{t,2}^c$, where c is the exponent of the variance function. An identical grid, with $\tilde{x}$ on the x-axis and c on the y-axis was used for all simulations. Values for c are in $[-1.5, 1.5]$ and for $\tilde{x}$ in $[-20, 20]$, with equal distances between the points. In total, all grids contained 2080 points. For every grid, a different vector $\mathbf{u}^*$ was chosen.

We started with $\mathbf{u}^* = (1, -\frac{1}{2})^\top$ to confirm whether the counter-example found in Section 4.3 also holds for $\mathbf{x}_1 \neq \mathbf{x}_2 \neq \mathbf{x}_3$. We then increased $\|\mathbf{u}^*\|_2^2$ by multiplying the elements of $\mathbf{u}^*$ by 100, forming $\mathbf{u}^* = (100, -50)^\top$, to verify whether the length of $\mathbf{u}^*$ made a difference on the location of the counter-examples. For examination of the effect of the direction of $\mathbf{u}^*$, we rotated vector $\mathbf{u}^* = (1, -\frac{1}{2})^\top$ to the y-axis without changing its length, forming $\mathbf{u}^* = (0, \sqrt{\frac{5}{4}})^\top$. We also rotated vector $\mathbf{u}^* = (100, -50)^\top$, in the same direction as before, forming $\mathbf{u}^* = (0, \sqrt{12500})^\top$.

All results are shown in heatmaps (Figure 4.1 - Figure 4.4). Losses for OWRR relative to ORR are found in the red areas. This relativity means that there is an upper boundary of 100, as OWRR can only peform 100% better than ORR before obtaining a Regret of 0. On the other hand, OWRR could perform infinitely worse than ORR; there's no lower boundary on the performance of OWRR relative to ORR. All heatmaps have their specific continuous scale which means that the scales are not comparable between the heatmaps. We can however, compare the locations of the counter-examples between the heatmaps.

Conform with our findings at the end of section 4.3, we found that ORR performs better than OWRR at $\mathbf{x} = (1, 2)^\top$ and $c = -1$ (thus $\sigma^2 = \frac{1}{2}$), which is read from Figure 4.1. The black cross in Figure 4.1 indicates the location of the theoretical counter-example. We confirm that the counter-example was also found for $s \neq 0$, i.e. when all $\mathbf{x}_t$ are not identical.

The area where $c > 0$ is mostly the area where $\sigma^2 > 1$ and where few counter-examples are found. The exception in the area $c > 0$ is where $|\tilde{x}| < 0$ and thus $\sigma^2 < 1$. The relative gain for OWRR grows with larger values for σ^2 and thus an increase in heteroskedasticity.

When the variance is close to being constant, around $c = 0$, the difference between OWRR and ORR was found to be small. The variance does either not or slightly depend on $\mathbf{x}_t$ in this area, which could indicate that one should be careful with application of OWRR when the variance is approximately homoskedastic. The reason is that both the counter-example of Section 4.3 and the heatmaps provide no evidence for OWRR doing well in the homoskedastic setting.

All counter-examples on every heatmap are found in the region where $\sigma^2 < 1$. Furthermore the location of the counter-examples depends on $s = \text{Var}[x_{t,2}]$, $\|\mathbf{u}^*\|_2^2$ and the direction of $\mathbf{u}^*$. We will discuss them individually.

1. s. Heteroskedasticity increases with larger s, thus fewer counter-examples should be found for $s = 1$ than for $s = 0.01$. This is supported by almost all heatmaps, but is a bit ambiguous for the heatmaps in Figure 4.1.

2. $\|\mathbf{u}^*\|_2^2$. The norm of $\mathbf{u}^*$ affects the performance of OWRR through $\lambda = 1/\|\mathbf{u}^*\|_2^2$ and through y_t. Larger $\|\mathbf{u}^*\|_2^2$ seems to increase the areas where counter-examples are found. With the current experimental set-up, it is not possible to clarify whether this effect arises through λ or through y_t. If the performance of OWRR is affected through λ then it is potentially

beneficial to define $\lambda^{WR} = \sigma^2/\|\mathbf{u}^*\|_2^2$ for weaker penalization where σ^2 is small. No problem arises from defining such a specific λ for OWRR, as σ^2 is already defined for this algorithm. It is a different story for ORR, where σ^2 remains defined.

3. The direction of $\mathbf{u}^*$. The direction of vector $\mathbf{u}^*$ affects the performance of OWRR through y_t. In our experimental set-up, more counter-examples are found when $\mathbf{u}^*$ moves vertical. This is specifically illustrated by Figure 4.4, where vector $\mathbf{u}^*$ has an x-coordinate of 0, lying on the y-axis. Even though the surface of found counter-examples is large, they are still only found in the area where $\sigma^2 < 1$.

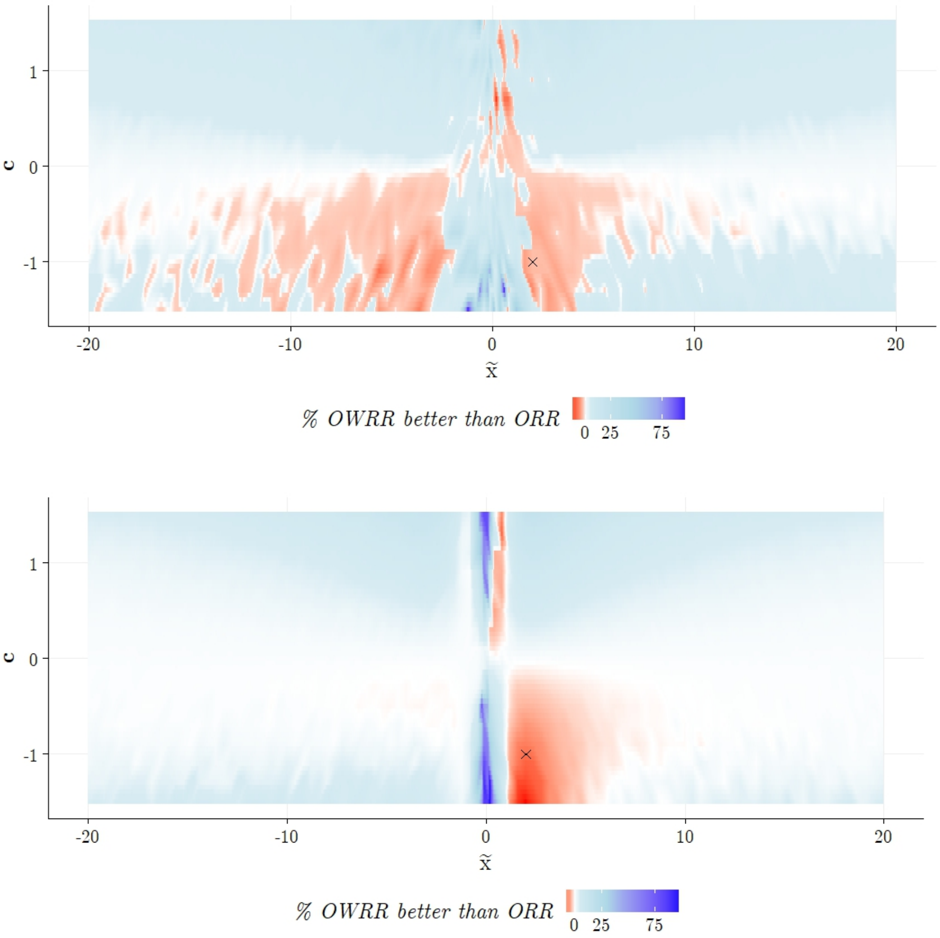

Figure 4.1: Heatmaps for random search experiments for varying c and $\tilde{\mathbf{x}}$. Vector $\mathbf{u}^* = (1, -\frac{1}{2})^\top$ and $\|\mathbf{u}^*\|_2^2 = \frac{5}{4}$. From top to bottom: $s = 1, = 0.01$. The black cross shows the location of the example found in Section 4.3.

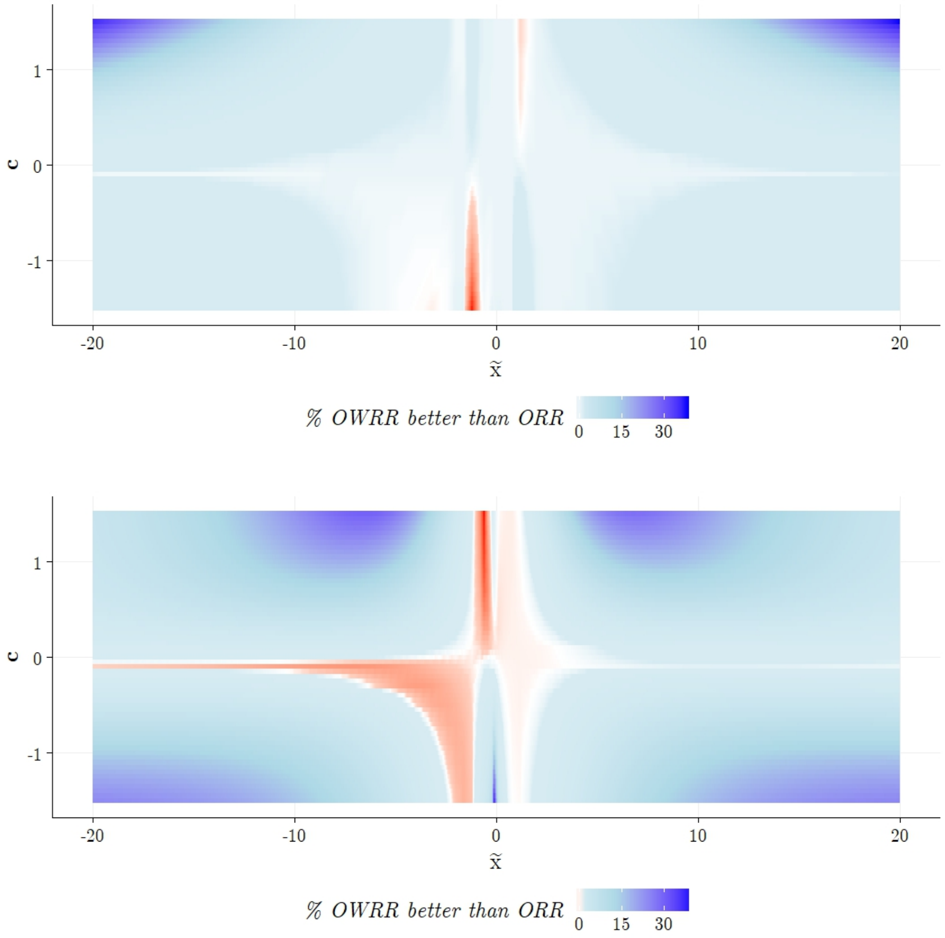

Figure 4.2: Heatmaps for random search experiments for varying c and $\tilde{\mathbf{x}}$. Vector $\mathbf{u}^* = (100, -50)^\top$ and $\|\mathbf{u}^*\|_2^2 = 12500$. From top to bottom: $s = 1$, $s = 0.01$.

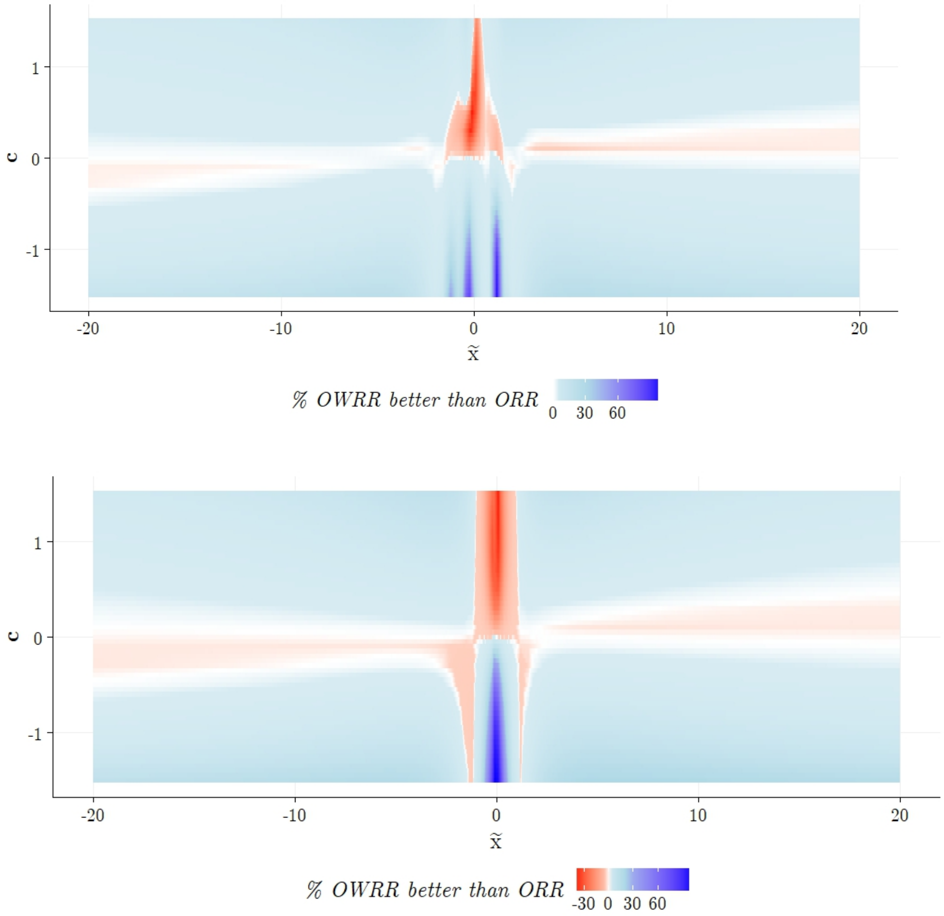

Figure 4.3: Heatmaps for random search experiments for varying c and $\tilde{\mathbf{x}}$. Vector $\mathbf{u}^* = (0, \sqrt{\frac{5}{4}})^\top$ and $\|\mathbf{u}^*\|_2^2 = \frac{5}{4}$. From top to bottom: $s = 1$, $s = 0.01$.

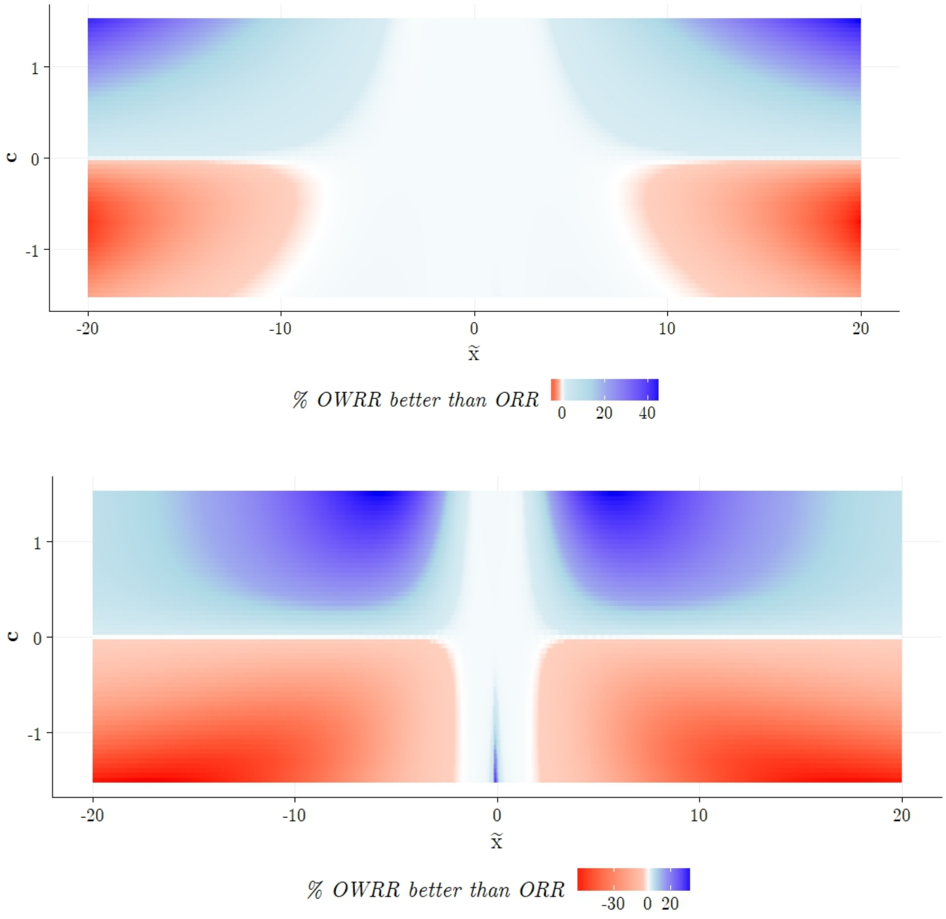

Figure 4.4: Heatmaps for random search experiments for varying c and $\tilde{\mathbf{x}}$. Vector $\mathbf{u}^* = (0, \sqrt{12500})^\top$ and $\|\mathbf{u}^*\|_2^2 = 12500$. From top to bottom: $s = 1$, $s = 0.01$.

4.5 Summary and conclusion

By Gaus-Markov theorem (Clapham and Nicholson, 2009), OWLR is always better than OLR. For the penalized versions of the algorithm, it was shown in Section 4.2 that OWRR is always better than ORR when employing a single feature. When employing two features, OWRR was not always better than ORR. A theoretical counter-example was derived in Section 4.3, and more counter-examples were found in Section 4.4 using data simulations.

With two features, OWRR was not always better than ORR as a consequence of σ^2, s and $\mathbf{u}^*$, with $\mathbf{u}^*$ affecting the performance of OWRR/ORR through λ and through y_t. To deal with counter-examples that arise from the effect of $\mathbf{u}^*$ on λ, a specific lambda for OWRR could prove to be beneficial: $\lambda^{WR} = \sigma^2 / \|\mathbf{u}^*\|_2^2$.

Chapter 5

Application on real-world data

Even though many counter examples of weighting in OLR were shown in Section 4.3 and Section 4.4, OWLS and OWRR do work in many situations. We can indeed exploit heteroskedastic noise and improve OnLS and ORR by including weights. In this chapter we will demonstrate these improvements on three different real-world datasets. We hereby use Iterative Reweighted Ridge Regression (IRRR) to estimate the variance function.

Before we introduce the datasets, we first discuss detection of heteroskedasticity in Section 5.1. Subsequently, in Section 5.2 we apply the weighted algorithms to real world datasets. For the first experiment,we model the number of cellphones within countries as a function of GDP. For the second experiment, we model the economic output of Belgian firms. In the third experiment, we model the housing prices in Boston . If we, for a specific dataset, make adjustments to the IRRR algorithm explained in Section 3.4, we elaborate on these adjustments in the specific section of that dataset. All findings are summarized and interpreted in Section 5.3.

5.1 Detecting heteroskedasticity

We detected heteroskedasticity with two tools: a residual plot and a hypothesis test. We found the residuals by running offline linear regression over all the data. We did not plot the residuals against any feature directly, as we were likely investigating more than one feature at a time. We also did not plot the residuals against the response, as $y_t = \mathbf{x}_t^\top \mathbf{u}^* + \epsilon_t$ and ϵ_t are correlated.

Instead, we plotted $\hat{\epsilon}_t$ against the prediction of y_t: $\mathbf{x}_t^\top \mathbf{u}^*$. To correct for leverage all residuals were studentized (Fox, 2008) and squared for a better visual interpretation. These Squared Studentized Residuals (SSR) were be plotted against the predicted values for two different scales in $\hat{\epsilon}_t$ and $\mathbf{x}_t^\top \mathbf{u}^*$.

A hypothesis test for heteroskedasticity detection is formulated by Breusch and Pagan. The Breusch-Pagan test was executed by running linear regression on all the data and subsequently obtaining the studentized residuals. The studentized residuals were then regressed on the features, after which we obtained the sum of squares that is explained by the model: the (ESS) (Verbeek, 2012). The test statistic was then defined as $\frac{1}{2}$ ESS, which is chi-square distributed with d degrees of freedom (Fox, 2008). We applied this test to all three datasets.

5.2 Experiments

All four algorithms (i.e. OnLS, ORR, OWLS, OWRR) were run on three datasets. For each dataset, three different experiments were conducted. In the first experiment, all algorithms were run with a variance function learned offline. In the second experiment, the variance function was learned online and weighting was applied from $t = 1$ onward. In the third experiment the variance function was also learned from $t = 1$ onward, but weighting was only applied from $t = 16$, giving the algorithms time to learn the variance function before using it. The results were averaged over 1200 repetitions. The threshold for convergence (Section 3.4) $\delta = 0.01$.

Countries

The data were collected from the US government (CIA). The data comprises 226 observations of 2 variables. The response variable is the total number of cellphones in a given country, with the feature being GDP (total, in USD). An intercept is added as well. The residual plots are shown in Figure 5.1, before and after weighting with the variance function learned offline. From this figure we read that the heteroskedasticity is caused by some outliers (observations with large SSR). Visually, the heteroskedasticity does not seem to persist after zooming in, as is seen in Figure 5.1. The Breusch-Pagan test was significant with p-value $< 2.42\text{e-}16$. After learning the variance function and weighing the observations we obtained $p = 1\text{e-}11$. The IRRR algorithm was not able to capture the heteroskedasticity completely, but it still gave us an advantage

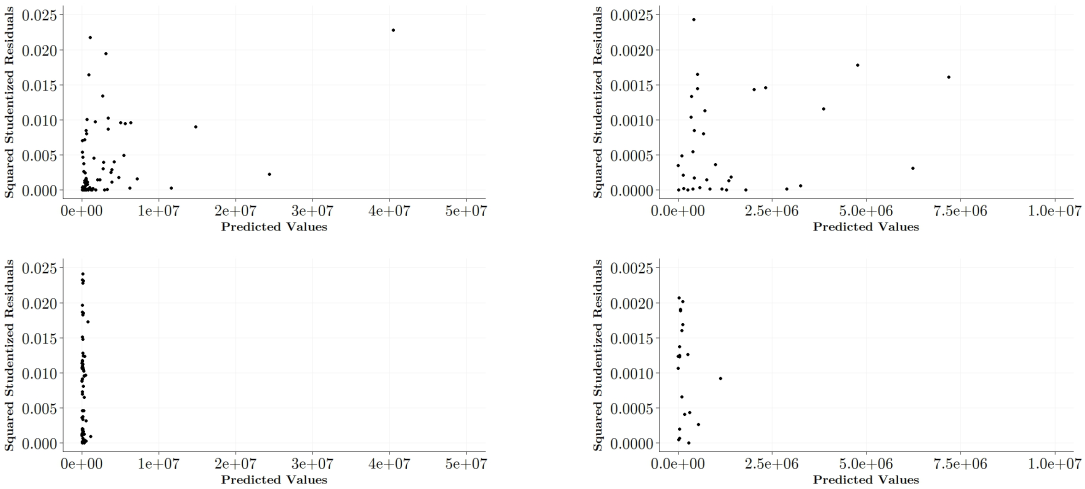

Figure 5.1: Countries. Squared Studentized Residuals against predicted values (Least Squares) for two different scales (left and right). Top: before weigthing. Bottom: after weighting.

over unweighted OLR. The Regret curves are found in Figure 5.4 with the relative Regret differences in the caption.

Belgian firms

The data were collected from an introductory guidebook to Econometrics (Verbeek, 2012). The data comprises 569 observations of 4 variables. The response is economic output (in euro's), with the features being capital (in euro's), average wage (in euro's) and labour (in number of people). An intercept is added. All variables are continuous. The residual plots are shown in Figure 5.2. From this Figure we read that the data are heteroskedastic on multiple scales: the heteroskedasticity persists even after we zoom in on the residuals. The Breusch-Pagan test was significant with p-value $<$ 2.2e-16. After learning the variance function and weighing the observations we obtained $p =$ 3.7e-06. The IRRR algorithm was not able to capture the heteroskedasticity completely, but weighting still gave us an advantage over unweighted OLR. The Regret curves are found in Figure 5.5.

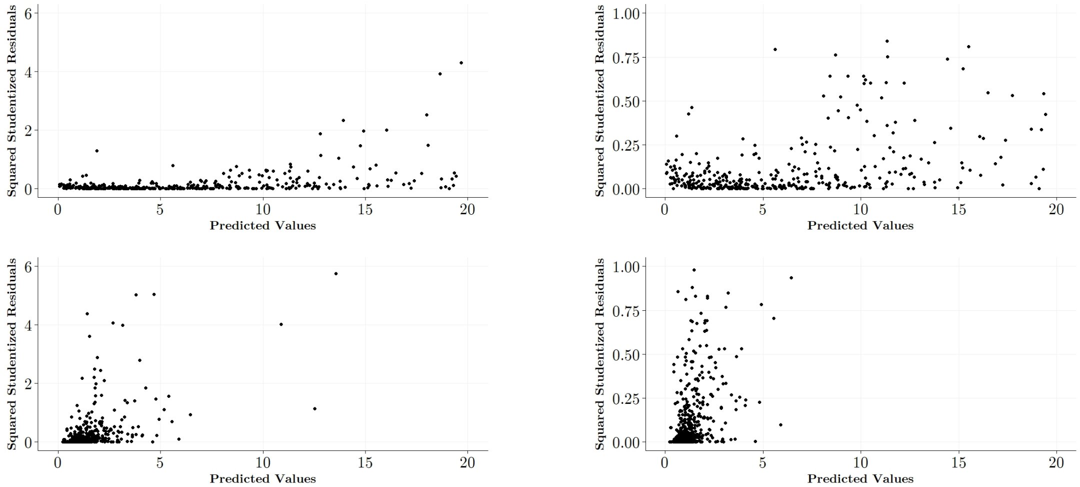

Figure 5.2: Belgian Firms. Squared Studentized Ressiduals against predicted values (Least Squares) for two different scales (left and right). Top: before weighting. Bottom: after weighting.

Boston housing data

The data were collected online from Kaggle.com. The data comprises 506 observations of 14 variables. The response is the median value of owner-occupied homes in USD, with the dataset further consisting of 13 features related to the neighborhood and to the property itself. Dummy variable (CHAS) was removed, as well as some other variable (RAD), which seemingly is categorical. The residual plots are shown in Figure 5.3, but the heteroscedasticity is difficult to read graphically. The Breusch-Pagan test was significant with p-value < 2.56e-08. Application of IRRR as described in Section 3.4 resulted in a p-value of 2.13e-16. Instead of regressing the squared residuals on $\mathbf{x}_t$, we regressed the square root of the absolute residuals on $\mathbf{x}_t$, as described in Section 3.4. We obtained $p = 0.33$ on the Breusch-Pagan test. From the apparent benefit of this transformation of the residuals and from Figure 5.3, we suspected that the heteroskedasticity was caused by some severe outliers. Our suspicion was confirmed as we could remove heteroskedasticity by removing the 100 observations with the largest residuals, yielding $p = 0.09$ on the Breusch-Pagan test. The Regret curves are found in Figure 5.6.

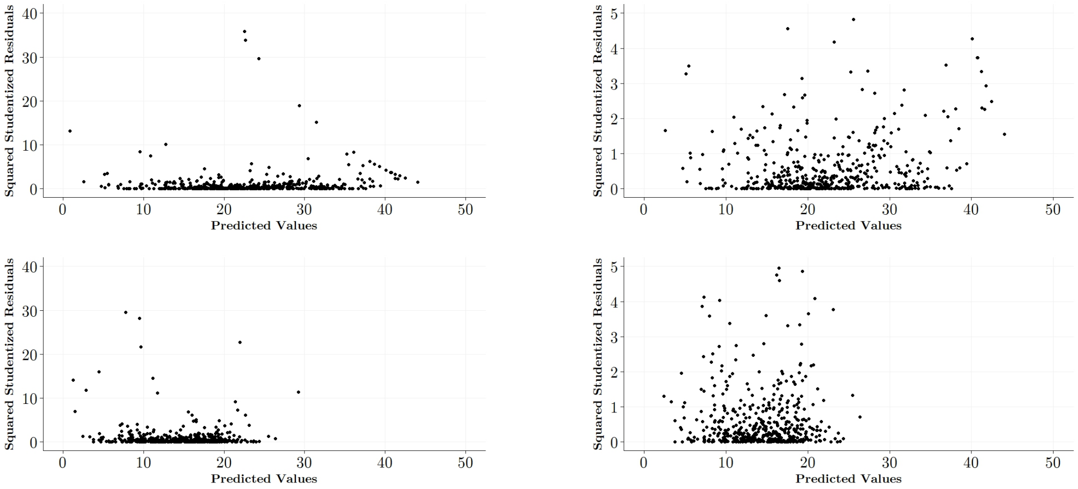

Figure 5.3: Boston housing. SSR against predicted values (Least Squares) for two different scales (left and right). Top: before weighting. Bottom: after weighting.

5.3 Summary and interpretation

The relative Regret differences of OWLS and OWRR against OnLS and ORR are summarized in Figure 5.1 and in Figure 5.2.

Dataset	Variance function offline	Variance function online	Variance function online (with skipped rounds)
Countries ($d = 2$)	25.26%	7.93%	8.01%
Belgian firms ($d = 4$)	13.86%	13.01%	5.51%
Boston housing ($d = 13$)	0.19%	0.15%	0.2%

Table 5.1: Relative performance of OWLS aginst OnLS

From Table 5.1 and Table 5.2 we read that, in order to learn a variance function online without skipping any rounds, the gain for weighting with an offline learned variance function must be relatively high. When this gain is small, as is the case for the Boston housing data, it becomes risky to learn the variance function online and we can suffer a relative loss.

Dataset	Variance function offline	Variance function online	Variance function online (with skipped rounds)
Countries ($d = 2$)	25.86%	8.12%	8.29%
Belgian firms ($d = 4$)	22.54%	15.81%	7.24%
Boston housing ($d = 13$)	6.15%	-21.51%	0.53%

Table 5.2: Relative performance of OWRR aginst ORR

The relative gain for weighting on the Boston housing data was small, even when the variance function was learned offline. As was explained in Section 5.2, the heteroskedasticity seemed to arise from around 100 outliers, which potentially means that weighting is not beneficial for a large portion of the dataset. When learning the variance function online, having outliers in the first few rounds could construct a variance function that does not work well for more moderate observations. A solution is skipping a few rounds to stabilize the variance function, as the variance function then would not be based on too little data. We indeed saw an improvement for the Boston housing data when we started applying the variance function after 15 rounds.

On the Boston housing data a large difference was found between the unpenalized and the penalized algorithms. This is due to the high dimensionality of $\mathbf{x}_t$ ($d = 14$). The larger the dimensionality of $\mathbf{x}_t$, the larger the benefit of ORR or OWRR, as the corresponding regression weights of these algorithms are always defined. The regression weights for OnLS and OWLS are not defined for $t < d$.

On the other hand, the Belgian firm data were found to be heteroskedastic on multiple scales. The heteroskedasticity did not seem to be (only) caused by outliers. The loss of an online learned variance function, relative to an offline learned variance function, was small. Skipping 15 rounds made the performance of an online learned variance function worse; the unweighted estimators were used for too long, leading to a deteriorating performance compared to a variance function applied from the first round. A linear variance function seemed to do quite well here and did not need to be stabilized. This was indicated by Figure 5.2, where we saw that the SSR increased with the predicted values in a fairly linear way.

The SSR for the countries dataset did not seem to increase linearly with the predicted values (Figure 5.1). Nevertheless, the linear variance function did still perform reasonably well and the weighted algorithms had a relative gain over the unweighted algorithms.

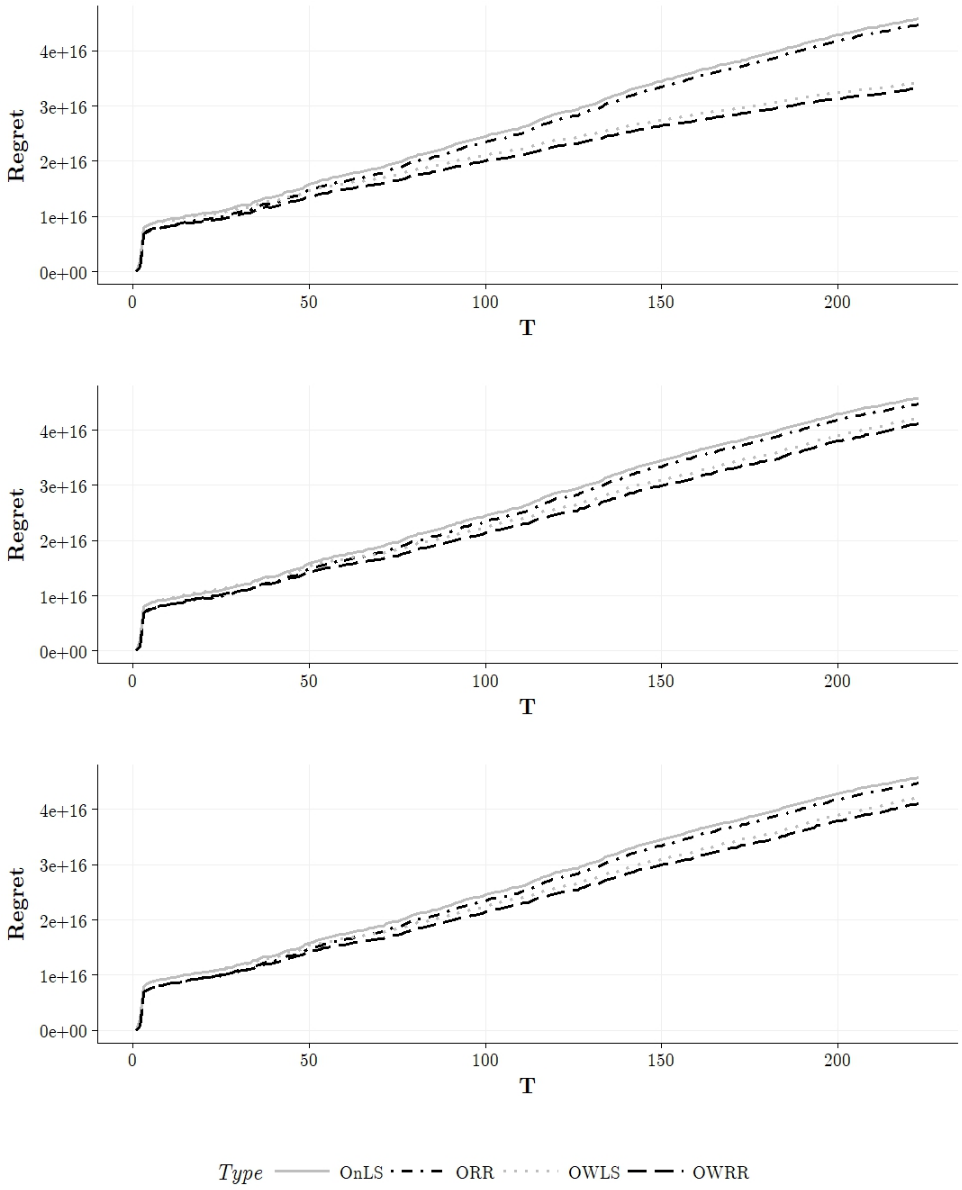

Figure 5.4: Regret curves for the country population data. From top to bottom with OWLS/OnLS and OWRR/ORR improvements between brackets. Variance function learned offline (25.26%, 25.86%), variance function learned online (7.93%, 8.12%) and variance learned online with first 15 rounds skipped (8.01%, 8.29%).

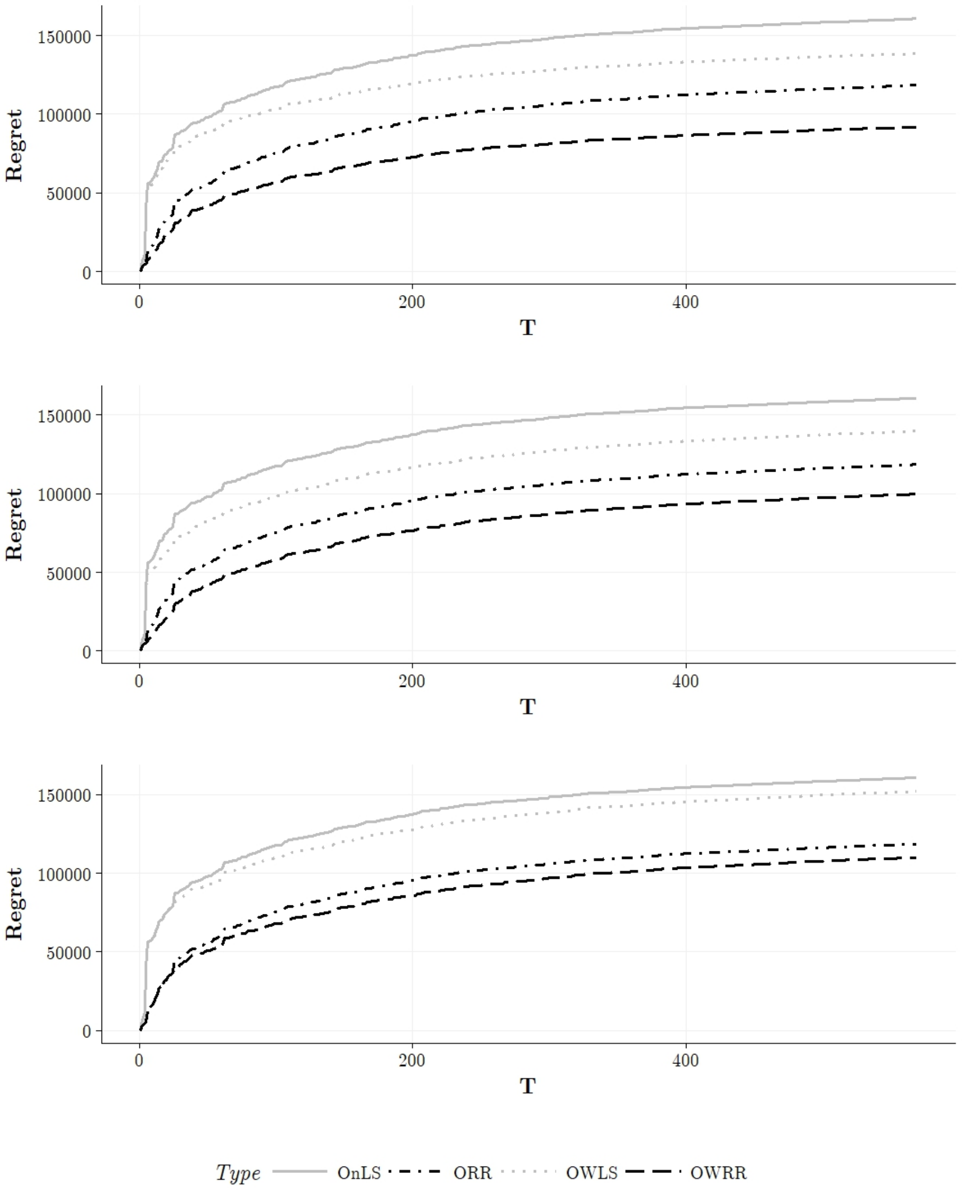

Figure 5.5: Regret curves for the country population data.From top to bottom with OWLS/OnLS and OWRR/ORR improvements between brackets. Variance function learned offline (13.86%, 22.54%), variance function learned online (13.01%, 15.81%) and variance learned online with first 15 rounds skipped (5.51%, 7.24%).

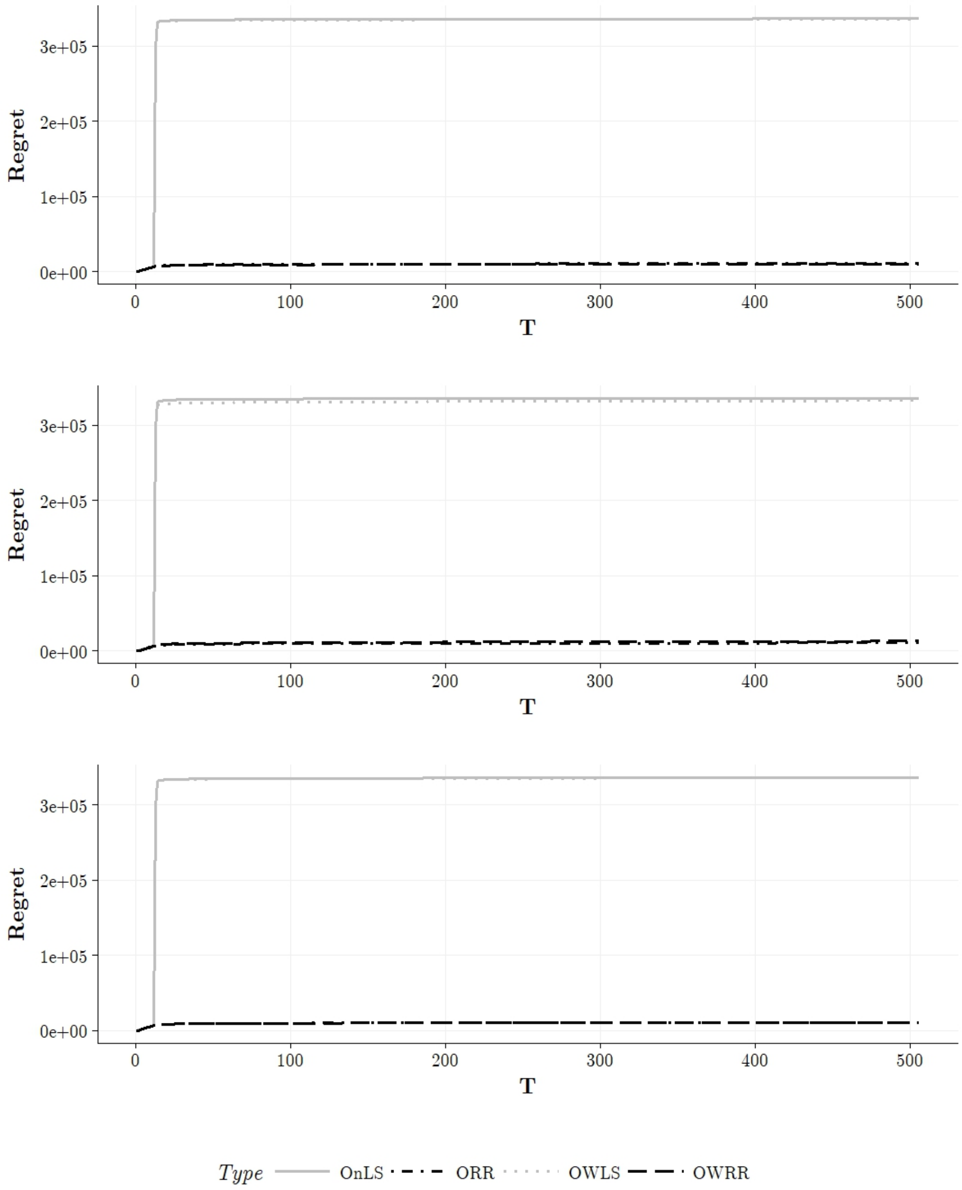

Figure 5.6: Regret curves for the Boston housing data. From top to bottom with OWLS/OnLS and OWRR/ORR improvements between brackets. Variance function learned offline (0.19%, 6.15%), variance function learned online (0.15%, −21.51%) and variance learned online with first 15 rounds skipped (0.2%, 0.53%).

Chapter 6

Concluding thoughts

As a consequence of the Gauss-Markov theorem (Clapham and Nicholson, 2009), we know that OWLS is BLUE under heteroskedasticity. Secondly, we know that when employing a single feature OWRR has a lower loss in expectation than OWR. Thirdly, we know that this is no longer true when the number of features exceeds one.

All counter-examples of Chapter 4 were found in the area where $\sigma^2 < 1$. Thus for low σ^2 , one should be careful with weighting in ORR. On real-world datasets, with an offline learned variance function, we were able to outperform ORR on all three datasets. With an online learned variance function, we performed better than unweighted ORR on two out of three datasets. The exception was the Boston housing data. On this dataset the offline learned linear variance function did not perform well in the first place. When the relative gain for weighting with an offline learned variance function is not large enough, it is possible that learning the variance function online results in a relative loss to unweighted OLR. The results can then be improved by skipping rounds and giving the algorithm some time to stabilize the variance function.

6.1 Open questions and future work

We do not know how to optimally choose λ, but instead we chose to set $\lambda = 1/\|\mathbf{u}\|_2^2$. The simulations showed that OWRR is not always better than ORR; an eventual loss for OWRR was found when $\sigma_t^2 < 1$. A partial solution could be defining $\lambda^{WR} = \sigma^2/\|\mathbf{u}^*\|_2^2$ to penalize smaller variances softer than larger variances. As we already defined σ^2 for OWRR, letting the variance enter the definition of λ^{WR} would not cause any problems. It is an open question whether the general λ should be redefined to contain the variance. The variance is not defined for Online Ridge Regression, so defining such a λ for ORR would mean a further entanglement between OLR and Statistics.

A relative gain was found for OWRR on two out of three datasets. A relative loss for OWRR was found on the Boston Housing data. A potential cause is the heteroskedasticity being caused by outliers. A solution for heteroskedasticity by outliers is a different loss function, that is less susceptible to outliers. An example of such a loss function is the absolute loss. Application of a different variance function could prove beneficial too. Throughout this thesis, we have assumed a linear variance function, but non-linear alternatives could be considered. Especially in the case of heteroskedasticity by outliers, such a variance function could be constructive. When the heteroskedasticity is caused by outliers, it is not likely that the variance function is just some linear function of $\mathbf{x}_t$.

The largest drawback of learning the variance function online, as described in this thesis, is that it learns the variance function for every round t on all the previous $t-1$ rounds. This is not efficient and needless to say, very slow. For further application of weighting in the OCO framework, an algorithm is needed that completely learns the variance function online, without having to review all the information of the previous $t-1$ rounds.

Bibliography

O. Anava and S. Mannor. Heteroscedastic sequences: Beyond gaussianity. *Proceedings of The 33rd International Conference on Machine Learning*, pages 755–763, 2016.

R. G. Askin and D. Montgomery. Augmented robust estimators. *Technometrics*, 22:333–341, 1980.

R. Carroll and D. Ruppert. *Transformation and Weighting in Regression.* Chapman and Hall, New York, second edition, 1988.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, Cambridge, United Kingdom, first edition, 2006.

CIA. The world factbook. `https://www.cia.gov/library/publications/the-world-factbook`.

C. Clapham and J. Nicholson. *The Concise Oxford Dictionary of Mathematics.* 01 2009. ISBN 9780199235940.

J. Fox. *Applied Regression Analysis and Generalized Linear Models.* SAGE Publications, 2008. ISBN 9780761930426.

G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Pres, Baltimore, United States, third edition, 1996.

E. Hazan. Introduction to online convex optimization. *Foundation and Trends in Optimization*, 2(3-4):157–325, 2015.

Kaggle.com. Boston housing. `https://www.kaggle.com/c/boston-housing`.

L. Kuptsov. *Encyclopedia of Mathematics.* Springer Science+Business Media B.V. / Kluwer Academic Publishers, 2001.

S. Salev-Shwartz. *Online Learning:Theory, Algorithms, and Applications.* PhD thesis, Hebrew University, 2007.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundation and Trends in Machine Learning*, 4(2):107–194, 2012.

W. van Wieringen. Lecture notes on ridge regression. 2015.

M. Verbeek. *A Guide to Moden Econometrics.* Wiley, Chichester, United Kingdom, fourth edition, 2012.