Competitive Regression

by

Waqas Jamil

the thesis is submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in the

Computing & Informatics department

Bournemouth University

Abstract

This thesis is about investigating the predictive complexity of online regression. In essence, supervised learning from a data sequence consisting of *n*-dimensional input and the corresponding output is considered. In this work online learning scenario considered consists of sequential arrival of data, without making any stochastic assumptions on the nature of the arriving data. At each trial, the learner receives the input, it produces a prediction. Then as a second step, the true output is observed, and the learner suffers a loss which is consequently used to learn. The goal of online learning regression in this thesis, is to minimise the regret suffered when considering the loss of the minimum of the sum of squares.

In the present work, three novel algorithms (Online Shrinkage via Limit of Gibbs sampler (OSLOG), Competitive Iterated Ridge Regression (CIRR) and Competitive Normalised Least Squares (CNLS)) are derived and analysed. The development of these algorithms is driven by Kolmogorov complexity (known also as "competitive analysis"). OSLOG appraises the Bayesian approach, CIRR relies on game theory, whereas CNLS makes use of gradient descent methods. The analysis of the algorithms investigates two aspects: (1) formulating the upper bound on the cumulative square loss and (2) identifying the precise conditions under which they perform better than other algorithms. In fact, the theoretical results indicate that they have a better guarantee than the state-of-the-art algorithms. The empirical study conducted on real-world datasets show also that the performance of the proposed algorithms is better than the state-of-the-art algorithms and close to the minimum of the sum of squares.

Acknowledgements

This work concludes my rather long process of formal education. My family,especially my mother and father, have greatly contributed to the development of my motivation to study. Therefore, my utmost gratitude and appreciation are reserved for my family, their constant love, support and encouragement was the main ingredient in producing this thesis. I dedicate this thesis to them.

I'm very grateful to Professor Abdelhamid Bouchachia for giving me the opportunity to pursue the doctorate degree. His unprecedented kindness and patience allowed me to develop myself as a person and as a researcher. His constant feedback on my work allowed me to greatly improve my writing and presentation. Above all, he taught me the ways of science.

Moreover, I thank all my teachers in particular Professor Martin Ridout, whose suggestion of considering further post-graduate studies proved to be invaluable. My regular meetings with Dr Owen Lyne helped a lot in building the foundations of probability. I'm in debt to Professor Vladimir Vovk and Dr Yuri Kalnishkan for introducing me to game-theoretic probability and learning theory – I feel, I would have not found anything more suitable to study for myself.

My colleagues at Bournemouth University also played a vital role in making my journey pleasant, I thank them for their help and support.

Last, but not the least I thank all the reviewers of the journals and conferences who gave me invaluable feedback.

Financial support was given by Bournemouth University and the European Commission under the Horizon 2020 Grant 687691 related to the project PROTEUS: Scalable Online Machine Learning for Predictive Analytics and Real-Time Interactive Visualisation.

Acronyms

AA aggregating algorithm

AAR aggregating algorithm for regression

ARMA auto regressive moving average

CIRR competitive iterated ridge regression

CNLS competitive normalised least squares

EGD exponentiated gradient descent

CD coordinate descent

GD gradient descent

HA halving algorithm

LASSO least absolute shrinkage selection operator

LMS least mean squares

LS least squares

NLS normalised least squares

NLMS normalised least mean squares

NGD normalised gradient descent

OSLOG online shrinkage via limit of Gibbs

SLOG shrinkage via limit of Gibbs

OGD online gradient descent

ORR online ridge regression

ONS online newton step

RLS recursive least squares

SLOG shrinkage via limit of Gibbs

RR ridge regression

WMA weighted majority algorithm

LASER last step adaptive regression algorithm

AROWR adaptive regularisation of weights regression

APA aggregating pseudo algorithm

WMA weighted majority algorithm

FLA follow the leader algorithm

HA halving algorithm

Symbols

$\mathbb{Z}^+$	positive integers
$\mathbb{N}$	natural numbers
$\mathbb{R}$	real numbers
$\mathbb{P}$	probability measure
Ω	outcome space
Θ	decision or parameter space
$\mathcal{F}$	filtration
$\mathcal{K}(.)$	complexity
$\mathcal{H}$	Hessian
$\mathcal{O}$	order of complexity
f	function
t	trial
T	data length
$\nabla f(.)$	differential of $f(.)$
$\|.\|_p$	$p-$norm
x'	transpose of vector x
inf	infimum
sup	supremum
min	minimum
max	maximum
argmin	minimum point
argmax	maximum point
i.i.d	independent and identically distributed

List of Figures

Figure 1 ℓ_1-norm approximation. 52
Figure 2 Tunable loss function (see Theorem 16) 80

List of Tables

Table 1 Cook distance, mean & variance 85
Table 2 Algorithms accuracy comparison on real-world datum 88

List of Protocols

1 Experts based prediction system 24

2 A two player repeated game 40
3 A prediction game with experts advice 40
4 Bayesian strategy 41

5 RLS 42

6 OSLOG strategy 51
7 OSLOG 55

8 CIRR strategy 66
9 CIRR 68

10 CNLS strategy 75
11 CNLS 76

Contents

I INTRODUCTION 21

1. Online learning 23
2. Online regression 26
3. Research questions 33
4. Publications 34
5. Organisation 35

II COMPETITIVE REGRESSION 37

1. Background 39
2. Related work 41
3. Problem formulation 45

III OSLOG: ONLINE SHRINKAGE VIA LIMIT OF GIBBS 49

1. Derivation 51
2. Analysis 56

IV CIRR: COMPETITIVE ITERATIVE RIDGE REGRESSION 63

1. Derivation 65
2. Analysis 69

V CNLS: COMPETITIVE NORMALISED LEAST SQUARES 73

1. Derivation 75
2. Analysis 77

VI EMPIRICAL STUDY 83

1. data description 85
2. Experimental setting 86
3. Results 86

VII CONCLUSION AND FUTURE WORK 89

1. Conclusion 91
2. Future work 92

References 95

I

Introduction

The most we can know is in terms of probabilities.

—Richard Feynman

1- Online learning

Most learning algorithms fall under the umbrella of supervised or unsupervised learning. In this work, the discussion is confined to supervised learning problem. This work considers sequentially arriving data $\mathcal{S} = \{(x_1, y_1), ..., (x_t, y_t)\}$ with output $y_t \in \mathbb{R}$, input $x_t \in \mathbb{R}^n$ and a fixed pre-defined loss function $f_{\mathcal{S}}$. The learning is defined in the following sense [73]:

Definition 1. *A computer program is said to "learn" from experience, with respect to some task and a performance measure, if its performance improves with experience with respect to the performance measure.*

Broadly speaking, online and batch mode of leaning are the two most popular modes of learning. Online mode of learning considers sequential arrival. At each trial $t = 1, 2, ...$, an input is observed and the prediction is made by updating the parameters based on some pre-defined performance measure. Therefore, the online learning fits nicely with the Definition 1. In contrast, in batch learning, the data is divided into training and testing sets, and the performance is optimised on the training set, usually by making several passes [43].

Recently, the framework of online learning has been formalised using Game theory by Shafer and Vovk [90]. However, the core idea of online learning can be tracked back to the communication between Pascal and Fermat [78].

Kolmogorov defines the complexity of a string t, with respect to an algorithm A that transforms a binary sequences into words by considering a set of all words over some finite alphabet [60]:

Definition 2. *The complexity ($\mathcal{K}$) of the string t with respect to the algorithm A is defined as the length (l) of the shortest program which computes it, i.e.*

$$\mathcal{K}_A(t) = \min_{A(p)=t} l(p) \tag{1}$$

If $A(p) \neq t$ for all binary strings p, then $\mathcal{K}_A(t) = +\infty$. By using the result given in [98], one can consider a base line algorithm B and say:

$$\mathcal{K}_A(t) \leq \mathcal{K}_B(t) + c \tag{2}$$

where c is a constant that depends on A and B, but not on t.

Please note that the Definition 2 does not require any stochastic assumptions. This will help to define the scope of this work.

Throughout the thesis, the output is considered to be generated from some unknown mechanism, which could be deterministic, stochastic, or even adversarial. It is assumed that a forecaster observes the sequence of observations $x_t \in \mathbb{R}^n$ for $t \in \mathbb{Z}^+$ and predicts $y_t \in \mathbb{R}$, where the vector space $\mathbb{R}^n$ is yet to be remarked upon. Instead of predicting a distribution, a single point is predicted sequentially. Also, since at each trial only a single point of data is considered by the model, thus, this approach is memory efficient. Formally, one can say the learning algorithms receives an input at each trial; makes predictions; receives the ground truth and then update the parameters, more on this later. This theory consider, a class of *reference forecasters*. The predictions of the reference forecasters are available before the actual outcome is revealed. This allows the forecaster to predict by considering the reference forecasters predictions. To measure the performance of the forecasters prediction, a loss function is used, which measures the discrepancy between the forecasters predicted value and the actual outcome. Often the reference forecasters are referred as experts or decision strategies (used interchangeably). Perhaps, an example will help in providing the intuition of the protocol for the online learning used in this work.

Example 1. *Say, I am interested in forecasting the probability that it will rain on a given day. I ask the experts to provide their respective probabilities of forecast. Then I pass the experts forecasts to the forecaster to predicts the probability of the rain. Notice, that my reference forecaster is an expert on weather forecasting. After each round the expert update their prediction on the basis of the discrepancy between actual outcome and the forecasters prediction. For, further details please see for example [81, 101, 72]*

Example 1 can be generalised using some notation. Suppose, elements of a sequence $\omega_1, \omega_2, ...,$ come from an *outcome space* Ω. Predictions are within the *prediction space* Γ. To measure the quality of our prediction a *loss function* $\lambda : \Gamma \times \Omega \to [0, +\infty]$ is used. So, the prediction game is a triplet consisting of $(\Omega, \Gamma, \lambda)$. Suppose there are N experts $\theta_1, \theta_2, ..., \theta_N \in \Theta$ that predict the outcome, then the predictions made by experts at time t are γ_t^n, where $n = 1, ..., N$, for further details please see [110, 111]. Protocol 1 is used to contextualise the generalisation of Example 1:

Protocol 1: Experts based prediction system

```
$L_0 = 0$
$L_0^\theta = 0, \theta \in \Theta$
FOR t=1,2,...
    (1) Experts $\theta_{1,2,...,N} \in \Theta$ predicts $\gamma_t^{1,2,...,N} \in \Gamma$
    (2) Learner output $\gamma_t \in \Gamma$
    (3) Actual output $\omega_t \in \Omega$
    (4) $L_t = L_{t-1} + \lambda(\omega_t, \gamma_t)$
    (5) $L_t^\theta = L_{t-1}^\theta + \lambda(\omega_t, \gamma_t^n)$
END FOR
```

Here onward, the key concept for the main topic will be discussed formally. Next, defining the notion of algorithm's "competitiveness" − the mathematical comparison apparatus of the algorithms complexities. Competitive prediction can be viewed as a sub-field of game-theoretic probability that uses the technique of competitive analysis. Competitive analysis were invented especially to analyse online learning algorithms by Sleator and Tarjan [97]. The performance of the online algorithm is compared to the best learning *strategy* in the hindsight. Often, the learning strategy is the optimal offline algorithm that can view all the sequence of observations and outcomes. From here on, the algorithms that will adhere to the following definition will be classified as competitive algorithms.

Definition 3. *An algorithm is said to be competitive if the difference between the performance of the online and the optimal offline algorithm is bounded* [1]*. More precisely:*

$$L_T \leq L_T^* + R_T \tag{3}$$

where L_T is the cumulative loss of the learning algorithm at trial T, L_T^* is the cumulative loss of the optimal learning strategy and R_T is the regret of the learning algorithm. The measure of competitiveness is the regret term i.e. it tells how well the learning algorithm learns.

The traditional worst-case analysis are only done for the "hard" inputs, whereas if an algorithm gives competitive prediction, then it means it is competitive for the "hard" and the "easy" inputs, where hard and easy is defined by the optimal offline learning algorithm. Also, the above definition of competitiveness is a stronger notion than the Probably Approximately Correct (guarantees hold on expectation) and statistical convergence (algorithm converges to the true solution in probability).

Competitive analysis can be thought of worst case analysis for online and stochastic algorithms where input can be easy or hard. The analysis are done by assuming that an *adversary* deliberately chooses a strategy that maximises the difference between the algorithm prediction and the actual outcome. An adversary could be oblivious (unaware of the learner's moves) or adaptive (adopts according to the learner's moves) for a stochastic algorithm. In online learning, an adversary is considered to be oblivious as adaptive adversary can always outsmart the learning algorithm for sequentially arriving out-

[1] The competitive prediction may be thought of the predictive complexity [53] that generalises Kolmogorov complexity [115, 116].

comes. Thus, the distinction of the adversary make little sense for online learning algorithms.

Remark 1. *Definition 3 directly relates to Definition 2. However, the fundamental difference is that the regret term may be some function of t. Also, throughout this thesis, algorithm is a concept that is computable in one of the various equivalent ways that have been proposed, e.g. by means of the theory of partial recursive functions [38].*

2- Online regression

Unlike offline learning, online learning observes data as a sequence of data instances (data stream) where learning happens over consecutive rounds without seeing data more than once. Each round consists of *iii*) main steps: i) Predict the output: the learner receives an instance, $x_t \in \mathbb{R}^n$ and predicts an output, $\hat{y}_t \in \mathbb{R}$. ii) Reveal the ground truth: the learner obtains the correct output, $y_t \in \mathbb{R}$. iii) Adjust the model: the learner suffers a loss $L_t(y_t; \hat{y}_t) \in \mathbb{R}$ and learns by adjusting its model. Clearly, online learning has been designed for supervised learning [43] and reinforcement learning [7] and can be seen as a game between the learner and the nature. Such modelling has given rise to a rich body of theoretical work around online learning.More on this will follow in subsequent sections and chapters. The previous protocol refers to learning in each round from one single point, extensive work is done to deal with window (batch) of data at a time. In batch-based sequential processing, the data is split into overlapping or independent windows, known in general as sliding windows. Therefore, input is generalised to $\mathbf{X}_t \in \mathbb{R}^{m \times n}$, output and prediction is $\mathbf{Y}_t, \hat{\mathbf{Y}}_t \in \mathbb{R}^n$, and the loss suffered is $L_t(\mathbf{Y}_t; \hat{\mathbf{Y}}_t) \in \mathbb{R}^n$.

A sliding window can be of fixed or variable length. In sliding window, a window is formed over some part of data, and this window can slide over the fixed or a variable length of data to capture different portions of it. For fixed length window please see [88, 102, 49, 66, 12] and for variable length please refer to [71, 58, 32, 122, 64]. Sliding windows, in which time windows do not intersect are known as non-overlapping windows, for details please see for example [95].

Online learning community has also been motivated by addressing computationally efficient algorithms to capture dynamic changes in the data streams. Often, the dynamic change in the data is referred to as *concept drift*. The word dynamic change is very broad. Often the concept of drift is defined in a probabilistic sense. Kelly et al. [55], Gama et al. [33] and Webb et al. [117] define the concept drift (CD) using prior probabilities, class conditional probabilities and posterior probabilities, which is equivalent of considering joint probabilities:

$$CD = \mathbb{P}(\mathbf{X}|\mathbf{Y})\mathbb{P}(\mathbf{Y}) = \mathbb{P}(\mathbf{X}, \mathbf{Y}) \tag{4}$$

where **X** and **Y** denote the input and output respectively. Some work has been done to quantify the level of drift in absolute terms, see for example [117], where magnitude and the duration of the drift is quantified by using Hellinger Distance [48]. Webb et al. [117] justifies the choice of Hellinger Distance by opposing the popular Kullback-Leibler Divergence [63] due the difficulty of interpretation and generalisation to many scenarios.

That has been said, the definition of drift might be extended to the regression case. Another approach to handle concept drift could be to use ensemble methods. One of the methods could be to have models to cover certain number of scenarios and use weighted average of them as final prediction, please see [51, 22]. It is also possible to use Kriging (Gaussian process regression) to address drift, as an example please see [59, 100]. Typically modelling regression tree [15] based approaches can be used to handle drift in the data. The fundamental advantage of using tree-based regression is that there is no assumption of linearity in input and output, but if there is a linear relation in input and output, linear models are likely to outperform tree-based approaches. Furthermore, as the dimensionality of the data increases the advantage disappears, for details please see [43]. In [50] it is shown how regression trees can handle gradual and abrupt drifts. Reflecting on this state-of-the-art of drift handling, our contribution can be summarised as follows. The algorithm competitive normalised least squares (CNLS) mentioned in Chapter V is not based on co-variance updates, thus it does not converge and is able to handle the abrupt, incremental and gradual drifts mentioned in [33] Figure 2. Algorithms online shrinkage via limit of Gibbs (OSLOG) and competitive iterated ridge regression (CIRR) mentioned in the Chapters III and IV, maybe can be extended to handle drift as done in [75]. The algorithm CIRR mentioned in the Chapters IV handles outliers (the last case mentioned in Fig 2 of [33]). However, algorithms discussed in this work consider no generative mechanism on either input or the output, they are allowed to be any numbers range following any pattern. Furthermore, the mean and variance may change over time, providing capabilities to deal with non-stationary data and excluding any assumption of independence and identical distribution or other stochastic assumptions. Our main restriction is that there is one fixed optimal function against which the algorithm competes over time. The fundamental difference in sequential algorithms processing batches and processing single data point is that, one predicts a distribution while the later predicts a real number. Thus, the later can avoid all the distributional assumptions.

In game-theoretic terms, single data processing with no delayed feedback of output or the outcome are referred as perfect information games (for details please see for example [80]). The link to game-theory has only recently been discovered and talked in lengths by Shafer and Vovk [91]. The rise of the approach of the game-theoretic approach can be tracked back to the paradigm of *prediction with expert advice* introduced in the late 1980′s by DeSantis et al. [24], who presented predicting sequentially using experts advice. Then further work was done by Littlestone [67], Littlestone et al. [69], Littlestone and Warmuth [68], Foster [29], Foster and Vohra [30], Freund [31], Cesa-Bianchi et al. [19, 18], Haussler et al. [45], Vovk [108], Yamanishi [120].

Perhaps, a good starting point to explain the online learning framework used in this thesis is by explaining halving algorithm (HA) in light of the prediction with experts advice paradigm (please see Protocol 2). In HA it is assumed that there exists an expert among N experts that predicts correctly, i.e., the loss of the expert at every trial is null. Initially, giving equal importance to all N experts, but if an expert makes a mistake, it is discarded.

Theorem 1. ***(Theorem 1 [67])*** *The upper bound on HA is as follows:*

$$L_T \leq \lfloor \log_2 N \rfloor \tag{5}$$

Theorem 1 applies no matter what the outcome is on any trial or no matter what the experts predict, provided there exist at least one expert that is always correct. If the nature and the experts unite, the learner will make $\lfloor \log_2 N \rfloor$ mistakes, provided there exists an expert that makes no mistake. Notice, unification of nature and the experts, is the worse that could happen for the learner. The lower bound on halving algorithm, provided there exist at least one expert who makes no mistake is equivalent to the upper bound.

$$L_T \geq \lfloor (\log_2 N) \rfloor \tag{6}$$

Another way of viewing HA is by assigning weights to the experts. Let $s_T = \sum_{n=1}^{N} w_T^n$, where w denotes the experts weights. Setting $s_0 = N$ and weights can stay the same or decrease at each trial. If the learner makes a mistake on the T-th trial then there are at least half of the experts with non-zero weights. The weights can never increase and the weight of the expert who makes no mistake is one. Experts weights are updated according to the following rule:

$$w_T^n = w_{T-1}^n (1 - \lambda(\gamma_T^n, \omega_T)) \tag{7}$$

HA is only applicable when there exists a perfect expert. By resting this condition, the guarantee does not hold. The weighted majority algorithm (WMA) algorithm was introduced by Littlestone and Warmuth [68] with a strong performance guarantee. In WMA, there is no restriction of having a perfect expert in the pool of experts. The difference between HA and WMA is in the weights update rule. When an expert makes a mistake, its weight is multiplied by coefficient $\beta = e^{-\eta} < 1$, where $\eta > 0$. So, instead of (7), the update rule is reformulated as follows:

$$w_T^n = w_{T-1}^n \beta^{\lambda(\gamma_T^n, \omega_T)} = w_{T-1}^n e^{-\eta\lambda(\gamma_T^n, \omega_T)} \tag{8}$$

Notice that η and β have an inverse relation. When β is small weights are small and vice versa. The mistake bound for WMA is as follows:

Theorem 2. *(**Theorem 2.1 [68]**) For $\beta \in [0, 1]$ and N experts, then for every outcome arriving sequentially and for any expert the following holds:*

$$L_T \leq \frac{\ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}} L_T^\theta + \frac{\ln N}{\ln \frac{2}{1+\beta}} \tag{9}$$

One can optimise β depending on specific scenarios. For example, if one expects at least one expert to be very good then β is close to zero.

Theorem 3. *(**Main result [110]**) For a sequential perfect information prediction game with only two experts and $\Omega = \{0, 1\}$, no strategy can guarantee the following for all outcomes:*

$$L_T \leq c L_T^\theta + a \tag{10}$$

where $c < 2$ and $a > 0$.

Theorem 2 also directly follows from the main result proven by Vovk [110]. Each expert can not make more than $\frac{T}{2}$ mistakes. Whatever the situation the bound (10) holds because the following holds:

$$T \leq c\frac{T}{2} + a \Rightarrow c \geq 2 - \frac{a}{T} \tag{11}$$

if $c < 2$, then, as $T \to \infty$, than $\frac{a}{T} \to 0$ and (11) is violated. The proof of the case when $c = 2$ leads to generalisation of WMA. Vovk [110] shows that there does not exist any learning strategy that can improve the upper loss bound of the WMA, by showing that for $c = 2$ the bound mentioned in (11) is violated. A comparable approach to WMA is an algorithm known as follow the leader algorithm (FLA) [93]. The algorithm predicts according to the following rule:

$$\gamma_T = argmin \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t^n), \;\; n = 1, ..., N \tag{12}$$

In (12) the prediction γ_T is essentially based on the cumulative loss of all expert(s), and the optimisation problem is solved usually using online gradient descent (OGD). The expert that has the least loss up until time T is selected to predict $T+1$. The fundamental disadvantage of this approach is that it is impossible to have a good upper loss bound because the expert that has performed well up until time T may not perform well at the next step. There is no guarantee that the expert that performed well until time T will perform well later as well.

In this thesis, the design and analysis of the algorithms adhere a similar framework as of the HA and WMA, but the problem considered is more general. Specifically, the input and outcome space could be a real number. The supervised learning algorithms proposed in this thesis are primarily dedicated to regression [43]. They are based on sequential perfect-information games, Bayesian learning and convex optimisation.

In the literature of signal processing the problem of predicting on the fly is handled using various filters [41, 25]. The goal of the filter is to recover the noisy observations, whereas the purpose of the game-theory based learning algorithms is to predict the response variable. Despite, having different objectives, signal processing filters can be adapted to solve the online regression problem, as more clearly defined in this thesis later. Half a century ago Widrow and Walach [118] developed an algorithm for reducing noise via adaptive filtering, the algorithm is known as least mean squares (LMS). Bershad [11], Bitmead and Anderson [14] performed analysis on LMS showing that normalised least mean squares (NLMS) is insensitive to scaling of the input. Hayes [46] proposed recursive least squares (RLS) for online regression. RLS uses a correction factor to update covariance matrix at each iteration. RLS was mainly inspired from a priori filtering and a posterior filtering. A priori filtering is used to recover un-corrupted output $w_t'x_t$, before receiving the output y_t. The overall discrepancy (error) after t steps is given in the form of a cumulative sum: $\sum_{s=1}^t (w_s'x_s - w_{s-1}'x_s)^2$. In a posterior filtering, filtering out the noise using the output y_t is done. The error is formulated as a cumulative sum: $\sum_{s=1}^t (w_s, x_s - w_s'x_s)^2$. Notice in a posterior filtering, one makes use of the most recent weight vector w_t to measure the quality of the filter (error). In contrast, for a priori filtering, w_{t-1} is used due to unavailability of the output y_t, which resembles the online learning setting. However, in filtering the goal is not to estimate the output, instead to recover the output by assuming that it is corrupted by some noise.

A popular area of deep learning can use some of the discussed regression algorithms as their building blocks. For example, the learning in Neural Network is done using gradient descent (GD), in this thesis an alternate approach to is also discussed. One may also be able to use the other discussed algorithms to perform learning tasks on specific type of data. For example, please see [82].

On a more practical note, some work has been done to handle data streams using computationally efficient methods. The two well known libraries that handle data streams are Massive Online Learning (MOA) and Spark [3, 40]. In Chapter 2 section 2.2 of [13], same protocol as the one in this work is considered. In this work, a solid mathematical foundation to regression algorithms is considered, which differs from the work done in MOA and Spark for online regression algorithms. Spark designed for batch processing of data [99]. In contrast, MOA defines online processing of batches of data. Almost the algorithms mentioned in Chapter 9 in [13], make use of stochastic assumptions, in my understanding they are evaluated by considering sliding window. Thus, processing batches of data sequentially. Arguably one can adopt Spark and compare it to MOA as done in [3]. Comparing the regression algorithms derived and analysed in this thesis differ from the regression algorithms in Spark and MOA. Most of the discussed algorithms in this thesis can be found in SOLMA[1], standing for Scalable Online machine Learning and data Mining Algorithms. This thesis only focuses on regression algorithms mentioned in SOLMA library, in particular their theoretical motivation and analysis. That been said, MOA and Spark libraries have a vast variety of algorithms, so these libraries have the ability to capture wider range of scenarios and data sets. SOLMA is limited to least squares and its variant, but has much stronger theoretical case for these algorithms than MOA or Spark.

Let us view the classical least squares (LS) approach. Considering the following model, with $\mathbf{Y}, \epsilon \in \mathbb{R}^{t\times 1}$, $w \in \mathbb{R}^{n\times 1}$ and $\mathbf{X} \in \mathbb{R}^{t\times n}$:

$$\mathbf{Y} = \mathbf{X}w + \epsilon$$

Computing the weight vector w, that minimises the squared sum of errors:

$$\epsilon'\epsilon = \mathbf{Y}'\mathbf{Y} - 2w'\mathbf{X}'\mathbf{Y} + \hat{w}'\mathbf{X}'\mathbf{X}\hat{w} = 0$$

[1] SOLMA intends to cover two classes of algorithms: basic streaming routines such as moments, sampling, heavy hitters feature extraction, and advanced machine learning algorithms such as classification, clustering, regression, drift handling and anomaly detection.

From Gauss-Markov theorem [36] $\hat{w}$ is normally distributed with mean w and variance $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$, for further details please see for example [92]. Notice, w is:

$$\min_{w} \|\mathbf{Y} - \mathbf{X}w\|_2^2 \implies w = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \tag{13}$$

In (13) it is necessary for $\mathbf{X}'\mathbf{X}$ to be invertible. In certain scenarios the condition of invertibility might not be satisfied. So, a penalty or regularisation term is added, for instance by adding $a\|w\|_2^2$, where $a > 0$ in (13) the following estimate of beta is obtained:

$$\min_{w} \|\mathbf{Y} - \mathbf{X}w\|_2^2 + a\|w\|_2^2 \implies w = (\mathbf{X}'\mathbf{X} + a\mathbf{I})^{-1}\mathbf{X}'\mathbf{Y} \tag{14}$$

It is easy to see that $a\|w\|_2^2$ and $\|\mathbf{Y} - \mathbf{X}w\|_2^2$ commute and thus, it is possible to obtain a closed form solution. To obtain an online weights updating rule, one can write (14) as:

$$w = \left(\sum_{t=1}^{T} x_t y_t\right)' \left(\sum_{t=1}^{T} x_t x_t' + a\mathbf{I}\right)^{-1} \tag{15}$$

and consequently the forecasts at time $t = 1, 2, ...$ are $w'x_t$. One may chose to add ℓ_1 penalty instead, but then there is no obvious way to obtain a closed form solution. Often statistical literature refers to ℓ_1 regularised regression as least absolute shrinkage selection operator (LASSO) and ℓ_2 regularised regression as ridge regression (RR). Both ℓ_1 and ℓ_2 regularisation have their advantages and disadvantages. ℓ_1 and ℓ_2 regularisation are probably the two most popular regularisation methods for regression and are useful for many real world applications [93, 119]. ℓ_1 regularised regression aims to obtain a sparse solution. If the aim is to have a model that outputs few non-zero entries then ℓ_1 regularised regression is a good choice [44, 47, 39] . On the other hand, ℓ_2 regularised regression increases the bias and has lower variance than regression without regularisation and is a useful technique for dealing with the data that has high multicollinearity. For a more a detailed explanation please see [104].

The forecasts by using (15) leads to the time complexity of $\mathcal{O}(n^3)$ with the following bound [28, 113, 21]:

$$L_T \leq L_T^* + P^2X^2 + nP^2R^2 \ln(T+1) \tag{16}$$

where L_T and $\hat{L}_T$ denote the cumulative square loss of the online prediction algorithm, L_T^* denote the total squared loss of the offline solution. Also, the signals are taken from $\ell_\infty-$ball $\{x \in \mathbb{R}^n : \|x\|_\infty \leq R\}$ and $w \in \Theta$ from ℓ_1-ball $\{w \in \mathbb{R}^n : \|w\|_1 \leq P\}$

It is possible to make use of Sharman-Morrison [94] to achieve time complexity $\mathcal{O}(n^2)$ without sacrificing (16). One may make use a gradient based approach to reduce time complexity to $\mathcal{O}(n)$, but this reduction leads to the following bond [20]:

$$L_T \leq 2.25 \inf_w (L_T + (XP)^2) \tag{17}$$

Cesa-Bianchi et al. [20] showed that it is possible to relax the condition of bounded signals and weights for (17) by using normalised square loss and obtained the following bond:

$$\hat{L}_T \leq 2.25 \inf_w (\hat{L}_T + \|w\|_2^2) \tag{18}$$

where $\hat{L}_T$ denote the cumulative normalised cumulative square loss.

3- Research questions

The main objective of this work is to address the following questions:

1. Is it possible to have online algorithms that improves the bound (16)?
   → To answer the question two online algorithms CIRR and OSLOG are proposed using game theory and Bayesian theory respectively. Also, a detailed analysis of their upper bounds on the cumulative square loss are presented. For details please see Chapter III and IV.

2. If the answer to the first question is yes, then by how much and under what conditions? Else, why not?
   → The work presents the precise circumstances under which CIRR and OSLOG upper bounds are better than the state-of-the-art.

3. Is it possible to have a tuning parameter next to the regularisation term in the bound (17) and (18)?
   → By making no assumptions on the data Cesa-Bianchi et al. [20] presented an algorithm, where the performance guarantee on cumulative normalised square loss is given by using the generalised gradient descent. In order to obtain the performance guarantee (17), first a lower bound on the progress (see Lemma IV.4 in [20]) is computed by assuming that $\eta = \frac{\alpha}{\|x_t\|^2}$ (see Theorem IV.2 in [20]), where $0 < \alpha < 2$. The chosen α ensures that the performance guarantee on the normalised cumulative loss is held. Chapter V dwell further on the discussion done by Cesa-Bianchi et al. [20] and do not impose a similar restriction

on η when bounding the progress of the proposed algorithm CNLS. In Chapter V, a performance guarantee comparable to (17) is shown first by bounding the input. Later, the algorithm is studied by considering no conditions on the input, output and wights with the normalised square loss. Consequently, the proposed algorithm's guarantees have the tuning parameter next to the ridge penalty, implying a superior bias variance trade-off properties than the generalised GD update rule.

4. If the answer to the third question is yes, then at what cost? Else, why not?
   → For the case when the regularisation parameter is set to one, the CNLS bound is 4 times worse than the true regression function instead of 2.25 obtained using generalised GD approach by Cesa-Bianchi et al. [20], for further details see Chapter V.

5. Does knowing the answers to the above questions have any implications?
   → The implication are given in Chapter VII.

4- Publications

1. Waqas Jamil, N-C Doung, Wenjuan Wang, Chemseddine Mansouri, Saad Mohamad and Abdelhamid Bouchachia, "Scalable online learning for flink: SOLMA library", *European Conference on Software Architecture 2018* (published)

2. Waqas Jamil and Abdelhamid Bouchachia, "Competitive Regularised Regression", *Neurocomputing*, ELSEVIER (published)

3. Waqas Jamil and Abdelhamid Bouchachia, "Online Bayesian Shrinkage Regression", *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning 2019* (published)

4. Waqas Jamil and Abdelhamid Bouchachia, "Online Bayesian Shrinkage Regression", *Neural Computing and Applications*, Springer (published)

5. Waqas Jamil and Abdelhamid Bouchachia, "Competitive Normalised Least Squares Regression", *IEEE Transaction on Neural Networks and Learning Systems* (accepted with revisions)

5- Organisation

The structure of the thesis is as follows:

- Chapter II contains three sections, where the first section gives some background on the tools used in the later chapters. The second section discusses the relevant literature in detail. The final section presents the problem precisely.

- Chapter III and IV are based on the material from the publications $2, 3, 4$ mentioned in the previous section. Chapter III presents the derivation and detailed analysis of OSLOG. The structure of Chapter IV is similar to Chapter III, but instead of OSLOG, the CIRR algorithm's derivation and analysis are done. Also, an explicit link between OSLOG and CIRR is given.

- Chapter V is based on publication number 5, containing the derivation and analysis of CNLS algorithm.

- An empirical study is given in Chapter VI. The main aim of the study is to give an illustration of the performance of algorithms relative to the optimal offline solution. The code is available publicly, please see publication number 1.

- The conclusion and future directions are given in Chapter VII.

II

Competitive Regression

Prediction is very difficult, especially if it's about the future.

—Nils Bohr

1- Background

In this section, *Game-theoretic* and *Bayesian* learning are briefly discussed respectively to make this work self contained. The *Game-theoretic* learning approach is presented in light of the following example [79].

Example 2. *Consider a zero-sum perfect information game defined by matrix $G_{i,j} \in \mathbb{R}^{n \times n}$ such that n is some finite integer. The game is played between two players A and B. A chooses a strategy α over the rows and simultaneously B chooses a strategy β that is over columns. So, for a sequential play let i, j be pure strategies[1] and α, β be the mixed strategies[2]. If A plays before B and A chooses α, then B will pick β to maximise $A_{\alpha,\beta}$, so the loss L will be:*

$$L_\alpha = max_\beta G_{\alpha,\beta} = max_j G_{\alpha,j} \tag{19}$$

So, A should pick α to minimise L_α and the loss will be:

$$min_\alpha L_\alpha = min_\alpha max_\beta G_{\alpha,\beta} = min_\alpha max_j G_{\alpha,j} \tag{20}$$

Similarly, if B plays first loss will be $max_\beta min_\alpha G_{\alpha,\beta}$. Notice playing second can not be much worse than playing first i.e.

$$\underbrace{min_\alpha max_\beta G_{\alpha,\beta}}_{\texttt{A plays first}} \geq \underbrace{max_\beta min_\alpha G_{\alpha,\beta}}_{\texttt{A plays second}} \tag{21}$$

From Von Neumann's minimax theorem [96] the value (v) of the game is as follows:

$$min_\alpha max_\beta G_{\alpha,\beta} = max_\beta min_\alpha G_{\alpha,\beta} = v \tag{22}$$

which implies order does not matter and the result will always be the same.

Remark 2. *Even if player B knows player A's strategy, player B can't get better outcome than v. It is the best possible value. This can be more formally translated as: $\exists$ min-max strategy $\alpha^* =$ arg min-max$G_{\alpha,\beta}$ such that for any β, $G_{\alpha^*,\beta} \leq v$.*

Remark 3. *Regardless of player A's strategy, the outcome at worst is v i.e. $\exists$ max-min strategy β^* =arg max-min$G_{\alpha,\beta}$ such that for any α, $G_{\alpha,\beta^*} \geq v$.*

[1] A pure strategy determines all your moves during the game (and should therefore specify your moves for all possible other players' moves).

[2] A mixed strategy is a probability distribution over all possible pure strategies (some of which may get zero weight). After a player has determined a mixed strategy at the beginning of the game, using a randomising device, that player may pick one of those pure strategies and then stick to it.

For a two player finite zero sum game the best strategy against the opponent is to find the min-max strategy and always play it. Notice for Example 2, the Nash equilibrium is (α^*, β^*). In any finite game with a finite number of pure strategies and a mixed-strategy Nash equilibrium is guaranteed to exist [76].

What was discussed until now was the classical game theory. Without further ado, a direct link between online learning and game theory is presented.

Protocol 2: A two player repeated game

```
FOR t=1,2,...
  (1) Player $A$ chooses $\alpha_t$
  (2) Player $B$ chooses $\beta_t$
  (3) Player $A$ loss is $f(\alpha_t, \beta_t)$
  (4) Player $A$ observes loss $f(i, \beta_t)$
      for each pure strategy $i$
END FOR
```

Protocol 2 matches with Protocol 1 for two experts ($N = 2$). Now incorporating the concept of experts and extending the Protocol 2 to N players a protocol synonymous to Protocol 1 can be obtained, which is as follows:

Protocol 3: A prediction game with experts advice

```
FOR t=1,2,...
  (1) Input $x_t \in \mathbb{R}^n$ arrives
  (2) Experts strategy $w \in \mathbb{R}^n$
  (3) Prediction $\hat{y}_t$ made using strategy $w$
  (4) Receive actual output $y_t$
  (5) Observe loss $f(y_t, \hat{y}_t)$
END FOR
```

Popular algorithms like HA [67], WMA [68], aggregating algorithm (AA) [110], etc. operate under these protocols to ensure a guarantee on the performance. These algorithms are fine examples of the game-theoretic framework of probability. For further details on the mathematics of game-theoretic probability, please see [108, 112, 109, 90, 91].

Ahead is a briefly overview of an alternative approach (Bayesian learning) to Protocol 3. Conventionally, probability theory considers a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and then $\mathcal{F}$ is a $\sigma-$algebra on sample space Ω, and $\mathbb{P}$ is the probability measure on $(\Omega, \mathcal{F})$. Also, $\mathcal{F}$ is a $\sigma-$algebra of subsets of Ω.

Definition 4. *For two events $A, B \in \Omega$, $\mathbb{P}(A|B)$ and $\mathbb{P}(B|A)$ is:*

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \;\;\&\;\; \mathbb{P}(B|A) = \frac{\mathbb{P}(B \cap A)}{\mathbb{P}(A)} \tag{23}$$

such that $\mathbb{P}(A)$ & $\mathbb{P}(B) \neq 0$.

Example 3. *A simple manipulation of (23) leads to the Bayes rule [4, 9]:*

$$\mathbb{P}(A \cap B) = \mathbb{P}(B|A)\mathbb{P}(A) \Rightarrow \mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)} \tag{24}$$

which is at the heart of Bayesian theory.

Bayes rule mentioned in Example 3 can be considered in both discrete and in continuous sense, without making the assumption of i.i.d. Often, in literature the terms $\mathbb{P}(A|B)$, $\mathbb{P}(B|A)$, $\mathbb{P}(A)$ and $\mathbb{P}(B)$ in (24) are referred as the: posterior probability, likelihood, prior and normalising constant respectively. In continuous case, measure theory is needed due to the use of the integrals on the likelihood times prior and the normalising constant, but this use of measurability does not lead to any distributional assumptions.

Bayesian learning strategy mentioned in Protocol 4 often provides the probabilistic interpretation to the game-theoretic algorithms and helps in further understanding of the algorithms.

Protocol 4: Bayesian strategy

```
FOR t=1,2,...
    (1) Input $x_t \in \mathbb{R}^n$ arrives
    (2) Predict $\hat{y}_t$ by computing posterior
    predictive distribution
    (3) Actual output $y_t \in \mathbb{R}$ arrives and loss
    is observed to update $w$ in the next round
END FOR
```

2- Related work

Vovk [113], Azoury and Warmuth [8] enriched the literature of learning theory and performed rigorous analysis on a variant of RLS regression to know its predictive power. They presented the following type of bound by confining the input and weights to the unit balls in the metrics ℓ_∞ and ℓ_1:

$$L_T \leq L_T^* + \mathcal{O}(\ln T) \tag{25}$$

Forster [28], Cesa-Bianchi and Lugosi [21] obtain the same bound as above using a more compact approach. In this thesis, the algorithm with the bound (25), is referred as the aggregating algorithm for regression (AAR). RLS, AAR and other algorithms have some similarities, to notice them considering the following weights update rule of RLS:

$$w_t = argmin_w \left(\sum_{s=1}^{t} r^{t-s} (y_s - w' x_s)^2 \right)$$

At each iteration, t, the prediction $\hat{y}_t = w'_{t-1} x_t$ is made and after receiving the true output, y_t and then the weights are updated by using the following rule:

$$w = w + \frac{(y_t - x'_t w) A x_t}{r + x'_t A x_t} \tag{26}$$

Incorporating (26) in Protocol 4 leads to the Protocol of RLS:

Protocol 5: RLS

```
Initialise: $0 < r \leq 1$ and $A^{-1} = I \in \mathbb{R}^{n \times n}$
FOR t=1,2,...
    (1) Input $x_t \in \mathbb{R}^n$
    (2) $\hat{y} = x'_t w$
    (3) Observe $y_t \in \mathbb{R}$
    (4) $A^{-1} = r A^{-1} + x_t x'_t$
    (5) $w = w + \frac{(y_t - x'_t w) A x_t}{r + x'_t A x_t}$
END FOR
```

There exist many similar algorithms such as AAR, RR and adaptive regularisation of weights regression (AROWR). In particular, the weight update rule of AROWR is the same as RLS. The only difference is how the covariance matrix (see line (4) in Protocol 5) is updated: $A_t^{-1} = A_{t-1}^{-1} + \frac{1}{r} x_t x'_t$. AAR's and RR's update rules can be obtained by setting $r = 1$ in (26) with $A_t^{-1} = A_{t-1}^{-1} + x_t x'_t$ and initialisation $A = a^{-1} I$, where $a > 0$. The main difference between AAR and the other algorithms is that AAR divides its prediction $w'_{t-1} x_t$ by $1 + x'_t A_{t-1} x_t$, whereas RR, AROWR and RLS don't do that. So for AAR line (2) in Protocol 5 is replaced by $\hat{y}_t = \frac{x'_t w_{t-1}}{1 + x'_t A_{t-1} x_t}$. Thus, AAR is the only algorithm among the four that has the ability to perform shrinkage on prediction.

The implementation of (26) has the time complexity $\mathcal{O}(n^2)$ which is significant for high dimensional data. Often LS [118] is considered as a less demanding solution since its time complexity is $\mathcal{O}(n)$. LS replaces the term $\frac{A_{t-1}^{-1}}{r + x'_t A_{t-1}^{-1} x_t}$ by the learning rate $\eta > 0$ yielding the following update weights estimate w at time t:

$$w_t = w_{t-1} + \eta (y_t - x'_t w_{t-1}) x_t \tag{27}$$

The LS algorithm not only has a better time complexity, but it is also H^∞ optimal for $\eta \leq \frac{1}{\|x_t\|_2^2}$ i.e.:

$$\max_w \frac{\sum_{t=1}^T (w'x_t - w'_{t-1}x_t)^2}{\sum_{t=1}^T (w'x_t - y_t)^2 + \frac{\|x_t\|_2^2}{\eta}} \leq 1 \tag{28}$$

In contrast, in the case of RLS, the right side of (28) is replaced by 4. For further details on the matter, please see [42].

In practice, often normalised least squares (NLS) performs better than LS, because NLS is not sensitive to the scale of the input [11, 14]. The existing work on NLS applies a normalised square loss to derive the update of the weights [20, 57]:

$$w_t = w_{t-1} + \frac{\eta}{\|x_t\|_2^2}(y_t - \hat{y}_t)x_t \tag{29}$$

$$w_t = w_{t-1} + \frac{\eta}{1 + \eta\|x_t\|_2^2}(y_t - \hat{y})x_t \tag{30}$$

for $\eta > 0$. When $x_t = 0$ or $\eta = 0$ with the convention that $\frac{0}{0} = 0$, the rules (29) and (30) output $w_t = w_{t-1}$.

In contrast to this, Cesa-Bianchi et al. [20] studied the bounds of generalised GD based online regression with square loss. Later Kivinen and Warmuth [57] replaced gradient descent by exponentiated gradient descent (EGD). The assumptions made in the GD approach are that for all data points and weights, ℓ_2 norm is bounded by 1. For EGD, it is assumed that ℓ_∞ and ℓ_1 norm for data points and weights are bounded by 1. GD based regression is usually computationally efficient. However, its fundamental disadvantage is that the difference between the learner and the best linear regression function (L_T^*) is bounded by $\mathcal{O}(1)$ under online setting. An example of the upper bound on the cumulative square loss of a GD-based linear regression algorithm is as follows [20]:

$$\sum_{t=1}^T (\hat{y}_t^{GD} - y_t) \leq 9 \inf_{w \in \mathbb{R}^n} \left(\sum_{t=1}^T (w'x_t - y_t)^2 + \sup_{t=1,\dots,T} \|x_t\|_\infty^2 \|w\|_2^2 \right) \tag{31}$$

where $\hat{y}_t^{GD}$ denotes the prediction at step t. For the noise-free case, by assuming $\|x_t\|_\infty \leq R$, Inequality (31) reduces to [20]:

$$\sum_{t=1}^T (\hat{y}_t^{GD} - y_t) \leq 2.25 \inf_{w \in \mathbb{R}^n} \left(\sum_{t=1}^T (w'x_t - y_t)^2 + R\|w\|_2^2 \right) \tag{32}$$

Inequality (31) and (32) are not comparable to the bounds obtained by Kivinen and Warmuth [57], but EGD has a much smaller loss if only few predictors are relevant to the prediction. AAR's upper bound on the cumulative loss of the learning algorithm for the noise- free case under the assumption $\|x_t\|_2 \leq R$ is not as good as inequality (32) [20]. However, in online setting like AAR's where true regression function is corrupted by Gaussian noise, the upper bounds on the cumulative loss derived by Cesa-Bianchi et al. [20], Kivinen and Warmuth [57] are of the following type:

$$L_T \leq L_T^* + \mathcal{O}(\sqrt{L_T^*}) \tag{33}$$

where L_T is the loss of the online algorithm at trial T, L_T^* is the loss of the best linear regression function at trial T. Using the GD and EGD approach, the difference $L_T - L_T^*$ is at best bounded by $\sqrt{T}$ that requires *a priori* knowledge about L_T^*. Orabona et al. [77] obtained the upper loss bound using online Newton step of the of type:

$$L_T \leq L_T^* + \mathcal{O}(\ln L_T^*) \tag{34}$$

Inequality (34) is overall better than AAR's upper loss bound when T is large and when L_T^* grows sub-linearly. This is because for the case $L_T^* = 0$, $L_T \leq \mathcal{O}(1)$ and at most $L_T^* = \mathcal{O}(T)$. For AAR's upper bound on the cumulative loss when $L_T^* = 0$, is $L_T - L_T^* \leq \mathcal{O}(\ln T)$. However, upper bound on cumulative loss proven in [77] requires prior knowledge about $\|w_t\|_1 \leq P$ and the multiplicative factor of their bound is $(PR + Y)^2$, which is strictly greater than AAR's upper bound on the cumulative loss [114] multiplicative factor of Y^2. AAR and the algorithm proposed by Orabona et al. [77] both have computational complexity of $\mathcal{O}(n^2)$.

Anava et al. [5] and Liu et al. [70] proposed GD based approach for auto-regression. Their game-theoretic version of auto regressive moving average (ARMA) algorithms use online GD and online newton step (ONS), that allow the noise term to be unbounded. It is worth noting that the algorithms based on GD have a worse bound but are computationally more efficient ($\mathcal{O}(n)$) than ONS based algorithms.

Monti et al. [74] used coordinate descent (CD) to deal with ℓ_1 regularisation. The distinct feature of the algorithm is that it can handle non-stationarity, but with no mention of loss bounds. Few algorithms can handle non-stationarity and give a competitive prediction. For example, Moroshko et al. [75] extended AAR by using Forster [28] methodology and called it last step adaptive regression algorithm (LASER). They also considered extension of the algorithms discussed in [23, 105]. Authors in [54] proved similar bound to LASER by extending Busuttil and Kalnishkan [17] work.

Recently, Rajaratnam et al. [85] presented a recursive Bayesian deterministic algorithm that performs ℓ_1 regularisation by considering the limit of Gibbs sampling, along with its bounds on convergence. Also, Langford et al. [65] developed an online learning algorithm by replacing the gradient of losses by the sub-gradient of losses in stochastic gradient descent, showing that such algorithm has a strong theoretical guarantee for bounded loss functions and weights. Using topology and considering homotopy of LASSO, Garrigues and Ghaoui [35] proposed an online regression algorithm. However, they did not study the bounds.

3- Problem formulation

The description of the linear benchmark or the reference function is defined as follows:

Definition 5. *A sequence of instances and their corresponding outcomes $(x_1, y_1), ..., (x_t, y_t)$ arrive sequentially and $w = w_t \in \Theta = \mathbb{R}^n$ denotes the decision strategy at time t. Where $w_{t,i}$, for $i = 1, ...n$ denote the $i-th$ component of the decision vector at time t.*

The input signal and the weight vector is defined in the following sense, unless stated otherwise.

Definition 6. *The $\ell_\infty-$ ball $\{x_t \in \mathbb{R}^n : \|x\|_\infty \leq R\}$ of radius R, the vector w_t is indexed by $\Theta = \{w_t \in \mathbb{R}^n : C \leq \|w_t\|_1 \leq P\}$ and the prediction on trial t is given by $w_t' x_t$.*

Following definition, defines some of the notation extensively used.

Definition 7.

$$b_t := \sum_{s=1}^{t} y_s x_s \in \mathbb{R}^n \tag{35}$$

$$A_t := \left(a D_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s' \right) \in \mathbb{R}^{n \times n},\ a > 0 \tag{36}$$

where,

$$D_{w_{t-1}} = diag(abs(w_{t-1})) = diag(|w_{t-1,1}|, ..., |w_{t-1,n}|) \tag{37}$$

letting w_0 to be initialised in $\mathbb{R}^n$ with uniform distribution. Also, we define the square loss as follows:

$$L_t = L_t(w) = \sum_{s=1}^{t} (y_t - w' x_t)^2 \ \ \& \ \ L_T^* := \min_w \|\mathbf{Y} - \mathbf{X} w\|_2^2 \tag{38}$$

where $w \in \mathbb{R}^n$,$\mathbf{X} \in \mathbb{R}^{t \times n}$, $y \in \mathbb{R}$ and $Y \in \mathbb{R}^t$. Also, in the thesis $L_t(.)$ is used to represent the squared loss of the algorithm, where $(.)$ is filled with the name of the learning algorithm. Denoting $\nabla f(w_t)$ for the first derivative and $\mathcal{H} \nabla f(w_t)$ (where $\mathcal{H}$ is for the Hessian matrix) for second derivative with respect to w_t.

The aim is to compete against the reference forecaster (38) to achieve an upper bound on the cumulative loss of the type stated in Definition 3. Often in literature ℓ_p norm penalty is considered for mathematical and statistical reasons. Mathematically, optimal weight update rule requires inverting covariance matrix which can be singular, while addition of a penalty prevents it, for example by adding squared ℓ_2 penalty, it is still possible to obtain a closed form solution, since the squared error and the squared norm commute. ℓ_1 and ℓ_2 regularisation are the two most popular regularisation methods for regression in the statistical and computer science literature. ℓ_1 regularised regression aims to obtain a sparse solution. ℓ_1 regularised regression [44, 47] is useful when the requirement is to obtain few non-zero entries in the output. For instance, [39] obtains sparsity in the model by considering prior belief about the sparsity in the model. On the other hand, ℓ_2 regularised regression increases the bias, has lower variance than regression without regularisation and is a useful technique for dealing with multicollinearity. Often statistical literature refers to ℓ_1 regularised regression as Least Absolute Shrinkage and Selection Operator (LASSO) and ℓ_2 regularised regression as Ridge Regression (RR). Both ℓ_1 and ℓ_2 regularisation have their advantages and disadvantages. For detailed explanation see [104]. Formally, following optimisation is considered:

$$\inf_{w \in \mathbf{R}^n} \left(L_t + a\|w\|_p^j \right) \tag{39}$$

where $a > 0$ and $p, j = 1, 2$. The problem (39) with $j = p = 1$ is very difficult to bound because ℓ_1 norm is non differentiable but is convex. Hence, we may use sub-differentials to differentiate, but the problem is that the sub-differentiation of ℓ_1 norm does not lead to a unique dual vector [89, 104, 27]. Schmidt [89], Tibshirani [104], Fan and Li [27] proposed an approximation of ℓ_1 norm in batch setting by considering $w^{k+1} \in \mathbb{R}^n$ where k denotes the number of passes with the condition $w_i \neq 0$ for $i = 1, 2, ..., n$:

$$\|w^{k+1}\|_1 \approx \sum_{i=1}^{n} \frac{\left(w_i^{k+1}\right)^2}{|w_i^k|} = \|D_{w^k}^{-\frac{1}{2}} w^{k+1}\|_2^2 \tag{40}$$

such that $D_{w^k}^{-\frac{1}{2}} = \operatorname{diag}(1/\sqrt{|w_1^k|}, ..., 1/\sqrt{|w_n^k|})$. In [27], it is argued that (40) is a good approximation to ℓ_1 norm, due to its similarities to the Newton's method. Using (40) implies solving the following optimisation problem:

$$\inf_{w_t \in \mathbf{R}^n} \left(L_t + a\|D_{w_{t-1}}^{-\frac{1}{2}} w_t\|_2^2 \right) \tag{41}$$

for $|w_{t-1,1}|, ..., |w_{t-1,n}| \neq 0$. The restriction of the decision strategy w_t not being zero can be easily lifted by simple algebraic manipulation, which will be done latter, but for simplicity considering this restriction for now. Interestingly, in batch setting solving (41) coincides with a more recent algorithm known as shrinkage via limit of Gibbs (SLOG) presented by Rajaratnam et al. [85] where it is shown that under mild regularity assumptions, SLOG converges to LASSO.

In this work, algorithms are only allowed to make one pass over the data. For the ease of interpretation, Cauchy-Schwartz inequality is used, thus, the following reference forecaster is considered:

$$\inf_{w_t \in \mathbf{R}^n} \left(L_t + a\|D_{w_{t-1}}^{-\frac{1}{2}} w_t\|_2^2 \right) \leq \inf_{w_t \in \mathbf{R}^n} \left(L_t + aSS\|w_t\|_2^2 \right) \tag{42}$$

where SS is the sum of squares of the diagonal matrix $D_{w_{t-1}}$ elements. Later, it is shown that the inequality (42) holds. However, the bound will imply for (41). The use of (42) is for the ease of interpretation.

Remark 4. *For completeness, the notion of vector space, refers to the space $V \neq \emptyset$ set that must be closed under vector addition and scalar multiplication. In order to classify a vector space, the axiom of commutativity, associativity, additive identity and distributivity must hold for vectors and scalars [37]. In this work the vector space is almost always an n-dimensional Euclidean space $\mathbb{R}^n$ i.e. every element of this space is represented by a list of n real numbers with scalars in $\mathbb{R}$.*

Remark 5. *In this thesis ℓ_p norms for $1 \leq p < \infty$ are defined in a conventional manner i.e. ℓ_p norm of a vector $x_t \in \mathbb{R}^n$ with coordinates $x_{t,i}$ is $(\sum_{i=1}^{n} |x_{t,i}|^p)^{\frac{1}{p}}$ and $\|x_{t,i}\|_\infty = \max_i |x_{t,i}|$, for $i = 1, 2, ..., n$.*

III

OSLOG: Online shrinkage via limit of Gibbs

The most probable value of the unknown quantities will be that in which the sum of the squares of the differences between the actually observed and the computed values multiplied by numbers that measure the degree of precision is a minimum.

—Carl Friedrick Gauss

1- Derivation

The SLOG algorithm proposed by Rajaratnam et al. [85] maximises the posterior distribution $w \in \mathbb{R}^n$ given the response $\mathbf{y} \in \mathbb{R}^n$ i.e., $argmax_{w \in \mathbb{R}^n} p(w|\mathbf{y})$. It is assumed that $\mathbf{y}|w$ follows the normal distribution and w follows the Laplace or double exponential distribution. To derive SLOG, Rajaratnam et al. [85] tweaks the approach mentioned by Park and Casella [83] for Bayesian LASSO algorithm. Both SLOG and the Bayesian Lasso consider a hierarchical model by writing the Laplace distribution as a scale mixture of the Gaussian distribution [6]. The weight updating rule of the Bayesian LASSO is the joint posterior obtained through the hierarchical model. Then, it is shown that by using the Gibbs sampler on the joint posterior converges to the ℓ_1-regularisation regression solution. SLOG uses the same approach as the Bayesian Lasso with a different tuning parameter. SLOG replaces the tuning parameter $a > 0$ in (39) by $a\sqrt{\sigma^2}$ with known variance σ^2. Consequently, as the limit $\sigma^2 \to 0$ of the Gibbs sampler, it reduces to a deterministic sequence, giving the weight updating rule of SLOG. For OSLOG, same weight updating equation as SLOG is obtained but without the use of Gibbs Sampler.

The online protocol which assumes that at each trial the input arrives. Then, the algorithm predicts the outcome before the actual outcome is revealed and the adjustment of the weights is conducted. OSLOG follows the following protocol:

Protocol 6: OSLOG strategy

```
FOR $t = 1, 2, ...$
  (1) Read $x_t \in \mathbb{R}^n$
  (2) Learner prediction $\hat{y}_t \in \mathbb{R}$
  (3) Read $y_t \in \mathbb{R}$
  (4) Learner chooses weights $w \in \Theta$
END FOR
```

$$p(w) = \left(\frac{a\eta}{2}\right)^n \exp\left(-a\eta w' D^{-1}_{w_{t-1}} w\right) \tag{43}$$

The selected prior distribution on weights is inspired by the Laplace distribution which is written as [104]:

$$\frac{1}{2\tau} e^{\|w\|_1/\tau}, \quad \tau = \frac{1}{\lambda}, \; \lambda > 0$$

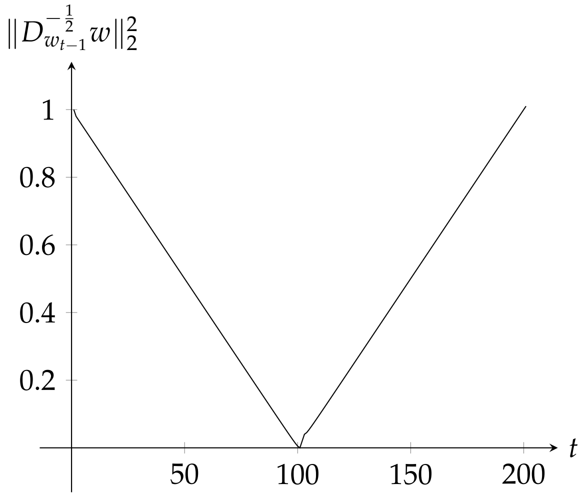

Figure 1: ℓ_1-norm approximation.

Considering: $\tau = \frac{1}{a\eta}$, with the scalar $\eta = \frac{1}{2\sigma^2}$ such that $a, \eta > 0$. Also, replacing $\|w\|_1$ by $\|D_{w_{t-1}}^{-\frac{1}{2}}w\|_2^2$. Clearly in the expression $\|D_{w_{t-1}}^{-\frac{1}{2}}w\|_2^2$ a restriction on weights is required. So, at trial $T-1$ absolute value of each element of the weight vector should not to be zero in (43). Despite this restriction Figure 1 shows reasonable similarity to $\|w\|_1$. A visible difference is near the kink point $(100, 0)$. The following lemma resolves the issue of $\frac{\mathbb{R}}{0}$:

Lemma 1. *For all $t = 1, 2, ...$*

$$\left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s'\right)^{-1}$$

$$= D_{w_{t-1}}^{\frac{1}{2}} \left(a\mathbf{I} + D_{w_{t-1}}^{\frac{1}{2}} \left(\sum_{s=1}^{t} x_s x_s'\right) D_{w_{t-1}}^{\frac{1}{2}}\right)^{-1} D_{w_{t-1}}^{\frac{1}{2}}$$

Proof.

$$\left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s'\right)^{-1} = \left(aD_{w_{t-1}}^{-\frac{1}{2}} D_{w_{t-1}}^{-\frac{1}{2}} + \sum_{s=1}^{t} x_s x_s'\right)^{-1}$$

$$= D_{w_{t-1}}^{\frac{1}{2}} \left(a\mathbf{I} + D_{w_{t-1}}^{\frac{1}{2}} \left(\sum_{s=1}^{t} x_s x_s'\right) D_{w_{t-1}}^{\frac{1}{2}}\right)^{-1} D_{w_{t-1}}^{\frac{1}{2}}$$

□

Lemma 2. *For any $x, b \in \mathbb{R}^n$ and a symmetric positive definite matrix A:*

$$x'Ax - 2b'x = (x - A^{-1}b)'A(x - A^{-1}b) - b'A^{-1}b$$

Proof. Expanding quadratic form:

$$(x - A^{-1}b)'A(x - A^{-1}b) = x'Ax - 2b'A^{-1}Ax + b'A^{-1}AA^{-1}b$$

$$= x'Ax - 2b'x + b'A^{-1}b$$

□

To obtain the predictive distribution for Normal/Gaussian likelihood with sequence S considering:

$$p(y|x_T, S_{T-1}) = \int_{\mathbb{R}^n} p(y|x_T, w)p(w|S_{T-1})\mathrm{d}w \tag{44}$$

where the (uniformly initialised) prior distribution is as defined in (43)and the posterior is:

$$p(w|S_{T-1}) = \frac{\left(\prod_{t=1}^{T-1} p(y_t|x_t, w)\right) p(w)}{\int_{\mathbb{R}^n} \left(\prod_{t=1}^{T-1} p(y_t|x_t, w)\right) p(w)\mathrm{d}w} \tag{45}$$

Thus, the predictive distribution at time T for y given the sequence $S_{T-1} = x_1, y_1, .., x_{T-1}, y_{T-1}$ requires evaluation of the following integral:

$$\frac{\int_{\mathbb{R}^n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(w'x_T - y)^2}{2\sigma^2}} \prod_{t=1}^{T-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(w'x_t - y_t)^2}{2\sigma^2}} \exp\left(-\frac{a}{2\sigma^2} w'D^{-1}_{w_{t-1}} w\right) \mathrm{d}w}{\int_{\mathbb{R}^n} \prod_{t=1}^{T-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(w'x_t - y_t)^2}{2\sigma^2}} \exp\left(-\frac{a}{2\sigma^2} w'D^{-1}_{w_{t-1}} w\right) \mathrm{d}w} \tag{46}$$

proceeding further in a more structured manner leads to the following results:

Lemma 3. *If an algorithm follows Bayesian strategy with Gaussian likelihood and prior (43) such that absolute value of the each element of the weight vector is not zero, w_0 is initialised uniformly and $a > 0$, then the posterior distribution is:*

$$\mathcal{N}\left(\left(\sum_{t=1}^{T-1} x_t y_t\right)' \left(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}}\right)^{-1}, \frac{1}{2\sigma^2} \left(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}}\right)^{-1} \right)$$

Proof. Expanding posterior (45), by using (43) and ignoring the normalising constant to get:

$$p(w|S_{T-1}) \propto \exp\left(-\eta \sum_{t=1}^{T-1} (y_t - w'x_t)^2 - a\eta w'D^{-1}_{w_{t-1}} w\right)$$

$$= \exp\left(-\eta\left(w'\left(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}}\right) w - 2w'\sum_{t=1}^{T-1} x_t y_t + \sum_{t=1}^{T-1} y_t^2\right)\right)$$

$$
\begin{aligned}
&= \exp \Bigg(-\eta \Bigg(w - \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \Bigg)' \\
&\bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg) \Bigg(w - \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \Bigg) \\
&\qquad - \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg) \\
&\qquad + \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg) + \sum_{t=1}^{T-1} y_t^2 \Bigg)
\end{aligned}
$$

$$
\begin{aligned}
&\propto \exp \Bigg(-\eta \Bigg(w - \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \Bigg)' \\
&\bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg) \Bigg(w - \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \Bigg)
\end{aligned}
\tag{47}
$$

The last and the second last equality follows from (38) and (48) respectively. The last proportionality (47) can be recognised as probability density function of the multivariate Normal distribution. □

Theorem 4. *If an algorithm follows a Bayesian strategy with Gaussian likelihood and prior (43) such that weights at trial $T-1$ are not null, w_0 is initialised uniformly and $a > 0$, then the predictive distribution is expressed as:*

$$
\mathcal{N}\Bigg(\Big(\textstyle\sum_{t=1}^{T-1} x_t y_t\Big)' \Big(\textstyle\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}}\Big)^{-1} x_T, \frac{1}{2\sigma^2} x_T \Big(\textstyle\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}}\Big)^{-1} x_T \Bigg)
$$

Proof. From Lemma 2 immediately follows:

$$
\begin{aligned}
&w' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg) w - 2w' \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg) = \\
&\Bigg(w - \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \Bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg) \\
&\qquad \Bigg(w - \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \Bigg) - \\
&\qquad\qquad \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)' \bigg(\sum_{t=1}^{T-1} x_t x_t' + aD^{-1}_{w_{t-1}} \bigg)^{-1} \bigg(\sum_{t=1}^{T-1} x_t y_t \bigg)
\end{aligned}
\tag{48}
$$

The posterior distribution obtained in Lemma 3 can be thought of an online variant of the posterior obtained by Park and Casella [83]. Since, the posterior predictive distribution is a weighted average over parameter space where each parameter is weighted by its posterior probability (see (44) and for further details see for example [86]), thus the predictive distribution is as follows:

$$\mathcal{N}\left(\left(\sum_{t=1}^{T-1} x_t y_t\right)'\left(\sum_{t=1}^{T-1} x_t x_t' + aD_{w_{t-1}}^{-1}\right)^{-1} x_T,\right.$$
$$\left.\frac{1}{2\sigma^2} x_T \left(\sum_{t=1}^{T-1} x_t x_t' + aD_{w_{t-1}}^{-1}\right)^{-1} x_T\right)$$

□

By applying Lemma 1, the condition on weights (mean of posterior, see Lemma 3) can be lifted and an explicit algorithm for OSLOG is obtained, which is as follows:

Protocol 7: OSLOG

```
Initialise: $a > 0, M = \mathbf{0}^{n\times n}$, $b = \mathbf{0}^{n\times 1}$ and $w = \mathbf{1} \in \mathbb{R}^{n\times 1}$
FOR $t = 1, 2, \ldots$
    (1) Read $x_t \in \mathbb{R}^n$
    (2) $D_{w_{t-1}} = \mathtt{diag}(\mathtt{abs}(w))$
    (3) $\hat{y}_t = w'x_t$
    (4) $M = M + x_t x_t'$
    (5) $A^{-1} = \sqrt{D_{w_{t-1}}}\left(a\mathbf{I} + \sqrt{D_{w_{t-1}}}M\sqrt{D_{w_{t-1}}}\right)^{-1}\sqrt{D_{w_{t-1}}}$
    (6) Read $y_t \in \mathbb{R}$
    (7) $b = b + y_t x_t$
    (8) $w = A^{-1}b$
END FOR
```

Remark 6. *In Algorithm 7 line 8 can be allowed to make passes until convergence to have higher level of sparsity. From the sequential compactness theorem (see for example [61]) it follows that any closed and bounded sequence in Euclidean space converges. Further details can be found in [1, 87, 103]. Theorem 8 in [85] shows that SLOG converges to the LASSO solution under some regularity conditions.*

Notice the matrix A^{-1} in Algorithm 7 is symmetric and positive definite, so its inverse exists at each trial. At each trial, the system of equations solved is unique without making any stochastic assumptions. However, calculating the posterior predictive distribution involves measures and integrals. Therefore, assuming consistency with the topological space, prediction space is a topological space equipped with $\sigma-$algebra, and the set of parameter $w \in \Theta = \mathbb{R}^n$ is equipped with $\sigma-$ algebra[1].

Theorem 4 implies that the prediction of Algorithm 7 corresponds to the mean of the posterior predictive parameter w weighted by the posterior probability [86]. Interestingly, Kivinen and Warmuth [56] showed that the likelihood of the weighted average can be interpreted as the loss of the Online Bayesian Strategy.

2- Analysis

Next, the analysis of the OSLOG algorithm are presented, following are some useful results that are used to prove some of the main theorems.

Lemma 4. *For $D \in \mathbb{R}^{m\times n}$ with entries a_{ij} and $w \in \mathbb{R}^n$ with entry w_j*

$$\|Dw\|_2^2 \leq \|D\|_F^2 \|w\|_2^2$$

Proof. From Cauchy-Schwartz inequality:

$$\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij})^2\right)\sum_{k=1}^{n} w_k = \sum_{i=1}^{m}\left(\sum_{j=1}^{n}(a_{ij})^2\sum_{k=1}^{n}(w_k)^2\right) \geq \sum_{i=1}^{m}\left(\sum_{j=1}^{n}a_{ij}w_j\right)^2$$

□

Remark 7. *For $n = m$ in Lemma 4*

$$\left(\sum_{i=1}^{m}\sum_{j=1}^{m}(a_{ij})^2\right)\sum_{k=1}^{m} w_k \geq \sum_{i=1}^{m}\left(\sum_{j=1}^{m}a_{ij}w_j\right)^2$$

Notice $\|D\|_F^2$ by definition is $Tr(DD^H)$ (where Tr denotes the trace of a matrix and D^H is the conjugate transpose). In other words, $\|D\|_F^2$ is the Sum of Squares (SS) of the absolute value of the entries of D. Also, if D is a diagonal matrix then $\|D\|_F^2$ is simply the sum of squares of the diagonal elements. This justifies the Inequality (42).

Lemma 5. *For all $t = 1, 2, ..., T$, then*

$$\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \leq \|w\|_2^2$$

provided that every element of $w \geq 1$.

Proof. Since every element of $w \geq 1$, and $0 < \|D_{w_{t-1}}^{-\frac{1}{2}}\|_2^2 \leq 1$. Therefore, the above inequality holds. □

[1] This is a mild assumption which is always satisfied in practice. Not making such assumption will lead to counter intuitive results such as Banach-Tarski paradox. For details see, for example, [103]

Theorem 5. *For any point in time $t = 1, 2, ..., T$*

$$L_T(OSLOG) \le \inf_{w \in \mathbb{R}^n} \left(L_T + aSS\|w\|_2^2\right) + 4Y^2 \ln\det\left(\frac{1}{a}A_{T-1}\right) \tag{49}$$

where $a > 0$, $Y > 0$,$n \in \mathbb{N}^+$. Also, if $\|x_t\|_\infty \le R$ and $C \le \|w\|_1 \le P$, such that $C \neq 0$, $|w_{t,i}| \neq 0$ $\forall i = 1, 2, ..., n$ then $\forall t$:

$$L_T(OSLOG) \le L_T^* + aP^2C^{-1} + 4Y^2 n \ln\left(C^{-1} + \frac{TR^2}{a}\right) \tag{50}$$

provided that all $y_t \in [-Y, Y]$.

Proof. From Theorem 1 in [113] and Theorem 3 in [28] the upper bound on the cumulative loss of AAR is as follows:

$$L_T(AAR) \le \inf_{w \in \mathbb{R}^n} \left(w'B_Tw - 2w'b_T + \sum_{t=1}^{T} y_t^2\right) + Y^2 \ln\det\left(\frac{1}{a}B_T\right) \tag{51}$$

where $B_T = \left(a\mathbf{I} + \sum_{t=1}^{T} x_t x_t'\right)$ and $w'B_Tw - 2w'b_T + \sum_{t=1}^{T} y_t^2 = L_T^* + \|w\|_2^2$, here b_T and L_T^* are defined as (35) and (38) respectively. Notice (51) is only true for positive definite matrices and A_T is positive definite so, replacing $Y^2 \ln\det\left(\frac{1}{a}B_T\right)$ with $Y^2 \ln\det\left(\frac{1}{a}A_T\right)$. To elaborate further, expanding and performing some algebraic manipulation on the function $f(w)$ in Lemma 1 to obtain:

$$y_t^2 x_t' A_{t-1}^{-1} x_t + b_{t-1}(A_t^{-1}x_tx_t'A_t^{-1} - A_{t-1}^{-1} + A_t^{-1})b_{t-1} \tag{52}$$

Since, $A_{t-1} - A_t = x_t x_t'$, so $A_{t-1}^{-1} - A_t^{-1} = A_t^{-1}x_tx_t'A_{t-1}^{-1}$ and consequently $A_{t-1}^{-1} - A_t^{-1} - A_t^{-1}x_tx_t'A_t^{-1} = A_t^{-1}x_tx_t'A_t^{-1}x_tx_t'A_t^{-1}$. Thus, (52) can be written as:

$$y_t^2 x_t' A_t^{-1} x_t - (x_t'A_{t-1}^{-1}x_t)b_{t-1}'A_t^{-1}x_tx_t'A_t^{-1}b_{t-1} \tag{53}$$

It is easy to see that the term $(x_t'A_{t-1}^{-1}x_t)b_{t-1}'A_t^{-1}x_tx_t'A_t^{-1}b_{t-1}$ in (53) can be written as $(x_t'A_{t-1}^{-1}x_t)\hat{y}_t^2$ and,

$$y_t^2 x_t' A_t^{-1} x_t - (x_t'A_{t-1}^{-1}x_t)\hat{y}_t^2 \le Y^2 x_t A_t^{-1} x_t$$

summing over $t = 1, 2, ..., T$ and from Remark 3 in [113], the updating line 4 in Algorithm 9 after making the prediction is at most 4 times worse leads to the following expression:

$$L_T(OSLOG) - \inf_{w \in \mathbb{R}^n} \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2\right) \le 4Y^2 \sum_{t=1}^{T} x_t A_t^{-1} x_t \tag{54}$$

Notice since at $t = 0$, $D_{w_{t-1}} = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. So, $\ln\det\frac{1}{a}A_0 = 0$. For the case when $t = 1, 2, ..., T$ we need to show that $x_t'A_t^{-1}x_t \le \ln\frac{\det A_t}{\det A_{t-1}}$. For $x_t = 0$, clearly $x_t'A_tx_t < 1$ holds and for

$x_t \neq 0$ noticing $(x_t A_{t-1} x_t)^2 < x_t' A_t^{-1} x_t$. Now since A_t is symmetric positive definite thus the determinant of such matrix is bounded by the product of the entries on the diagonal (see for example [10] Theorem 7 from Chapter 2). Hence, $x_t A_t^{-1} x_t \leq \ln \frac{\det A_t}{\det A_{t-1}}$ holds, which indeed shows that by replacing $Y^2 \ln \det \left(\frac{1}{a} B_T\right)$ with $Y^2 \ln \det \left(\frac{1}{a} A_T\right)$ we obtain the bound stated in (74), when A_T is positive definite.

We use the definition of (35), (36), (37) and (38) to write the following:

$$w' A_T w - 2w' b_T + \sum_{t=1}^{T} y_t^2 = a \| D_{w_{T-1}}^{-\frac{1}{2}} w \|_2^2 + L_T$$

To prove (50) we first need to show the following holds:

$$w' A_T w - 2w' b_T + \sum_{t=1}^{T} y_t^2 \leq aSS \|w\|_2^2 + L_T$$

From Lemma 4:

$$\|D_{w_{T-1}}^{-\frac{1}{2}} w\|_2^2 \leq SS \|w\|_2^2 \leq \frac{P^2}{C} \tag{55}$$

By assuming that $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$ for $t = 1, 2, ..., T$, continuing as follows:

$$\ln \det \left(\frac{1}{a} A_T\right) = \ln \det \left(a D_{w_{T-1}}^{-1} + \sum_{t=1}^{T} x_t x_t'\right)$$

$$\leq \sum_{i=1}^{n} \ln \left(C^{-1} + \frac{TR^2}{a}\right) \leq n \ln \left(C^{-1} + \frac{TR^2}{a}\right) = n \ln \frac{a + CTR^2}{aC} \tag{56}$$

Replacing the term $\sum_{t=1}^{T} x_t A_t^{-1} x_t$ in (54) by (56) gives (50). □

The following theorem presents a scenario where OSLOG's upper bound is better than AAR and online ridge regression (ORR).

Theorem 6. *If $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$ such that $C \geq 1$, $a > 0$, and n is some positive integer, then $\forall t$ the following holds:*

$$L_T^U(OSLOG) \leq L_T^U(AAR)$$

where L_T^U denotes the upper cumulative square loss bound.

Proof. Letting:

$$R_T^*(AAR) = L_T^U(AAR) - L_T^*$$

$$R_T^*(OSLOG) = L_T^U(OSLOG) - L_T^*$$

Proceeding by showing $R_T^*(OSLOG) \leq R_T^*(AAR)$ i.e.

$$aP^2 C^{-1} + nY^2 \ln \left(\frac{a + CTR^2}{aC}\right) - aP^2 - nY^2 \ln \left(\frac{a + TR^2}{a}\right) \leq 0$$

$$aP^2\left(\frac{1}{C}-1\right)+nY^2\ln\left(\frac{a+TCR^2}{aC+TCR^2}\right)\leq 0$$

Since, $C \geq 1$ and $a, n > 0$, so $P^2(\frac{1}{C}-1) \leq 0$. Also, $a+TR^2 \leq aC+TCR^2 \implies \ln\frac{a+TCR^2}{aC+TCR^2} \leq 0$, thus the above inequality holds. Since $R_T^*(ORR) \geq R_T^*(AAR) \implies R_T^*(OSLOG) \leq R_T^*(ORR)$. □

Remark 8. *The regret of OSLOG is smaller than that of the regret of AAR when $C > 1$.*

Following results support Theorem 5 and Theorem 6 further in a more detailed, precise and insightful manner:

Lemma 6. *For prior (43) at time $t = 1, 2, ...$ the cumulative loss of OSLOG is:*

$$L_t(OSLOG) = \log_\beta \int_{\mathbb{R}^n} \beta^{L_T^*} p(w)\mathrm{d}w$$

where $\beta = e^{-\eta}$.

Proof. Noticing that Bayesian Strategy Q such that $\{Q_w | w \in \mathbb{R}^n\}$ with prior $p(w)$ is defined by:

$$Q = \int_{\mathbb{R}^n} Q_w p(w)\mathrm{d}w$$

So, the main statement of the Lemma is the definition of $\log_\beta Q$. Hence, it holds by the definition of the Bayesian decision rule. This is a popular approach for Online Bayesian algorithms, see for example [52]. □

Theorem 7. *For any trial $t = 1, 2, ..., T$, any $a > 0$ the following holds:*

$$L_T(OSLOG) \leq \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right) + Y^2\left(2n\ln\left(\tfrac{16Y^2}{a\sqrt{\pi}}\right) + \ln\det\tfrac{A_T}{8Y^2}\right) \quad (57)$$

where $y_t \in [-Y, Y]$ such that $Y > 0$ and absolute value of each element of the weight vector at $T-1$ is not zero.

From [56] the equality " $=$ " in the last lemma is replaced by the inequality " $\leq$ " for $\eta = \frac{1}{8Y^2}$ such that and the outcomes are bounded in $[-Y, Y]$. In other words for any value of $\eta > \frac{1}{8Y^2}$, $\beta^{(y_t - w'x_t)^2}$ will not be concave for $w'x_t$.

The problem is reduced to evaluating the integral of Lemma 6. For direct evaluation of the integral see Theorem 3 of Chapter 2 in [10].

$$\log_\beta \int_{\mathbb{R}^n} dw \left(\frac{a\eta}{2}\right)^n \times \exp\left(-\eta w'\left(\sum_{s=1}^{t} x_s x_s' + aD_{w_{t-1}}^{-1}\right)w + 2\eta\left(\sum_{s=1}^{t} y_s x_s\right)w - \eta\sum_{s=1}^{t} y_s^2\right) \quad (58)$$

Remark 9. *The integral to be calculated is of the form:*

$$\int_{\mathbb{R}^n} e^{-f(w)} dw = e^{-f_0} \frac{\pi^{n/2}}{\sqrt{\det A}}$$

where $f_0 = \inf_w f(w)$. Notice,

$$f(w) = -\left(\sum_{s=1}^{t} 2y_s(w'x_s)\right) + w'\left(aD^{-1}_{w_{t-1}} + \sum_{s=1}^{t} x_s x_s'\right) w + \sum_{s=1}^{t} y_s^2$$

Proceeding by differentiating with respect to w:

$$\nabla f(w) = 0 - \left(\sum_{s=1}^{t} 2y_s x_s\right) + 2w'\left(aD^{-1}_{w_{t-1}} + \sum_{s=1}^{t} x_s x_s'\right)$$

and clearly the second differential is negative implying the infimum is attained and by substitution the result is obtained.

From (58) and as per the above Remark:

$$\begin{aligned}
L_T(OSLOG) &= \log_\beta \int_{\mathbb{R}^n} dw \left(\frac{a\eta}{2}\right)^n \\
&\times \exp\left(-\eta w'\left(\sum_{t=1}^{T} x_t x_t' + aD^{-1}_{w_{t-1}}\right) w + 2\eta\left(\sum_{t=1}^{T} y_t x_t\right) w - \eta \sum_{t=1}^{T} y_t^2\right) \\
&= \log_\beta e^{-\eta \inf\left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right)} \frac{\pi^{n/2}}{\det \eta \left(\sum_{t=1}^{T} x_t x_t' + aD^{-1}_{w_{t-1}}\right)} \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) + \log_\beta \left(\left(\frac{a\eta}{2}\right)^n \frac{\pi^{n/2}}{\sqrt{\det \eta A_T}}\right) \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) + \log_\beta \left(\left(\frac{a\eta}{2}\right)^{\frac{2n}{2}} \frac{\pi^{n/2}}{\sqrt{\det \eta A_T}}\right) \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) - \frac{1}{2} \log_\beta \left(\left(\frac{2}{a\eta}\right)^{2n} \frac{\det \eta A_T}{\pi^n}\right) \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) - \frac{1}{2} \log_\beta \left(\left(\frac{4}{a^2\eta^2\pi}\right)^{n} \det \eta A_T\right) \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) - \frac{1}{2} \frac{\ln\left(\left(\frac{4}{a^2\eta^2\pi}\right)^{n} \det \eta A_T\right)}{\ln \beta} \\
&\leq \inf_w \left(L_T^* + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) - \frac{1}{2} \frac{\ln\left(\left(\frac{16Y^4}{a^2\pi}\right)^{n} \det \frac{A_T}{8Y^2}\right)}{-\frac{1}{8Y^2}} \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) + Y^2 \ln\left(\left(\frac{256Y^4}{a^2\pi}\right)^{n} \det \frac{A_T}{8Y^2}\right) \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) + Y^2 n \ln\left(\frac{256Y^4}{a^2\pi}\right) + Y^2 \ln \det \frac{A_T}{8Y^2} \\
&= \inf_w \left(L_T + a\|D^{-\frac{1}{2}}_{w_{t-1}} w\|_2^2\right) + Y^2 \left(2n \ln\left(\frac{16Y^2}{a\sqrt{\pi}}\right) + \ln \det \frac{A_T}{8Y^2}\right) \qquad (59)
\end{aligned}$$

AAR mentioned in [113] has the following guarantee:

$$L_T(AAR) \leq L_T^* + aP^2 + nY^2 \ln\left(1 + \frac{TR^2}{a}\right) \tag{60}$$

and the guarantee of OSLOG is as follows:

Corollary 1. *For any trial $t = 1, 2, ..., T$ and any $a > 0$ such that $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$, the following holds:*

$$L_T(OSLOG) \leq L_T^* + aP^2C^{-1} + nY^2 \ln\left(\frac{32Y^2(a + CTR^2)}{a^2C\pi}\right)$$

for $y_t \in [-Y, Y]$, such that $Y > 0$ and $C \neq 0$.

Proof. Bounding $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$ for $t = 1, 2, ..., T$. Denoting elements of diagonal matrix $D_{w_{t-1}}$ by d_{ij}. Now, upper bounding the following expression:

$$\ln \det A_T = \ln \det \left(aD_{w_{t-1}}^{-1} + \sum_{t=1}^{T} x_t x_t'\right)$$

Using Beckenbach and Bellman [10] Theorem 7 (in Chapter 2) to bound the determinant i.e.:

$$\ln \det A_T \leq \ln \prod_{i=1}^{n} \left(\frac{a}{d_{ii}} + \sum_{t=1}^{T} (x_{t,i})^2\right) \leq \sum_{i=1}^{n} \ln\left(aC^{-1} + TR^2\right)$$

$$\ln \det A_T \leq n \ln\left(aC^{-1} + TR^2\right) = n \ln \frac{a + CTR^2}{C} \tag{61}$$

Since:

$$\begin{aligned} L_T(OSLOG) &\leq \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2\right) \\ &+ Y^2 \left(2n \ln \frac{16Y^2}{a\sqrt{\pi}} + n \ln \frac{a + CTR^2}{8Y^2C}\right) \\ = \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2\right) &+ Y^2 \left(n \ln \frac{256Y^4}{a^2\pi} + n \ln \frac{a + CTR^2}{8Y^2C}\right) \\ = \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2\right) &+ Y^2 \left(n \ln \left(\frac{256Y^4(a + CTR^2)}{8a^2\pi Y^2C}\right)\right) \\ = \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2\right) &+ Y^2 \left(n \ln \left(\frac{32Y^2(a + CTR^2)}{a^2C\pi}\right)\right) \end{aligned}$$

The rest follows from (55). □

Theorem 8. *If $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$ such that $C \geq 1$, $a \geq \frac{32Y^2}{\pi}$, and n is some positive integer, then $\forall t$, the following holds:*

$$L_T^U(OSLOG) \leq L_T^U(AAR)$$

where $L_T^U(.)$ denotes the upper bound on the cumulative square loss.

Proof. Showing that $L_T^U(OSLOG) - L_T^U(AAR) \leq 0$. From (60) and Corollary 1:

$$aP^2\left(\frac{1}{C}-1\right)+nY^2\ln\left(\frac{32Y^2(a+CTR^2)}{a^2C\pi}\right)-nY^2\ln\left(\frac{a+TR^2}{a}\right)\leq 0$$

$$aP^2\left(\frac{1}{C}-1\right)+nY^2\ln\frac{32Y^2(a+CTR^2)}{aC\pi(a+TR^2)}\leq 0$$

For $C \geq 1$, $aP^2\left(\frac{1}{C}-1\right) \leq 0$. It is clear that $\|w\| \geq \|D_w^{-\frac{1}{2}}w\|$ for $C \geq 1$. The condition $a \geq \frac{32Y^2}{\pi}$ ensures that $\pi aC(a+TR^2) \geq 32Y^2(a+CTR^2)$. This concludes the proof. □

IV

CIRR: Competitive iterative ridge regression

I can observe the game theory is applied very much in economics. Generally, it would be wise to get into the mathematics as much as seems reasonable because the economists who use more mathematics are somehow more respected than those who use less. That's the trend.

—John Forbes Nash, Jr.

1- Derivation

For OSLOG a prior is assigned on weights and then a posterior predictive distribution is obtained, which is inline with the Bayesian Theory mentioned earlier. Herein, same problem as OSLOG is studied from game-theoretic point of view. Inspired by AAR algorithm CIRR algorithm with its performance guarantee is given. Also, an explicit relationship between CIRR and OSLOG is studied.

As mentioned earlier, at the beginning of each trial t the learner receives a signal $x_t \in \mathbb{R}^n$, which is processed by the decision pool $w \in \Theta$, and the prediction is denoted by $\hat{y}_t = w'x_t$. AAR relies on the aggregating pseudo algorithm (APA) prediction $-$ slices the weights of the decision pool by fixing the learning rate $\eta = \frac{1}{2Y^2}$ and by assigning *prior* probability distribution $P_0 \in \Theta$ to the weights of the decision pool or the experts. AA uses APA prediction $g : \Omega \rightarrow \mathbb{R}$ for mapping a given value of incurred loss at each trial t. The main role of APA is to reduce the weights of the strategies who suffer greater loss at previous trial i.e.:

$$P_T(M) = \int_M \beta^{(y_T - \hat{y}_t^w)^2} P_{T-1}(dw) \tag{62}$$

for all measurable $M \subseteq \Theta$, where $\beta = e^{-\frac{1}{2Y^2}}$. At each trial, APA chooses a prediction by the following rule:

$$g_T(y) = \log_\beta \frac{\int_\Theta \beta^{(y_T - \hat{y}_t^w)^2} P_{T-1}(dw)}{P_{T-1}(\Theta)} \tag{63}$$

and Lemma 1 in [113] shows that the loss of APA is as follows:

$$L_T(APA) = \log_\beta \int_\Theta \beta^{\sum_{t=1}^T (y_t - \hat{y}_t^w)^2} P_0(dw) \tag{64}$$

The loss of AAR is less than or equal to the loss of APA. So to obtain the loss of AAR, "$=$" in (64) is replaced by "$\leq$". The previous statement holds because the *substitution function* $\Sigma_{\frac{1}{2Y^2}}$ satisfies:

$$\forall \eta \forall y : (y - \Sigma_{\frac{1}{2Y^2}}(g))^2 \leq c(\eta) g(y) \tag{65}$$

for any pseudo-prediction g, here $c(\eta)$ with $\eta = \frac{1}{2Y^2}$ is the mixability curve defined as follows:

$$c(\eta) = \inf\{c \mid \forall g \; \exists \delta \in \hat{y}_t \; \forall y : (y - \delta)^2 \leq c g(y)\}$$

A computationally efficient approach to satisfy (65) (under mild conditions on the game mentioned in [112, 108]) require:

$$\Sigma_{\frac{1}{2Y^2}}(g) \in \arg \inf_{\hat{y}_t \in \hat{y}_t} \sup_{y \in \Omega} \left((y - \hat{y}_t)^2 - c \left(\frac{1}{2Y^2} \right) g(y) \right) \tag{66}$$

Since $g_1(y) - g_2(y)$ does not depend on y, thus:

$$\Sigma_{\frac{1}{2Y^2}} g_1(y) = \Sigma_{\frac{1}{2Y^2}} g_2(y) = ... \tag{67}$$

The advantage of using (67) is that when running AAR one does not need to *normalise* weights i.e. using (64) instead of (63). In [113] the following substitution function for the square loss game is obtained:

$$\hat{y}_t = \frac{1}{4Y} \log_\beta \frac{\beta^{g(-Y)}}{\beta^{g(Y)}} \tag{68}$$

where $\Sigma_{\frac{1}{2Y^2}}$ maps every prediction $g : \Omega \to [0, \infty]$. In [28], it is shown that AAR chooses $\hat{y}_t$ such that the following:

$$\sup_{y \in [-Y, Y]} \left(L_T(AAR) - \inf_{w \in \mathbb{R}^n} \left(a\|w\|_2^2 + \sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \right) \right) \tag{69}$$

is minimal. Next, by inspiring from the literature on AAR, protocol of CIRR is presented.

Remark 10. *The concept of probability space $(\Omega, \mathcal{F}, \mathbb{P})$ in the game-theoretic framework is not needed, instead, a game of triple $(\Omega, \Gamma, \lambda)$ that indicate respectively a set of possible outcomes, a set of allowed predictions, and a function measuring the loss is considered. In discrete-time game-theoretic framework, Ω is constructed from the actual outcomes arriving from the data stream, thus, it is not necessary to consider a probability measure on Ω, as done in the classical probability theory.*

Protocol 8 shows the framework under which CIRR work. In this protocol, notice that the learner does not know the label at the time of prediction, but it knows the moves made by the decision pool $w \in \mathbb{R}^n$ at each trial t and prediction, $w_t' x_t$, is computed. It is worth noting that CIRR strategy interacts with the decision pool twice. In contrast to AAR, the learner does not need to interact with the decision pool explicitly.

Protocol 8: CIRR strategy

```
FOR  $t = 1, 2, ...,$
     (1) Read $x_t \in \mathbb{R}^n$
     (2) Learner chooses $w \in \Theta$
     (3) Learner predicts $\hat{y}_t \in \mathbb{R}$
     (4) Read $y_t \in \mathbb{R}$
     (7) Update $w \in \Theta$
END FOR
```

Lemma 7. *For all $t \geq 0$, $f(w) := a\|D_{w_{t-1}}^{-\frac{1}{2}}w\|_2^2 + L_T$ is minimal at a unique point w and the function $f(w)$ is as follows:*

$$w = A_t^{-1}b_t \quad \textit{and} \quad f(w) = \sum_{s=1}^{t} y_s^2 - b_t' A_t^{-1} b_t$$

such that none of the elements of the weight vector has its absolute value at any step equal to zero.

Proof. By definition

$$f(w) = a\|D_{w_{t-1}}^{-\frac{1}{2}}w\|_2^2 + \sum_{s=1}^{t}(y_s - w'x_s)^2$$

$$= aw'D_{w_{t-1}}^{-1}w + \sum_{s=1}^{t}(y_s^2 - 2y_s w'x_s + w'(x_s \otimes x_s)w)$$

$$= \sum_{s=1}^{t} y_s^2 - 2w' \sum_{s=1}^{t} y_s x_s + w' \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s \otimes x_s \right) w$$

$$= \sum_{s=1}^{t} y_s^2 - 2w'b_t + w'A_t$$

$$f(w) = \sum_{s=1}^{t} y_s^2 - \left(\sum_{s=1}^{t} 2y_s w'x_s \right) + w' \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s \otimes x_s \right) w$$

Proceeding by differentiating with respect to w (treating w_{t-1} as a constant):

$$\nabla f(w) = 2\sum_{s=1}^{t} y_s x_s + 2w' \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s \otimes x_s \right) \implies$$

$$H\nabla f(w) = 2aD_{w_{t-1}}^{-1} + 2\sum_{s=1}^{t} x_s \otimes x_s$$

and Since $\nabla f(w) = 0 - 2b_t + 2A_t w$ and $H\nabla f(w) = 2A_t \Rightarrow f$ is convex, so to attain the minimal point, setting $\nabla f(w) = 0$ which gives $w = b_t' A_t^{-1}$. Thus,

$$f(w) = f(b_t' A_t^{-1}) = \sum_{s=1}^{t} y_s^2 - 2b_t' A_t^{-1} b_t + b_t' A_t^{-1} A_t A_t^{-1} b_t$$

$$= \sum_{s=1}^{t} y_s^2 - b_t' A_t^{-1} b_t$$

□

Theorem 9. *CIRR predicts $\hat{y}_t = b_{t-1}' A_t^{-1} x_t$.*

Proof. Considering following min-max problem:

$$\arg \inf_{\hat{y}_t \in \mathbb{R}} \sup_{y_t \in [-Y,Y]} \left(\sum_{s=1}^{t} (y_s - \hat{y}_s)^2 - \sum_{s=1}^{t} y_s^2 + b_t' A_t^{-1} b_t \right)$$

$$= \arg \inf_{\hat{y}_t \in \mathbb{R}} \sup_{y_t \in [-Y,Y]} \left(\sum_{s=1}^{t} (y_s - \hat{y}_s)^2 \right.$$

$$\left. - \sum_{s=1}^{t} y_s^2 + (b_{t-1} + y_t x_t)' A_t^{-1} (b_{t-1} + y_t x_t) \right)$$

$$= \arg \inf_{\hat{y}_t \in \mathbb{R}} \sup_{y_t \in [-Y,Y]} \left(\sum_{s=1}^{t} (y_s - \hat{y}_s)^2 - \sum_{s=1}^{t} y_s^2 \right. \tag{70}$$

$$\left. + b_{t-1} A_t^{-1} b_{t-1} + 2 y_t b_{t-1}' A_t^{-1} x_t + y_t^2 x_t' A_t^{-1} x_t \right)$$

$$\implies \arg \inf_{\hat{y}_t \in \mathbb{R}} \sup_{y_t \in [-Y,Y]} \left(-2 y_t \hat{y}_t + \hat{y}_t^2 + 2 y_t b_{t-1}' A_t^{-1} x_t + y_t^2 x_t' A_t^{-1} x_t \right)$$

$$= \arg \inf_{\hat{y}_t \in \mathbb{R}} \sup_{y_t \in [-Y,Y]} \left(2 y_t (b_{t-1}' A_t^{-1} x_t - \hat{y}_t) + y_t^2 (x_t' A_t^{-1} x_t) + \hat{y}_t^2 \right) \tag{71}$$

Given $y_t \in [-Y, Y]$ and that A_t is positive definite, asserts $\hat{y}_t$ should be chosen such that:

$$2Y \left(b_{t-1} A_t^{-1} x_t - \hat{y}_t \right) + \hat{y}_t^2 \tag{72}$$

(72) is minimised. Since:

- **Case 1**: $b_{t-1} A_t^{-1} x_t \in [-Y, Y]$. If $b_{t-1} A_t^{-1} x_t \geq Y$ than (72) is decreasing when $\hat{y}_t \leq Y$ and increasing when $\hat{y}_t \geq Y$, similar arguments holds for the case when $b_{t-1} A_t^{-1} x_t \geq -Y$, thus for (72) minimum is attained at Y.

- **Case 2**: $\hat{y}_t \leq b_{t-1} A_t^{-1} x_t$ attains minimum on the domain $\min(Y, b_{t-1} A_t^{-1} x_t)$.

- **Case 3**: $\hat{y}_t \geq b_{t-1} A_t^{-1} x_t$ attains minimum on the domain $\max(-Y, b_{t-1} A_t^{-1} x_t)$.

Thus, for $\hat{y}_t = b_{t-1} A_t^{-1} x_t$ (70) attains minimum. □

Protocol 9: CIRR

```
Initialise: $a > 0, A = \mathbf{0}^{n \times n}$, $b = \mathbf{0}^{n \times 1}$ and $w = \mathbf{1} \in \mathbb{R}^{n \times 1}$
    .
```

```
FOR   $t = 1, 2, ...,$
      (1) Read $x_t \in \mathbb{R}^n$
      (2) $D_{w_{t-1}} = \text{diag}(\sqrt{\texttt{abs}(w)})$
      (3) $A = A + x_t \otimes x_t$
      (4) $A^{-1} = D_{w_{t-1}} \left(a\mathbf{I} + D_{w_{t-1}} A D_{w_{t-1}}\right)^{-1} D_{w_{t-1}}$
      (5) $\hat{y}_t = b' A^{-1} x_t$
      (6) Read $y_t \in \mathbb{R}$
      (7) $b = b + y_t x_t$
      (8) $w = A^{-1} b$
END FOR
```

2- Analysis

The following corollary presents the limiting behaviour of the algorithm. It shows that as $\|x_t\|_p \to \infty$, $\hat{y}_t \to 0$, thus making the algorithm less likely to overestimate in comparison to the usual convex optimisation methods that predict by multiplying the optimal decision strategy from the decision pool by x_t.

Corollary 2. *For all $s = 1, 2, ..., t$, the following result holds:*

$$\hat{y}_t = \frac{s_t}{1 + x_t' D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s\right) D^{\frac{1}{2}}_{w_{t-2}}\right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t}$$

where $\hat{y}_t$ denotes the prediction of CIRR and:

$$s_t = \left(\sum_{s=1}^{t-1} y_s x_s\right)' D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s\right) D^{\frac{1}{2}}_{w_{t-2}}\right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t$$

where $D_{w_{t-2}} = \text{diag}(w_{t-2,1}, ..., w_{t-2,n})$

Proof.

$$\begin{aligned}
\hat{y}_t &= \left(\sum_{s=1}^{t-1} y_s x_s\right)' D^{\frac{1}{2}}_{w_{t-1}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-1}} \left(\sum_{s=1}^{t} x_s \otimes x_s\right) D^{\frac{1}{2}}_{w_{t-1}}\right)^{-1} D^{\frac{1}{2}}_{w_{t-1}} x_t \\
&= \left(\sum_{s=1}^{t-1} y_s x_s\right)' D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s\right) D^{\frac{1}{2}}_{w_{t-2}}\right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t \\
&\qquad - \left(\sum_{s=1}^{t-1} y_s x_s\right)' \times \\
&\qquad \frac{\left(D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s\right) D^{\frac{1}{2}}_{w_{t-2}}\right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t\right)}{1 + x_t' D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s\right) D^{\frac{1}{2}}_{w_{t-2}}\right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t}
\end{aligned}$$

$$\times \frac{\left(D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s \right) D^{\frac{1}{2}}_{w_{t-2}} \right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t \right)'}{1 + x_t' D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s \right) D^{\frac{1}{2}}_{w_{t-2}} \right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t} x_t$$

By some simple algebraic manipulation:

$$\hat{y}_t = \frac{s_t}{1 + x_t' D^{\frac{1}{2}}_{w_{t-2}} \left(a\mathbf{I} + D^{\frac{1}{2}}_{w_{t-2}} \left(\sum_{s=1}^{t-1} x_s \otimes x_s \right) D^{\frac{1}{2}}_{w_{t-2}} \right)^{-1} D^{\frac{1}{2}}_{w_{t-2}} x_t} \tag{73}$$

□

Next the upper bounds on the cumulative square loss for CIRR are discussed. The main objective is to deduce the circumstances under which CIRR has a better regret than AAR (has the best upper bound on the cumulative square loss in an online setting). To achieve this goal first a performance guarantee of CIRR is obtained. Then, the input and weights are bounded for the sake of simplicity of the comparison. Finally, the regret of CIRR and AAR is compared. The construction of the upper bound is presented in two ways, which lead to distinct results.

Theorem 10. *For any point in time $t = 1, 2, ..., T$, the following holds:*

$$L_T(CIRR) \leq \inf_{w \in \mathbb{R}^n} \left(L_T + a \| D^{-\frac{1}{2}}_{w_{t-1}} w \|_2^2 \right) + Y^2 \ln \det \left(\frac{1}{a} A_T \right) \tag{74}$$

where $a > 0$, $Y \geq 0$. if $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$ $\forall t$ such that $C \neq 0$, $|w_{t,i}| \neq 0$ $\forall i = 1, 2, ..., n$ and n is some finite positive integer then:

$$L_T(CIRR) \leq L_T^* + aP^2C^{-1} + Y^2 n \ln \left(\frac{a + CTR^2}{aC} \right) \tag{75}$$

Proof. is analogous to Theorem 5. □

The following theorem presents a scenario when the CIRR upper bound on cumulative loss is better than AAR (and ORR).

Theorem 11. *If, $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$ such that $C \geq 1$, $a > 0$, and $n \in \mathbb{N}^+$, then $\forall t$ the following holds*

$$L_T^U(CIRR) \leq L_T^U(AAR)$$

where L_T^U denotes the upper cumulative square loss bound

Proof. is analogous to Theorem 6 □

Theorem 12. *For any time step $t = 1, 2, ..., T$, any $a > 0$ and $|w_{t-1,1}|, ..., |w_{t-1,n}| \neq 0$, the following holds for $\eta = \frac{1}{2Y^2}$:*

$$L_T(CIRR) \leq \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right) + Y^2 \left(2n \ln \left(\frac{4Y^2}{a\sqrt{\pi}} \right) + \ln \det \frac{A_T}{2Y^2} \right)$$

Proof. For any value of $\eta > \frac{1}{2Y^2}$, $\beta^{(y_t - w' x_t)^2}$ will not be concave for $w' x_t$, for details please see [113] Remark 3. Thus, replacing $\eta = \frac{1}{8Y^2}$ by $\eta = \frac{1}{2Y^2}$ in the proof of Theorem 7 leads to the desired result. □

Corollary 3. *For any point in time $t = 1, 2, ..., T$ and any $a > 0$ such that $\|x_t\|_\infty \leq R$ and $C \leq \|w\|_1 \leq P$ following holds:*

$$L_T(CIRR) \leq L_T^* + aP^2 C^{-1} + nY^2 \ln \left(\frac{8Y^2(a + CTR^2)}{a^2 C \pi} \right)$$

such that $C \neq 0$.

Proof. From (61)

$$\begin{aligned}
L_T(CIRR) &\leq \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right) \\
&\quad + Y^2 \left(2n \ln \frac{4Y^2}{a\sqrt{\pi}} + n \ln \frac{a + CTR^2}{2Y^2 C} \right) \\
&= \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right) + Y^2 \left(n \ln \frac{16Y^4}{a^2 \pi} + n \ln \frac{a + CTR^2}{2Y^2 C} \right) \\
&= \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right) + Y^2 \left(n \ln \left(\frac{16Y^4(a + CTR^2)}{2a^2 \pi Y^2 C} \right) \right) \\
&= \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right) + Y^2 \left(n \ln \left(\frac{8Y^2(a + CTR^2)}{a^2 C \pi} \right) \right) \\
&= \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right) + nY^2 \ln \left(\frac{8Y^2(a + CTR^2)}{a^2 C \pi} \right)
\end{aligned}$$

The rest follows from (55). □

Notice the restriction $C \leq \|w\|_1 \leq P$ with $C \neq 0$ in Corollary 3 can be reduced to $\|w\|_1 \leq P$ by using Lemma 1 and Lemma 7 and following the similar procedure as Corollary 3. So $L_T(IRR) = \inf_w \left(L_T + a\|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2 \right)$ can be written as $\sum_{t=1}^T y_t^2 - b' A^{-1} b$ (please see Lemma 7), and (61) becomes $n \ln(aP + TP^2 R^2)$. Notice that the upper bound on the determinant of CIRR is $\ln \frac{16Y^4}{a^2 \pi}(P(a + TPR^2))$ in comparison to AAR's $\ln \frac{a + TR^2}{a}$. When $C \leq 1$, $\inf_w \left(L_T^* + a\|w\|_2^2 \right) = L_T(RR) \leq L_T(IRR)$, setting $a \geq \frac{16Y^4}{\pi}$ ensures that $\ln \frac{16Y^4(P(a+TPR^2))}{a\pi(a+TR^2)} \leq 0 \implies \ln \frac{16Y^4}{a^2 \pi}(P(a + TPR^2)) \leq \ln \frac{a+TR^2}{a}$. Nevertheless, this way of analysis does not provide a clean comparison of AAR and CIRR.

However, it indicates that CIRR has a better bound when $\|w\|_1 \leq 1$ and the noise term has a greater influence on the prediction accuracy than the true regression function.

The following theorem presents circumstances under which the regret of CIRR is better than AAR's.

Theorem 13. *Let $\mathcal{R}_T = L_T(Learner) - L_T^*$, $\|x_t\|_\infty \leq R$, $C \leq \|w\|_1 \leq P$ and n be some positive integer, then $\forall t$, $\mathcal{R}_T(AAIR) \leq \mathcal{R}_T(AAR)$ when $C \geq 1$, $a \geq \frac{8Y^2}{\pi}$.*

Proof. To prove, it is sufficient to show $\mathcal{R}_T(AAIR) - \mathcal{R}_T(AAR) \leq 0$. From (60) and Corollary 3:

$$aP^2\left(\frac{1}{C} - 1\right) + nY^2 \ln\left(\frac{8Y^2(a + CTR^2)}{a^2 C\pi}\right) - nY^2 \ln\left(\frac{a + TR^2}{a}\right) \leq 0$$

$$aP^2\left(\frac{1}{C} - 1\right) + nY^2 \ln\frac{8Y^2(a + CTR^2)}{aC\pi(a + TR^2)} \leq 0$$

$C \geq 1$, $aP^2\left(\frac{1}{C} - 1\right) \leq 0$. Also, from Lemma 4 it is clear that $\|w\|_2^2 \geq \|D_{w_{t-1}}^{-\frac{1}{2}} w\|_2^2$ for $C \geq 1$. The condition $a \geq \frac{8Y^2}{\pi}$ ensures that $\pi a C(a + TR^2) \geq 8Y^2(a + CTR^2)$. This concludes the proof. □

Remark 11. *CIRR guarantee is better than OSLOG under all circumstances.*

V

CNLS: Competitive normalised least squares

The ordinary operations of algebra suffice to resolve problems in the theory of curves.

—Joseph-Louis Lagrange

1- Derivation

CNLS is an online regression algorithm with time complexity $\mathcal{O}(n)$. The algorithm considers (41) with $D_{w_{t-1}} = \mathbf{I}$. First, bounds are given without using Definition 6 at all, later bounds are given by only bounding the signals (one could easily obtain bounds by bounding weights and signals both - the given bounds are under milder conditions). Herein, for compactness, representing the inner product space for all $u, v, x \in \mathbb{R}^n$ and $l \in \mathbb{R}$ such that: $(u, v) = (v, u)$,$(lu, v) = l(u, v)$, $(u + v, x) = (u, x) + (v, x)$, $(x, x) > 0$ for $x \neq 0$, for further details see [121]. CNLS obtains an estimate $\hat{w}$ for the w to make predictions at each trial, by observing the following protocol:

Protocol 10: CNLS strategy

```
FOR t=1,2,...,T
   (1) receive  $x_t \in \mathbb{R}^n$
   (2) predict  $\hat{y}_t = (\hat{w}_t, x_t)$
   (3) receive  $y_t \in \mathbb{R}$
   (4) update   $\hat{w}_t \in \mathbb{R}^n$
END FOR
```

In Protocol 10, it is assumed that the prediction is given by $\hat{w}_t' x_t$. So, the problem in hand is to design the update rule, which leads us to the following lemma.

Lemma 8. *The following minimisation problem with respect to $\hat{w}_t$:*

$$min \left(\sum_{t=1}^{T} (\hat{w}_t - \hat{w}_{t-1})^2 \right)$$

with the constraint $y_t = w_t' x_t$ has the following solution for $t = 1, 2, ..., T$:

$$\hat{w}_t = \hat{w}_{t-1} + \frac{(y_t - \hat{y}_t) x_t}{\|x_t\|_2^2}$$

Proof. To minimise $\sum_{t=1}^{T} (\hat{w}_t - \hat{w}_{t-1})^2$ under the constraint $y_t - \hat{w}_t' x_t = 0$. Introducing Lagrangian multipliers α_t, $t = 1, 2, ..., T$. Instead of solving the primal optimisation problem mentioned earlier and find the saddle point of the following:

$$\sum_{t=1}^{T} (\hat{w}_t - \hat{w}_{t-1})^2 + \sum_{t=1}^{T} \alpha_t (y_t - \hat{w}_t' x_t) \tag{76}$$

In accordance with the Kuhn-Tucker theorem [62], there exists values of Lagrangian multipliers $\alpha = \alpha^{KT}$ for which solving the primal problem is equivalent to finding the saddle point. Thus:

$$\frac{\partial}{\partial \hat{w}_t} \left(\sum_{t=1}^{T} (\hat{w}_t - \hat{w}_{t-1})^2 + \sum_{t=1}^{T} \alpha_t (y_t - \hat{w}_t' x_t) \right) = 0 \tag{77}$$

For, $t = 1, 2, ..., T$

$$\begin{aligned} \frac{\partial}{\partial \hat{w}_t} \left((\hat{w}_t - \hat{w}_{t-1})^2 + \alpha_t (y_t - \hat{w}_t' x_t) \right) &= 0 \\ 2(\hat{w}_t - \hat{w}_{t-1}) - \alpha_t x_t &= 0 \\ \hat{w}_t = \hat{w}_{t-1} + \frac{1}{2} \alpha_t x_t & \qquad (78) \end{aligned}$$

Substituting the obtained value of $\hat{w}_t$ from (78) in the constraint and we get:

$$\begin{aligned} y_t &= (\hat{w}_{t-1} + \frac{1}{2} \alpha_t x_t)' x_t \\ (y_t - \hat{y}_t) &= \frac{1}{2} \alpha_t \|x_t\|_2^2 \\ \alpha_t &= \frac{2}{\|x_t\|_2^2} (y_t - \hat{y}_t) \qquad (79) \end{aligned}$$

Substitution of α_t from (79) in (78) gives:

$$\hat{w}_t = \hat{w}_{t-1} + \frac{(y_t - \hat{y}_t) x_t}{\|x_t\|_2^2} \tag{80}$$

In order to avoid the scenario $\hat{w}_t \to \infty$ as $\|x_t\|_2^2 \to 0$, using the convention $\frac{0}{0} = 0$. □

Remark 12. *The analysis of the following update rule:*

$$\hat{w}_t = \hat{w}_{t-1} + \frac{(y_t - \hat{y}_t) x_t}{\eta + \|x_t\|_2^2} \tag{81}$$

for $\eta > -\|x_t\|_2^2$. The obvious advantage of using (81) is that one do not require any convention for the case when $\|x_t\|_2^2 \to 0$. Later it is shown that the addition of η in the denominator results in a better performance guarantee.

CNLS protocol is presented in Protocol 11, where the weight vector is initially set to $\mathbf{0} \in \mathbb{R}^n$. The update rule can be written in the form: $\hat{w}_t = \hat{w}_{t-1} + \lambda x_t$, where $\lambda = \frac{y_t - \hat{y}_t}{\eta + \|x_t\|_2^2} \in \mathbb{R}$.

Protocol 11: CNLS

```
FOR t=1,2,...
  (1) receive $x_t \in \mathbb{R}^n$
```

```
    (2) predict $\hat{y}_t = (\hat{w}_t, x_t)$
    (3) receive $y_t \in \mathbb{R}$
    (4) update $\hat{w}_t$ using eq.(71)
END FOR
```

2- Analysis

Now, analysing CNLS using the difference of sum of squares analysis as suggested by Duda et al. [26]. First determining the amount of learning Algorithm 11 does from the error at each trial. The following Lemmas are a step in this direction.

Lemma 9. *Let $\hat{y}_t = (\hat{w}_{t-1}, x_t)$, $\hat{w}_t = \hat{w}_{t-1} + \lambda x_t$, where $x_t, \hat{w}_{t-1}, w \in \mathbb{R}^n, y_t \in \mathbb{R}$ and $\lambda = \frac{(y_t - \hat{y}_t)}{\eta + \|x_t\|^2}$, the following holds:*

$$\|\hat{w}_{t-1} - w\|^2 - \|\hat{w}_t - w\|^2 = (y_t - \hat{y}_t)^2 \left(\frac{2}{\eta + \|x_t\|^2} - \frac{\|x_t\|^2}{\eta + \|x_t\|^2} \right) - \frac{2(y_t - \hat{y}_t)(y_t - (w, x_t))}{\|x_t\|^2 + \eta} \tag{82}$$

Proof.

$$\|\hat{w}_t - w\|^2 - \|\hat{w}_{t-1} - w\|^2 = 2\lambda(x_t, (\hat{w}_{t-1} - w)) + \lambda^2 \|x_t\|^2$$

$$= 2\lambda(\hat{y}_t - y_t) + 2\lambda(y_t - (w, x_t)) + \lambda^2 \|x_t\|^2$$

Substitution of $\lambda = \frac{(y_t - \hat{y}_t)}{\eta + \|x_t\|^2}$ leads to the desired result □

Lemma 10. *For all $a, b, r, \beta \in \mathbb{R}$ such that $0 < \beta < 1$*

$$a^2 - ab \geq \beta(ab)^2 - \frac{b^2}{4(1-\beta)}$$

Proof. The inequality is equivalent to the following

$$a^2 - ab - \beta(ab)^2 + \frac{b^2}{4(1-\beta)} \geq 0$$

$$\frac{4a^2 - 8a^2\beta + 4a^2\beta^2 + b^2 - 4ab(1-\beta)}{4(1-\beta)} \geq 0$$

Clearly, left hand side can be written as $\frac{((2a - 2a\beta) - b)^2}{4(1-\beta)}$for $0 < \beta < 1$, thus the inequality holds. □

Next, proving a lower bound on Lemma 9 using inequality proven in Lemma 10. This can be interpreted as the lower bound on the progress per trial in Protocol 11.

Lemma 11. *Let $\hat{y}_t = (\hat{w}_{t-1}, x), \hat{w}_t = \hat{w}_{t-1} + \frac{y_t - \hat{y}_t}{\eta + \|x_t\|_2^2} x_t$, where $x_t, \hat{w}_t, w \in \mathbb{R}^n, y_t \in \mathbb{R}$, the following holds for $0 < \beta < 1$:*

$$\|\hat{w}_{t-1} - w\|_2^2 - \|\hat{w}_t - w\|_2^2 \geq \frac{\beta(y_t - \hat{y}_t)^2}{(\eta + \|x_t\|_2)^2} - \frac{(y_t - (w, x))^2}{(1-\beta)(\eta + \|x_t\|_2^2)}$$

Proof. Using Lemma 9 and 10 leads to:

$$\|\hat{w}_{t-1} - w\|^2 - \|\hat{w}_t - w\|^2 = (y_t - \hat{y}_t)^2 \left(\frac{2}{\eta + \|x_t\|^2} - \frac{\|x_t\|^2}{\eta + \|x_t\|^2}\right) - \frac{2(y_t - \hat{y}_t)(y_t - w' x_t)}{\|x_t\|^2 + \eta} \geq \left(\frac{2}{\eta + \|x_t\|^2} - \frac{1}{(\eta + \|x_t\|^2)^2}\right)(y_t - \hat{y}_t)^2 - \frac{2(y_t - \hat{y}_t)(y_t - w'x)}{\eta + \|x_t\|^2} \geq \frac{1}{\eta + \|x_t\|^2} \left(\beta(y_t - \hat{y}_t)^2 - \frac{(y_t - w'x)^2}{1-\beta}\right)$$

□

Theorem 14. *For any sequence $x_1, y_1, x_2, y_2, \ldots$ with predictions $\hat{y}_1, \hat{y}_2, \ldots$, given by Algorithm 11, the following holds:*

$$L_T \leq \frac{1}{\beta(1-\beta)} \inf_w \left((\eta + R^2)\|w\|_2^2 + L_T\right) + \mathcal{O}(1)$$

for $\|x_t\|_2 \geq R$. For $\eta = bR^2$ and $\beta = \frac{1}{2}$ the following holds:

$$L_T \leq 4 \inf_w \left((b+1)R^2\|w\|^2 + L_T\right) + \mathcal{O}(1)$$

where $b > -1$, L_T is the cumulative square loss and L_T is the cumulative square loss of the offline LS algorithm.

Proof. The left hand side of Lemma 4:

$$\sum_{t=1}^{T} \left(\|\hat{w}_t - w\|_2^2 - \|\hat{w}_{t+1} - w\|_2^2\right) = \|\hat{w}_1 - w\|_2^2 - \|\hat{w}_{T+1} - w\|_2^2 \leq \|w\|_2^2$$

since, initialisation of the weights is $\mathbf{0}$ and $\|\cdot\|$ is non-negative. So, the inequality can be written as follows:

$$\beta \sum_{t=1}^{T} \frac{(\hat{y}_t - y_t)^2}{\|x_t\|_2^2} \frac{\|x_t\|_2^2}{\eta + \|x_t\|_2^2} - \sum_{t=1}^{T} \frac{(y_t - (w, x_t))^2}{\|x_t\|_2^2} \frac{\|x_t\|_2^2}{(1-\beta)(\eta + \|x_t\|_2^2)} \leq \|w\|_2^2 \quad (83)$$

Setting $\|x_t\|_2 \geq R$ to get L_T:

$$\frac{\beta}{(\eta + R^2)} L_T - \frac{L_T}{(1-\beta)(\eta + R^2)} \leq \|w\|_2^2$$

Thus,

$$L_T \leq \frac{1}{\beta(1-\beta)} \left((\eta + R^2)\|w\|_2^2 + L_T\right)$$

□

The result obtained in Theorem 14 fulfils Definition (3) with $c = \frac{1}{\beta(1-\beta)}$, $L_T^* = \inf_w \left((\eta + R^2)\|w\|_2^2 + L_T\right)$ and $R_T = \mathcal{O}(1)$. Theorem 14[1] asserts for $\eta = 0$ and $\beta = \frac{1}{2}$ the following holds:

$$L_T \leq 4 \inf_w \left(R^2 \|w\|_2^2 + L_T\right) \tag{84}$$

Clearly, the addition of η in the update rule of Lemma 8 is advantageous. In Theorem 14, the addition of η decreases the dependence on the size of the data. It is worth noticing that (84) is the performance guarantee for the algorithm derived in Lemma 8. Here, the only assumption is that the input is bounded by the Euclidean norm. Also, notice when $\beta = \frac{1}{2}$ and as $b \to -1 \implies \inf_w \left((b+1)R^2\|w\|_2^2 + L_T\right) \to L_T \leq 4L_T^*$. That is, CNLS is at most 4 times worse than the true regression function.

The following theorem presents the performance guarantee on the normalised squared loss.

Theorem 15. *For any sequence $x_1, y_1, x_2, y_2, ...$ with $\eta = b\|x_t\|_2^2$ and predictions $\hat{y}_1, \hat{y}_2, ...$, given by Algorithm 11, the following holds:*

$$\bar{L}_T \leq \frac{1}{\beta(1-\beta)} \inf_{w \in \mathbb{R}^n} \left((b+1)(1-\beta)\|w\|_2^2 + \bar{L}_T\right) + \mathcal{O}(1)$$

such that $b > -1$, $0 < \beta < 1$, $\bar{L}_T$ is the normalised squared loss and $\bar{L}_T^*$ is the normalised squared loss of the offline algorithm.

Proof. Notice that:

$$\beta \sum_{t=1}^{T} \frac{(\hat{y}_t - y_t)^2}{\|x_t\|_2^2} \frac{\|x_t\|_2^2}{\eta + \|x_t\|_2^2} = \frac{\beta}{(b+1)} \sum_{t=1}^{T} \frac{(\hat{y}_t - y_t)^2}{\|x_t\|_2^2}$$

and

$$\sum_{t=1}^{T} \frac{(y_t - (w, x_t))^2}{\|x_t\|_2^2} \frac{\|x_t\|_2^2}{(1-\beta)(\eta + \|x_t\|_2^2)}$$
$$= \frac{1}{(1-\beta)(b+1)} \sum_{t=1}^{T} \frac{(y_t - (w, x_t))^2}{\|x_t\|_2^2}$$

[1] The guarantee is expressed as a function of $\inf_w \sum_t (y_t - (w, x_t))^2$. The infimum is taken over all w.

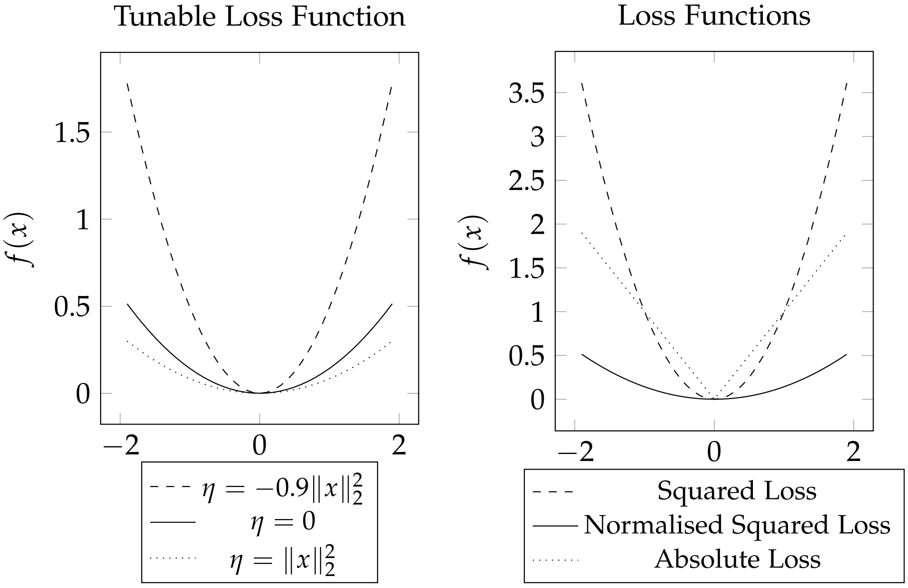

Figure 2: Tunable loss function (see Theorem 16)

.

for $\eta = b\|x_t\|_2^2$. Thus for (83) the following holds:

$$\frac{\beta}{(b+1)} \sum_{t=1}^{T} \frac{(\hat{y}_t - y_t)^2}{\|x_t\|_2^2} - \frac{1}{(1-\beta)(b+1)} \sum_{t=1}^{T} \frac{(y_t - (w, x_t))^2}{\|x_t\|_2^2} \leq \|w\|_2^2$$

and the result follows. □

In Theorem 15, the guarantee does not depend on the the size of the input and does not bound the input. Also, the performance guarantee has no assumptions on the input, the output and the weights.

The result of Theorem 15 fulfils Definition (3) with $c = 4$ (when $\beta = \frac{1}{2}$) instead of $c = 2.25$ as mentioned in [20], for normalised gradient descent (NGD) (Theorem $IV.2$). However, for NGD $L_T^* = \inf_{w \in \mathbb{R}^n} \|w\|^2 + \bar{L}_T$ instead of $\inf_{w \in \mathbb{R}^n} a\|w\|_2^2 + \bar{L}_T$ with $a > 0$.

Corollary 4. *As $\|w\|_2^2 \to \infty$, Algorithm 11 has a better guarantee than NGD for $\bar{L}_T$ at any given trial $T = 1, 2, ...$ if $0 < a \leq 0.5625$.*

Proof. By solving $4a\|w\|_2^2 + 4\bar{L}_T \leq 2.25\|w\|_2^2 + 2.25\bar{L}_T$ obtaining $0 < a \leq 0.5625 - 0.4375\frac{\bar{L}_T}{\|w\|_2^2}$. So, as $\|w\|_2^2 \to \infty$, $\frac{\bar{L}_T}{\|w\|_2^2} \to 0 \implies 0 < a \leq 0.5625$. □

Remark 13. *Similarly, for the case of L_T, NGD is outperformed as $\|w\|_2^2 \to \infty$ and $\|x_t\|_2 \geq R$ if $-1 < b \leq 0.5625R^2$.*

The guarantee that includes the learning rate η in the cumulative loss function is referred to as *tunable* loss function, from here onwards.

Theorem 16. *For any sequence $x_1, y_1, x_2, y_2, ...$ with predictions $\hat{y}_1, \hat{y}_2, ...,$ given by Algorithm 11, the following holds:*

$$\hat{L}_T \leq \inf_{w \in \mathbb{R}^n} \left(2\|w\|_2^2 + 4\hat{L}_T\right) + \mathcal{O}(1)$$

such that

$$\hat{L}_T = \sum_{t=1}^{T} \frac{(y_t - \hat{y}_t)^2}{\eta + \|x_t\|_2^2} \text{ and } \hat{L}^{=}_{T} \sum_{t=1}^{T} \frac{(y_t - (w_t, x_t))^2}{\eta + \|x_t\|_2^2}$$

with $\eta > -\|x_t\|^2$ and $0 < \beta < 1$.

Proof. Writing (83) as:

$$\hat{L}_T \leq \inf_{w \in \mathbb{R}^n} \frac{1}{\beta}\|w\|_2^2 + \frac{1}{\beta(1-\beta)}\hat{L}_T$$

By setting $\beta = \frac{1}{2}$, obtaining the desired result. □

Figure 2 compares some of the renowned loss functions and the behaviour of the loss function studied in Theorem 16. Notice that when the learning rate $\eta = 0$, the tunable loss is the same as the normalised squared loss. When $\eta = -0.9\|x\|_2^2$, the tunable loss penalty is in similar range as the absolute loss, but with the shape of the squared loss. Also, the tunable square loss is differentiable for all values of $\eta > -\|x\|_2^2$, at every value of x. The same statement does not hold for the absolute loss. So, the suggested tunable loss function has the robustness of the absolute loss, while in the shape of the squared loss.

VI

Empirical Study

It is exceptional that one should be able to acquire the understanding of a process without having previously acquired a deep familiarity with running it, with using it, before one has assimilated it in an instinctive and empirical way. Thus any discussion of the nature of intellectual effort in any field is difficult, unless it presupposes an easy, routine familiarity with that field. In mathematics this limitation becomes very severe.

—John von Neumann

1- data description

The data used in the experiments studies the effect of outlier(s) and or noise (Gaze and NO_2) and size ($F-16$ and weather), please see Table 1 and the following are the data descriptions:

- The Istanbul stock exchange (ISE) datum [2] - 536 observations with 8 attributes that are: S&P 500 Index, Deutscher Aktien Index, FTSE 100 Index, Nikkel Index, Bovespa Index, Bovespa Index, MSCI Europe Index and MSCU Emerging Markets Index. This datum is chosen due to its simplicity. There is no noise or outlier(s).

- Gaze datum [84] consists of 450 observations of 12 features related to measurements obtained from head-mounted cameras for eye tracking, estimating the positions of the eyes of the subject when the subject is looking at the monitor. This datum is chosen due to the presence of outlier(s).

- The NO_2 datum [107] consists of 500 observations from a road air pollution study collected by the Norwegian Public Roads Administration, measured at Alnabru in Oslo, Norway, between October 2001 and August 2003. There are 7 predictor variables: the logarithm of the number of cars per hour, temperature ($\times 2$), wind speed and direction, hour of the day and the date when the observations were taken.

- Ailerons ($F-16$) datum [106] consists of 13750 observations with a total of 40 attributes that describe the status of the $F-16$. This datum is chosen due to its size, has the highest number of features and illustrates algorithms shrinkage ability.

Table 1: Cook distance, mean & variance

datum	max.cook.dist	min.cook.dist	med.cooks.dist	label mean	label variance	lr.model variance
Gaze	1.90×10^{-1}	1.35×10^{-8}	7.18×10^{-4}	5.44×10^{2}	6.31×10^{4}	3.29×10^{3}
ISE	1.37×10^{-1}	7.28×10^{-10}	4.23×10^{-4}	1.55×10^{-3}	4.46×10^{-4}	3.23×10^{-5}
NO_2	4.25×10^{-2}	3.11×10^{-8}	7.52×10^{-4}	2.18×10^{-6}	1.00×10^{0}	4.98×10^{-1}
$F-16$	5.10×10^{-2}	1.50×10^{-6}	2.30×10^{-5}	-8.68×10^{-4}	1.69×10^{-7}	3.01×10^{-8}
Weather	9.18×10^{-6}	1.61×10^{-15}	9.83×10^{-4}	1.09×10	1.14×10^{2}	1.15×10^{0}

- Weather datum [16] has historical weather around Szeged, Hungary, from 2006 to 2016 with 9 features namely: temperature, apparent temperature, humidity, wind speed, wind bearing, visibility, cloud cover, precipitation type and summary. In total there are 96453 observations. This datum is chosen due to its length. This datum has the most number of observations among all data.

.

2- Experimental setting

The experiments are performed to illustrate the usefulness of the proposed algorithms by a comparative study against: RLS, AROWR, AAR, ORR, ONS, NGD and the optimal offline solution. To elaborate further:

- For all algorithms setting tuning parameter or the learning rate as $\frac{1}{T}$, where T denotes the length of the datum. So, it is assumed the length of the datum is known in advance.

- The naive baseline (using last step label as next step prediction) is also reported.

- The optimal solution using the entire datum is $\mathbf{X}w^*$, where $\mathbf{X} \in \mathbb{R}^{T \times n}$. This something all algorithms are trying achieve ideally. Also, it has direct link with the theoretical results. The bounds given are compared against $L_T^* = \inf_w \|\mathbf{Y} - \mathbf{X}w\|_2^2$, which is the optimal loss considered and $w^* = argmin_w \|\mathbf{Y} - \mathbf{X}w\|_2^2$. Meaning the baseline uses the optimal weights, where the optimal loss is achieved. Notice, when predicting one does not have access to entire datum.

3- Results

Table 2 reports the root mean square error (RMSE), coefficient of determination (R^2), mean absolute error (MAE) and error quantiles: lower quantile error (LQE (25%)), mean quantile error (MQE (50%)) and upper quantile error (UQE (75%)). Following are the central outcomes of the study:

- CIRR is overall the best algorithm in terms of $RMSE$, R^2 and MAE among all the algorithms in this experimental setting.

ONS, AROWR and RLS (AROWR and RLS fail to give a result on the weather datum) do not perform well in this experimental setting.

- CNLS is better on the small datum with no noise and outlier(s), please see Table 2 ISE results.
- None of the algorithm is able to outperform Xw^* on any of the datum. However, on weather datum CIRR is very close to the baseline in terms of RMSE and MAE. In terms R^2 CIRR and the optimal solution are the same. CIRR beats the naive baseline on all data.

Remark 14. *The tuning of the regularisation parameter is expected to have an impact on the performance of the algorithms. Optimisation of each algorithm on each datum is a different research direction and such out of the scope of this work.*

Table 2: Algorithms accuracy comparison on real-world datum

Algorithm	**RMSE**	**R^2**	**MAE**	**LQE**	**MQE**	**UQE**
datum: Gaze						
NGD	4.39×10^{2}	4.21×10^{-3}	3.66×10^{2}	1.42×10^{2}	3.46×10^{2}	5.60×10^{2}
CNLS	2.71×10^{2}	1.04×10^{-1}	2.19×10^{2}	-1.83×10^{2}	1.47×10	2.05×10^{2}
AROWR	4.88×10^{14}	5.91×10^{-5}	3.21×10^{13}	-3.31×10^{12}	-1.20×10^{12}	-3.69×10^{11}
RLS	2.19×10^{17}	1.19×10^{-4}	1.35×10^{16}	-6.26×10^{14}	-6.23×10^{12}	-7.61×10^{11}
ORR	2.19×10^{17}	1.19×10^{-4}	1.35×10^{16}	-6.26×10^{14}	-1.55×10^{10}	-1.77×10^{9}
AAR	1.48×10^{5}	7.63×10^{-3}	1.26×10^{5}	-1.84×10^{4}	-1.26×10^{5}	-6.23×10^{4}
ONS	5.33×10^{3}	9.91×10^{-4}	1.06×10^{3}	-5.52×10^{2}	-5.84×10	6.69×10^{2}
CIRR	1.61×10^{2}	6.65×10^{-1}	1.03×10^{2}	-2.04×10	4.37×10	1.13×10^{2}
OSLOG	2.35×10^{3}	1.37×10^{-3}	2.17×10^{2}	-6.01×10	-3.27×10^{0}	5.40×10
Naive	3.66×10^{2}	3.44×10^{-3}	2.99×10^{2}	-2.70×10^{2}	1.95×10	2.73×10^{2}
$\mathbf{X}w^*$	$\mathbf{5.65 \times 10}$	$\mathbf{9.49 \times 10^{-1}}$	$\mathbf{4.48 \times 10}$	$\mathbf{-3.94 \times 10}$	$\mathbf{-2.25 \times 10^{0}}$	$\mathbf{3.51 \times 10^{0}}$
datum: F$-$**16**						
NGD	9.41×10^{-4}	2.70×10^{-3}	8.45×10^{-4}	-1.04×10^{-3}	7.48×10^{-4}	-5.50×10^{-4}
CNLS	5.90×10^{-4}	2.02×10^{-1}	3.40×10^{-4}	-1.98×10^{-4}	7.28×10^{-6}	2.02×10^{-4}
AROWR	1.29×10^{11}	1.22×10^{-4}	1.15×10^{10}	-1.22×10^{8}	1.21×10^{7}	4.66×10^{8}
RLS	1.25×10^{11}	2.70×10^{-4}	1.10×10^{10}	-1.37×10^{8}	1.44×10^{7}	5.09×10^{8}
ORR	1.75×10^{7}	2.83×10^{-4}	1.60×10^{6}	-2.30×10^{4}	3.17×10^{3}	8.50×10^{4}
AAR	4.62×10^{-1}	1.64×10^{-4}	1.41×10^{-1}	-4.70×10^{-2}	8.49×10^{-4}	4.84×10^{-2}
ONS	2.30×10^{4}	1.11×10^{-2}	1.79×10^{4}	-1.23×10^{4}	1.29×10^{3}	1.72×10^{4}
CIRR	2.08×10^{-4}	7.82×10^{-1}	1.51×10^{-4}	-7.32×10^{-5}	4.21×10^{-5}	1.39×10^{-4}
OSLOG	6.03×10^{-1}	1.77×10^{-6}	7.28×10^{-3}	-6.37×10^{-5}	5.02×10^{-5}	1.43×10^{-4}
Naive	2.75×10^{-4}	6.05×10^{-1}	2.09×10^{-3}	-1.00×10^{-4}	-1.00×10^{-4}	-1.00×10^{-4}
$\mathbf{X}w^*$	$\mathbf{1.73 \times 10^{-4}}$	$\mathbf{8.24 \times 10^{-1}}$	$\mathbf{1.27 \times 10^{-4}}$	$\mathbf{-9.15 \times 10^{-5}}$	$\mathbf{3.36 \times 10^{-6}}$	$\mathbf{9.98 \times 10^{-5}}$
datum: NO$_2$						
NGD	9.65×10^{-1}	3.06×10^{-1}	7.63×10^{-1}	-6.27×10^{-1}	1.87×10^{-1}	6.55×10^{-1}
CNLS	8.86×10^{-1}	3.90×10^{-1}	6.62×10^{-1}	-0.49×10^{-1}	-1.04×10^{-2}	4.97×10^{-1}
AROWR	3.11×10^{5}	1.09×10^{-1}	1.40×10^{5}	-5.02×10^{4}	-4.29×10^{3}	3.81×10^{4}
RLS	3.15×10^{5}	1.14×10^{-1}	1.46×10^{5}	-5.90×10^{4}	-5.63×10^{3}	4.27×10^{4}
ORR	8.90×10^{2}	1.59×10^{-1}	4.78×10^{2}	-2.38×10^{2}	-2.51×10	1.69×10^{2}
AAR	4.35×10	1.95×10^{-1}	3.24×10	-3.16×10	5.71×10^{0}	1.37×10
ONS	8.25×10^{-1}	4.04×10^{-1}	6.23×10^{-1}	-4.78×10^{-1}	2.07×10^{-2}	5.11×10^{-1}
CIRR	7.31×10^{-1}	4.69×10^{-1}	5.72×10^{-1}	-3.56×10^{-1}	1.48×10^{-1}	5.58×10^{-1}
OSLOG	7.98×10^{-1}	3.98×10^{-1}	5.90×10^{-1}	-3.50×10^{-1}	1.25×10^{-1}	5.58×10^{-1}
Naive	1.09×10^{0}	1.58×10^{-1}	8.19×10^{-1}	-6.04×10^{-1}	-2.74×10^{-2}	5.99×10^{-1}
$\mathbf{X}w^*$	$\mathbf{7.01 \times 10^{-1}}$	$\mathbf{5.07 \times 10^{-1}}$	$\mathbf{5.47 \times 10^{-1}}$	$\mathbf{-4.13 \times 10^{-1}}$	$\mathbf{3.65 \times 10^{-2}}$	$\mathbf{4.62 \times 10^{-1}}$
datum: ISE						
NGD	1.89×10^{-2}	5.58×10^{-1}	1.40×10^{-2}	-8.84×10^{-3}	1.67×10^{-3}	1.11×10^{-2}
CNLS	7.12×10^{-3}	8.87×10^{-1}	4.87×10^{-3}	-4.06×10^{-3}	2.77×10^{-4}	3.52×10^{-3}
AROWR	1.80×10^{-2}	3.00×10^{-1}	1.30×10^{-2}	-8.62×10^{-3}	9.20×10^{-4}	1.01×10^{-2}
RLS	1.01×10^{-1}	5.94×10^{-1}	7.17×10^{-2}	-5.72×10^{-2}	-1.42×10^{-2}	1.28×10^{-2}
ORR	2.79×10^{-2}	4.85×10^{-1}	1.98×10^{-2}	-1.58×10^{-2}	-4.04×10^{-4}	1.23×10^{-2}
AAR	2.00×10^{-2}	3.77×10^{-1}	1.48×10^{-2}	-1.19×10^{-3}	2.04×10^{-3}	1.22×10^{-2}
ONS	2.08×10^{-2}	5.50×10^{-1}	1.56×10^{-2}	-9.54×10^{-3}	2.57×10^{-3}	1.34×10^{-2}
CIRR	7.61×10^{-3}	8.77×10^{-1}	5.07×10^{-3}	-4.25×10^{-3}	-1.47×10^{-4}	3.21×10^{-3}
OSLOG	7.82×10^{-3}	8.64×10^{-1}	5.02×10^{-3}	-4.18×10^{-3}	1.10×10^{-4}	3.13×10^{-3}
Naive	2.87×10^{-2}	5.22×10^{-3}	2.14×10^{-2}	-1.77×10^{-2}	-1.38×10^{-3}	1.61×10^{-2}
$\mathbf{X}w^*$	$\mathbf{5.64 \times 10^{-3}}$	$\mathbf{9.29 \times 10^{-1}}$	$\mathbf{4.30 \times 10^{-3}}$	$\mathbf{-3.351 \times 10^{-3}}$	$\mathbf{3.02 \times 10^{-4}}$	$\mathbf{3.24 \times 10^{-3}}$
datum: Weather						
NGD	1.29×10^{1}	1.19×10^{-3}	1.06×10^{1}	-1.68×10^{-0}	7.53×10^{0}	1.51×10^{-1}
CNLS	1.96×10^{0}	9.67×10^{-1}	7.23×10^{-1}	-3.83×10^{-1}	4.67×10^{-3}	3.68×10^{-1}
AROWR	–	–	–	–	–	–
RLS	–	–	–	–	–	–
ORR	5.38×10^{15}	1.55×10^{-5}	1.34×10^{14}	-9.16×10^{10}	-3.60×10^{8}	4.46×10^{9}
AAR	3.90×10^{7}	3.16×10^{-4}	1.53×10^{6}	-7.72×10^{5}	5.06×10^{5}	-2.56×10^{5}
ONS	5.73×10^{5}	5.17×10^{-1}	5.51×10^{5}	-6.63×10^{5}	-5.58×10^{5}	4.50×10^{5}
CIRR	1.09×10^{0}	9.89×10^{-1}	8.49×10^{-1}	-7.33×10^{-1}	-1.13×10^{-1}	6.56×10^{-1}
OSLOG	4.38×10^{0}	8.53×10^{-1}	8.58×10^{-1}	-7.39×10^{-1}	-1.24×10^{-1}	6.45×10^{-1}
Naive	1.81×10^{0}	9.71×10^{-1}	1.21×10^{-1}	-9.00×10^{-1}	-2.22×10^{-2}	9.22×10^{-1}
$\mathbf{X}w^*$	$\mathbf{1.07 \times 10^{0}}$	$\mathbf{9.89 \times 10^{-1}}$	$\mathbf{8.43 \times 10^{-1}}$	$\mathbf{-7.29 \times 10^{-1}}$	$\mathbf{-1.05 \times 10^{-1}}$	$\mathbf{6.61 \times 10^{-1}}$

VII

Conclusion and Future work

> Begin thus from the first act, and proceed; and, in conclusion, at the ill which thou hast done, be troubled, and rejoice for the good.
>
> —Pythagoras

1- Conclusion

In this thesis, three novel online algorithms, called CIRR, CNLS and OSLOG are discussed. The three algorithms are an improvement to the AAR, RR and GD, It shown theoretically, why and under what conditions on the proposed algorithms are better than the three state-of-the-art. More specifically, derivations and the analysis of the proposed algorithms unveil the following:

1. CIRR's and OSLOG's regret is bounded by a logarithmic function of time and are more competitive $-$ has a better regret than the state-of-the-art algorithms for the bounded weights. For details see Theorems 8, 6, 11 and 13. This implies (based on regret analysis), CIRR is a better learner.

2. Theorem 4 shows a simpler formulation of SLOG, which does not require a hierarchical structure as used in [85]. In other words, this means that SLOG hierarchical structure is an approximation of the $\ell_1 - norm$.

3. Theorem 7 highlights the difference in SLOG and OSLOG. SLOG requires variance $\sigma^2 \to 0$, while OSLOG requires $\sigma^2 = 4Y^2$. In this sense, OSLOG could be considered as an online variant of the Bayesian Lasso with known fixed σ^2. OSLOG considers a fixed variance, while SLOG sets variance to null.

4. CIRR has a better guarantee and regret than OSLOG under all circumstances. This implies from Theorems 11 and 13.

5. When $\|w\|_2^2 \to \infty$ and $-1 < b \leq 0.5625R^2$, the CNLS algorithm has a better guarantee than NGD's guarantee for the cumulative squared loss. Similarly, for the normalised squared loss when $\|w\|_2^2 \to \infty$ and $0 < a \leq 0.5625$ CNLS algorithm has a better guarantee than the NGD. Please see Corollary 4 and Remark 13.

6. Theorems 14 and 15 imply that the presence of $a > 0$ and $b > -1$ next to the ridge penalty in the guarantees and in the update rule of CNLS indicates better control over the bias variance trade-off in comparison to the NGD guarantee and the update rule where $a = b = 1$.

7. CNLS is computationally more efficient than CIRR and OSLOG. From Protocol 7 and 9, it is clear that the most expansive op-

eration is the inversion of the matrix, which with the application of Sherman-Morrison implies CIRR and OSLOG have the time complexity of $\mathcal{O}(n^2)$. Time complexity of CNLS is easy to see from Protocol 11, as there is no matrix inversion, only vector multiplications, thus the time complexity is $\mathcal{O}(n)$. NGD is the most computationally efficient algorithm among the three, while CIRR has the best regret.

Empirically, a set of data with different sizes, noise levels and outlier(s) are studied. It is shown that the proposed algorithms perform well better than the state-of-the-art in the setting where the tuning parameter is fixed and no data is given to learn to any algorithm. Practically, this means that scenarios where there no data is available to learn upon and the decision-making is in real time, the proposed algorithms are likely to outperform the state-of-the-art discussed.

2- Future work

There are number of possible future directions, for example:

1. The investigation of algorithms empirical properties in various experimental settings is an important possible direction for future research. One could compare these algorithms' performance with other batch algorithms, as a baseline.

2. In this thesis, there was no mention of the tightness of the mentioned bounds. This question is addressed in a lose sense[1] by Vovk [113] and Orabona et al. [77]. An interesting future direction will be to study the tightness of the bounds without making the stochastic assumptions. One may extend these algorithms to a setting where the best-performing function is not fixed, please see [75].

3. The discussed bounds only consider squared loss. One may study algorithms under different loss functions, such as logarithmic loss, absolute loss, etc.

4. The adjustment of the tuning parameter in online manner for these regression algorithms remains an important open question.

5. The algorithms presented in this thesis, may as well be studied in Hilbert and Banach spaces as done in [34, 123] to learn nonlinearity.

[1] tightness of the bound studied by making stochastic assumptions

6. Also, it will be interesting to extend the proposed algorithms in a continuous time step.

7. Another interesting direction to study non-linearity is to extend the proposed algorithms under various activation functions. A good starting point could be to make use of the same activation function as in [124].

References

History never really says goodbye. History says, ‘See you later’.

—Eduardo Galeano

[1] Abbott, S. (2001). *Understanding analysis*. Springer.

[2] Akbilgic, O., Bozdogan, H., and Balaban, M. E. (2014). A novel hybrid rbf neural networks model as a forecaster. *Statistics and Computing*, 24(3):365–375.

[3] Akgün, B. and Öğüdücü, Ş. G. (2015). Streaming linear regression on spark mllib and moa. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 1244–1247.

[4] Alpher, R. A., Bethe, H., and Gamow, G. (1948). The origin of chemical elements. *Physical Review*, 73(7):803.

[5] Anava, O., Hazan, E., Mannor, S., and Shamir, O. (2013). Online learning for time series prediction. In *COLT*, pages 172–184.

[6] Andrews, D. F. and Mallows, C. L. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(1):99–102.

[7] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

[8] Azoury, K. S. and Warmuth, M. K. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning.*, 43(3).

[9] Bayes, M. and Price, M. (1763). An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs. *Philosophical Transactions of the Royal Society of London Series I*, 53:370–418.

[10] Beckenbach, E. F. and Bellman, R. (2012). *Inequalities*, volume 30. Springer Science & Business Media.

[11] Bershad, B. (1986). Analysis of the normalized lms algorithm with gaussian inputs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):793–806.

[12] Bifet, A. and Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM.

[13] Bifet, A., Gavaldà, R., Holmes, G., and Pfahringer, B. (2018). *Machine learning for data streams: with practical examples in MOA*. MIT Press.

[14] Bitmead, R. and Anderson, B. (1980). Performance of adaptive estimation algorithms in dependent random environments. *IEEE Transactions on Automatic Control*, 25(4):788–794.

[15] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

[16] Budincsevity, N. (2016). Weather in szeged 2006-2016. `https://www.kaggle.com/budincsevity/szeged-weather#`.

[17] Busuttil, S. and Kalnishkan, Y. (2007). Online regression competitive with changing predictors. In *International Conference on Algorithmic Learning Theory*, pages 181–195. Springer.

[18] Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K. (1997). How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485.

[19] Cesa-Bianchi, N., Freund, Y., Helmbold, D. P., and Warmuth, M. K. (1996a). On-line prediction and conversion strategies. *Machine Learning*, 25(1):71–110.

[20] Cesa-Bianchi, N., Long, P., and Warmuth, M. (1996b). Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. *IEEE Transactions on Neural Networks*, 7(3):604–619.

[21] Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

[22] Choubin, B., Moradi, E., Golshan, M., Adamowski, J., Sajedi-Hosseini, F., and Mosavi, A. (2019). An ensemble prediction of flood susceptibility using multivariate discriminant analysis, classification and regression trees, and support vector machines. *Science of the Total Environment*, 651:2087–2096.

[23] Crammer, K., Dredze, M., and Pereira, F. (2009). Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems*, pages 345–352.

[24] DeSantis, A., Markowsky, G., and Wegman, M. N. (1988). Learning probabilistic prediction functions. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 110–119. IEEE.

[25] Diniz, P. S. et al. (1997). *Adaptive filtering*. Springer.

[26] Duda, R. O., Hart, P. E., and Stork, D. G. (1995). Pattern classification and scene analysis 2nd ed. *ed: Wiley Interscience*.

[27] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

[28] Forster, J. (1999). On relative loss bounds in generalized linear regression. In *International Symposium on Fundamentals of Computation Theory*, pages 269–280. Springer.

[29] Foster, D. P. (1991). Prediction in the worst case. *The Annals of Statistics*, pages 1084–1090.

[30] Foster, D. P. and Vohra, R. V. (1993). A randomization rule for selecting forecasts. *Operations Research*, 41(4):704–709.

[31] Freund, Y. (1996). Predicting a binary sequence almost as well as the optimal biased coin. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 89–98.

[32] Gama, J., Medas, P., Castillo, G., and Rodrigues, P. (2004). Learning with drift detection. In *Brazilian symposium on artificial intelligence*, pages 286–295. Springer.

[33] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37.

[34] Gammerman, A., Kalnishkan, Y., and Vovk, V. (2004). On-line prediction with kernels and the complexity approximation principle. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 170–176. AUAI Press.

[35] Garrigues, P. and Ghaoui, L. E. (2009). An homotopy algorithm for the lasso with online observations. In *Advances in neural information processing systems*, pages 489–496.

[36] Gauss, C.-F. (1823). *Theoria combinationis observationum erroribus minimis obnoxiae*, volume 1. Henricus Dieterich.

[37] Gelfand, I. and Ponomarev, V. (1970). Problems of linear algebra and classification of quadruples of subspaces in a finite-dimensional vector space. *Coll. Math. Spc. Bolyai*, 5:163–237.

[38] Gill, J. (1977). Computational complexity of probabilistic turing machines. *SIAM Journal on Computing*, 6(4):675–695.

[39] Golovin, D., McMahan, B., and Sculley, D. (2016). Online learning with maximal no-regret l_1 regularization. In *NIPS workshop on Optimization for Machine Learning*.

[40] Gomes, H. M., Barddal, J. P., Ferreira, L. E. B., and Bifet, A. (2018). Adaptive random forests for data stream regression. In *ESANN*.

[41] Goodwin, G. C. and Sin, K. S. (2014). *Adaptive filtering prediction and control*. Courier Corporation.

[42] Hassibi, B., Sayed, A. H., and Kailath, T. (1996). H/sup/spl infin//optimality of the lms algorithm. *IEEE Transactions on Signal Processing*, 44(2):267–280.

[43] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85.

[44] Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.

[45] Haussler, D., Kivinen, J., and Warmuth, M. K. (1995). Tight worst-case loss bounds for predicting with expert advice. In *European Conference on Computational Learning Theory*, pages 69–83. Springer.

[46] Hayes, M. (1996). 9.4: Recursive least squares. *Statistical Digital Signal Processing and Modeling*, page 541.

[47] Hazan, E. et al. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

[48] Hoens, T. R., Chawla, N. V., and Polikar, R. (2011). Heuristic updatable weighted random subspaces for non-stationary environments. In *2011 IEEE 11th International Conference on Data Mining*, pages 241–250. IEEE.

[49] Hulten, G., Spencer, L., and Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106.

[50] Ikonomovska, E., Gama, J., Sebastião, R., and Gjorgjevik, D. (2009). Regression trees from data streams with drift detection. In *International Conference on Discovery Science*, pages 121–135. Springer.

[51] Jamil, W. and Bouchachia, A. (2018). Model selection in online learning for times series forecasting. In *UK Workshop on Computational Intelligence*, pages 83–95. Springer.

[52] Kakade, S. M. and Ng, A. Y. (2005). Online bounds for bayesian algorithms. In *Advances in neural information processing systems*, pages 641–648.

[53] Kalnishkan, Y. (2015). Predictive complexity for games with finite outcome spaces. In *Measures of Complexity*, pages 117–139. Springer.

[54] Kalnishkan, Y. (2016). An upper bound for aggregating algorithm for regression with changing dependencies. In *International Conference on Algorithmic Learning Theory*, pages 238–252. Springer.

[55] Kelly, M. G., Hand, D. J., and Adams, N. M. (1999). The impact of changing populations on classifier performance. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 367–371.

[56] Kivinen, J. and Warmuth, M. (1999). Averaging expert predictions. In *Computational Learning Theory*, pages 638–638. Springer.

[57] Kivinen, J. and Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63.

[58] Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. *Intelligent data analysis*, 8(3):281–300.

[59] Knotters, M., Brus, D., and Voshaar, J. O. (1995). A comparison of kriging, co-kriging and kriging combined with regression for spatial interpolation of horizon depth with censored observations. *Geoderma*, 67(3-4):227–246.

[60] Kolmogorov, A. N. and Uspenskii, V. A. (1988). Algorithms and randomness. *Theory of Probability & Its Applications*, 32(3):389–412.

[61] Kotowicz, J. (1990). Convergent real sequences. upper and lower bound of sets of real numbers. *Formalized Mathematics*, 1(3):477–481.

[62] Kuhn, H. and Tucker, A. (1951). Proceedings of 2nd berkeley symposium.

[63] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

[64] Kuncheva, L. I. and Žliobaitė, I. (2009). On the window size for classification in changing environments. *Intelligent Data Analysis*, 13(6):861–872.

[65] Langford, J., Li, L., and Zhang, T. (2009). Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10(Mar):777–801.

[66] Lazarescu, M. M., Venkatesh, S., and Bui, H. H. (2004). Using multiple windows to track concept drift. *Intelligent data analysis*, 8(1):29–59.

[67] Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318.

[68] Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, 108(2):212–261.

[69] Littlestone, N., Warmuth, M. K., et al. (1989). *The weighted majority algorithm*. University of California, Santa Cruz, Computer Research Laboratory.

[70] Liu, C., Hoi, S. C., Zhao, P., and Sun, J. (2016). Online arima algorithms for time series prediction. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[71] Maloof, M. A. and Michalski, R. S. (1995). A method for partial-memory incremental learning and its application to computer intrusion detection. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pages 392–397. IEEE.

[72] Minku, L. L., White, A. P., and Yao, X. (2009). The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on knowledge and Data Engineering*, 22(5):730–742.

[73] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill, New York.

[74] Monti, R. P., Anagnostopoulos, C., and Montana, G. (2016). A framework for adaptive regularization in streaming lasso models. *arXiv preprint arXiv:1610.09127*.

[75] Moroshko, E., Vaits, N., and Crammer, K. (2015). Second-order non-stationary online learning for regression. *Journal of Machine Learning Research*, 16:1481–1517.

[76] Nash, J. F. et al. (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49.

[77] Orabona, F., Cesa-Bianchi, N., and Gentile, C. (2012). Beyond logarithmic bounds in online learning. In *Artificial Intelligence and Statistics*, pages 823–831.

[78] Ore, O. (1960). Pascal and the invention of probability theory. *The American Mathematical Monthly*, 67(5):409–419.

[79] Osborne, M. J. et al. (2004). *An introduction to game theory*, volume 3. Oxford university press New York.

[80] Osborne, M. J. and Rubinstein, A. (1994). *A course in game theory*. MIT press.

[81] Oza, N. C. and Russell, S. (2001). *Online ensemble learning*. University of California, Berkeley.

[82] Park, J. and Edington, D. W. (2001). A sequential neural network model for diabetes prediction. *Artificial intelligence in medicine*, 23(3):277–293.

[83] Park, T. and Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

[84] Quinonero-Candela, J., Dagan, I., Magnini, B., and d'Alché Buc, F. (2006). *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment, First Pascal Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944. Springer.

[85] Rajaratnam, B., Roberts, S., Sparks, D., and Dalal, O. (2016). Lasso regression: estimation and shrinkage via the limit of gibbs sampling. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):153–174.

[86] Robert, C. (2014). *Machine learning, a probabilistic perspective*. Taylor & Francis.

[87] Rudin, W. et al. (1976). *Principles of mathematical analysis*, volume 3. McGraw-hill New York.

[88] Salganicoff, M. (1993). Density-adaptive learning and forgetting. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 276–283.

[89] Schmidt, M. (2005). Least squares optimization with l1-norm regularization. *CS542B Project Report*, pages 14–18.

[90] Shafer, G. and Vovk, V. (2005). *Probability and finance: it's only a game!*, volume 491. John Wiley & Sons.

[91] Shafer, G. and Vovk, V. (2019). *Game-Theoretic Foundations for Probability and Finance*, volume 455. John Wiley & Sons.

[92] Shaffer, J. P. (1991). The gaussâĂŤmarkov theorem and random regressors. *The American Statistician*, 45(4):269–273.

[93] Shalev-Shwartz, S. et al. (2012). Online learning and online convex optimization. *Foundations and Trends ® in Machine Learning*, 4(2):107–194.

[94] Shaman, P. (1969). On the inverse of the covariance matrix of a first order moving average. *Biometrika*, 56(3):595–600.

[95] Shoeb, A. H. and Guttag, J. V. (2010). Application of machine learning to epileptic seizure detection. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 975–982.

[96] Sion, M. (1958). On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176.

[97] Sleator, D. D. and Tarjan, R. E. (1985). Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208.

[98] Solomonoff, R. J. (1964). A formal theory of inductive inference. part i. *Information and control*, 7(1):1–22.

[99] Spark, A. (2018). Apache spark. *Retrieved January*, 17:2018.

[100] Su, H., Liu, H., and Wu, Q. (2015). Prediction of water depth from multispectral satellite imageryâĂŤthe regression kriging alternative. *IEEE Geoscience and Remote Sensing Letters*, 12(12):2511–2515.

[101] Sun, Y., Tang, K., Minku, L. L., Wang, S., and Yao, X. (2016). Online ensemble learning of data streams with gradually evolved classes. *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1532–1545.

[102] Syed, N. A., Liu, H., and Sung, K. K. (1999). Handling concept drifts in incremental learning with support vector machines. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–321.

[103] Tao, T. (2011). *An introduction to measure theory*. American Mathematical Society Providence, RI.

[104] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

[105] Vaits, N. and Crammer, K. (2011). Re-adapting the regularization of weights for non-stationary regression. In *International Conference on Algorithmic Learning Theory*, pages 114–128. Springer.

[106] Van Rijn, J. N., Bischl, B., Torgo, L., Gao, B., Umaashankar, V., Fischer, S., Winter, P., Wiswedel, B., Berthold, M. R., and Vanschoren, J. (2013). Openml: A collaborative science platform. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 645–649. Springer.

[107] Vlachos, P. and Meyer, M. (2005). Statlib datasets archive. *URL http://lib. stat. cmu. edu/datasets*.

[108] Vovk, V. (1990). Aggregating strategies. In *Proc. Third Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann.

[109] Vovk, V. (1992). Universal forecasting algorithms. *Information and Computation*, 96(2):245–277.

[110] Vovk, V. (1995). A game of prediction with expert advice. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 51–60. ACM.

[111] Vovk, V. (1998). A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173.

[112] Vovk, V. (1999). Derandomizing stochastic prediction strategies. *Machine Learning*, 35(3):247–282.

[113] Vovk, V. (2001). Competitive on-line statistics. *International Statistical Review/Revue Internationale de Statistique*, pages 213–248.

[114] Vovk, V. and Zhdanov, F. (2009). Prediction with expert advice for the brier game. *Journal of Machine Learning Research*, 10(Nov):2445–2471.

[115] Vyugin, M. V. and V'yugin, V. V. (2002). Predictive complexity and information. In *International Conference on Computational Learning Theory*, pages 90–105. Springer.

[116] Vyugin, M. V. and V'yugin, V. V. (2005). Predictive complexity and information. *Journal of Computer and System Sciences*, 70(4):539–554.

[117] Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., and Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994.

[118] Widrow, B. and Walach, E. (1984). On the statistical efficiency of the lms algorithm with nonstationary inputs. *IEEE Transactions on Information Theory*, 30(2):211–221.

[119] Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596.

[120] Yamanishi, K. (1995). Randomized approximate aggregating strategies and their applications to prediction and discrimination. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 83–90.

[121] Young, N. (1988). *An introduction to Hilbert space*. Cambridge university press.

[122] Zhang, P., Zhu, X., and Shi, Y. (2008). Categorizing and mining concept drifting data streams. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 812–820.

[123] Zhdanov, F., Chernov, A., and Kalnishkan, Y. (2010). Aggregating algorithm competing with banach lattices. *arXiv preprint arXiv:1002.0709*.

[124] Zhdanov, F. and Vovk, V. (2010). Competitive online generalized linear regression under square loss. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 531–546. Springer.