Competitive Regularised Regression

Waqas Jamil*, Abdelhamid Bouchachia

Department of Computing and Informatics,Bournemouth University, Poole, UK

Abstract

Regularised regression uses sparsity and variance to reduce the complexity and over-fitting of a regression model. The present paper introduces two novel regularised linear regression algorithms: Competitive Iterative Ridge Regression (CIRR) and Online Shrinkage via Limit of Gibbs Sampler (OSLOG) for fast and reliable prediction on "Big Data" without making distributional assumption on the data. We use the technique of competitive analysis to design them and show their strong theoretical guarantee. Furthermore, we compare their performance against some neoteric regularised regression methods such as Online Ridge Regression (ORR) and the Aggregating Algorithm for Regression (AAR). The comparison of the algorithms is done theoretically, focusing on the guarantee on the performance on cumulative loss, and empirically to show the advantages of CIRR and OSLOG.

Keywords: Regression, Regularisation, Online learning, Competitive analysis

1. Introduction

Regularised regression is a general convex optimisation problem and is useful in many real-world applications [1, 2]. L_1 and L_2 regularisation are the two most popular regularisation methods for regression. L_1 regularised regression aims to obtain a sparse solution. When we require a model that outputs few non-zero entries we prefer L_1 regularised regression [3, 4]. By setting prior belief about

*Corresponding author

Email address: {wjamil,abouchachia}@bournemouth.ac.uk (Waqas Jamil*, Abdelhamid Bouchachia)

sparsity in the model, one can choose a suitable model [5]. 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 data that has high 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 [6].

AAR and ORR algorithms compete against the least squares as a benchmark. These online algorithms perform close to the benchmark in the worst case. Our proposed algorithms are in the same framework. However, unlike them we consider IRR algorithm instead of RR. IRR was suggested by [6] for LASSO approximation, but in [6] substantial details about the method were skipped, [7] filled in the details and argued that IRR is an efficient method due to its resemblance with the Newton method. Hence, the presented algorithms are close to L_1 regularised regression, which consequently allows the algorithms to have a better ability to deal with multicollinearity and sparsity in some sense.

Online learning framework in learning theory is formalised as a game played between a learner and the nature [8]. The goal of the learner is to predict the outcome that nature outputs. The learner updates its parameters after nature announces the actual outcome and the process is repeated for every new input. So, online learning is useful when the application lends itself continuous learning like a stockbroker who has to make regular decisions based on the experience acquired so far or when there is too much data that can't fit into the memory at once. Moreover, the input of the problem at hand may continuously evolve, that is, the underlying distribution of the input change over time leading to what is known as *concept drift* [9, 10]. Other genuine applications of online learning include click-through prediction, online advertising, marketing optimisation, real-time bidding, trading, etc.

In this paper, we consider competitive prediction where we make no assumptions on the data generating process [11]. Competitive analysis is a type of analysis for online learning algorithms where the upper bound is close to the

batch optimisation problem. In our case, the comparison to batch optimisation problem is the minimum of the difference of sum of squares. Thus, we use game-theoretic framework instead of combinatorial basis [12].

Consider a perfect-information game played between three players. The learner, the decision pool and the nature [13]. Let Ω denote the sample space, Γ decision space and Θ the parameter space. For each trial, the learner chooses a decision $w : \Theta \to \Gamma$. The learner then makes prediction $\hat{y}_t \in \Gamma$, after which the reality announces the outcome $y_t \in [-Y, Y]$. Here we consider fixed loss function $\lambda : \Omega \times \Gamma \to [0, \infty]$. The learner's objective is to ensure that its cumulative loss is at most equal to the cumulative loss of the the best decision strategy from the decision pool. More precisely,

$$L_T(\text{Learner}) \leq cL_T^* + p \ln n \tag{1}$$

where $L_T(\text{Learner}) = \sum_{t=1}^{T} \lambda(\omega, \hat{y}_t)$ and $L_T^* = \sum_{t=1}^{T} \lambda(\omega, w_t)$. Here w_t denotes the prediction of the best learning strategy up until time t, c and p are constants in $\mathbb{R}$. The number of decision strategies are finite $|\Theta| = n$. Inequality (1) was proven in [14, 15], where it is not assumed that the outcomes announced by nature are generated from some stochastic mechanism. The Aggregation Algorithm for regression (AAR) presented in [16] is a popular algorithm in the literature of competitive online prediction. In reference to perfect information game; the learner is AAR; the decision pool is denoted by w_t and the nature generates input $x_t \in \mathbb{R}^n$ and output $y_t \in \mathbb{R}$ signals. It achieves the following bound by confining the input and weights to the unit balls in the metrics L_∞ and L_1 respectively [16]:

$$L_T \leq L_T^* + \mathcal{O}(\ln T) \tag{2}$$

where L_T^* is the loss of RR on the data until trial T. In the present paper, instead of RR we consider Iterated Ridge Regression (IRR) as a penalty to:

1. derive two novel algorithms, Competitive Iterative Ridge Regression (CIRR) and Online Shrinkage via Limit of Gibbs Sampler (OSLOG) and we give upper bounds on the cumulative loss. A comparison of the upper bounds

with state-of-the-art, reveals, that the CIRR and OSLOG upper bounds are better in certain circumstances (Theorem 3 and Theorem 5).

2. compare the performance of CIRR and OSLOG against the optimal decision strategy and popular stat-of-the-art online algorithms, namely, AAR, Online RR (ORR), Online Gradient Decent (OGD) and Online Newton Step (ONS), on synthetic and real-world adaptive sparse datasets (Section 6).
3. the two algorithms can obtain a faster and accurate regression in comparison to other methods.

The paper is organised as follows. Section 2 reviews the relevant literature. Section 3 formulates the problem and introduces the notation. Section 4 gives the details of CIRR. Section 5 shows that the Bayesian formulation of CIRR leads to OSLOG. Section 6 discusses the empirical evaluation. Section 7 concludes the paper.

2. Literature review

The thought of comparing the best offline algorithm to online algorithms originated from [17] and the term "competitive analysis" was first used by [18]. The adjective "online" mostly appears in computer science literature which is synonymous with "prequential" in statistics. Prequential statistics introduced by [19] makes predictions sequentially, rather than just expressing information about the parameters [20]. Probably [21] were the first to perform competitive analysis on Bayesian mixing technique for the log loss prediction game. [22, 23] presented an online algorithm that makes the prediction based on the weights. Later [24] generalised the Bayesian mixing technique resulting in a new algorithm called the *Aggregation Algorithm* (AA). By using AA with Gaussian prior one can obtain AAR regression algorithm which has a strong performance guarantee, as mentioned in the previous section. There are few variants of AAR algorithm such as ARROW [25], RLS [26], ORR [16] etc.

Algorithm	Predictive complexity	Time complexity	bounded loss
OGD	$L_T \leq L_T^* + \mathcal{O}(\sqrt{T})$	$\mathcal{O}(n)$	Yes
ONS	$L_T \leq L_T^* + \mathcal{O}(\ln L_T^*)$	$\mathcal{O}(n^2)$	Yes
AAR	$L_T \leq L_T^* + \mathcal{O}(\ln T)$	$\mathcal{O}(n^2)$	No

Table 1: Algorithms complexity.

Table 1 summarises the popular algorithms with their respective complexities. OGD is the most computationally efficient algorithm, but has the weakest guarantee. ONS has a better guarantee, but under the condition that the losses are bounded. The rest of this section is mostly devoted on the discussion of the three algorithms and their variants mentioned in Table 1.

Researchers have studied the classical problem of online regression extensively in the past. Broadly speaking, there exist two main approaches to tackle the online regression problem. The first approach was introduced half a century ago by [27] for reducing noise via adaptive filtering. The algorithm is known as Least Mean Squares (LMS), it updates weights by using Gradient Descent (GD). Later [28, 29] performed analysis on LMS showing that Normalised LMS (NMLS) is insensitive to scaling of the input. In [26] an algorithm known as Recursive Least Squares (RLS) was introduced for online regression. RLS uses a correction factor to update covariance matrix at each iteration. [16, 30, 31] theoretically studied a variant of RLS known as AAR − an algorithm with the strongest theoretical guarantee under this approach.

In contrast to this, [32] studied the bounds of GD based online regression with square loss. Later [33] replaced GD 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 the square root of the number of trials T under online

setting. An example of the upper bound on the cumulative square loss of a GD-based linear regression algorithm is as follows [32]:

$$\sum_{t=1}^{T}(\gamma_t^{\mathrm{GD}} - y_t) \leq 9 \inf_{w\in\mathbb{R}^n}\left(\sum_{t=1}^{T}(w'x_t - y_t)^2 + \sup_{t=1,\ldots,T}||x_t||_\infty^2||w_t||_2^2\right) \quad (3)$$

where γ_t^{GD} denotes the prediction at step t, y_t denotes the outcome at step t, x_t is the input vector of attributes at step t and w_t is the weight vector at time t. For the noise free case, by assuming $||x_t||_\infty \leq R$, Inequality (3) reduces to [32]:

$$\sum_{t=1}^{T}(\gamma_t^{\mathrm{GD}} - y_t) \leq 2.25 \inf_{\theta\in\mathbb{R}^n}\left(\sum_{t=1}^{T}(w'x_t - y_t)^2 + R||w||_2^2\right) \quad (4)$$

Inequality (3) and (4) are not comparable to the bounds obtained by [33], 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 (4) [32]. 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 [32, 33] are of the following type:

$$L_T \leq L_T^* + \mathcal{O}(\sqrt{L_T^*}) \quad (5)$$

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^*, which is not required for AAR. [34] obtained the upper loss bound using online Newton step of the of type:

$$L_T \leq L_T^* + \mathcal{O}(\ln L_T^*) \quad (6)$$

Inequality (6) 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 [34] 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 [13] multiplicative factor of Y^2. AAR and the algorithm proposed by [34] both have computational complexity of $\mathcal{O}(n^2)$.

In [35] Coordinate Descent (CD) is used 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, [36] extended AAR by using [30] methodology and called it LASER. They also considered extension of the algorithms discussed in [37, 38]. Authors in [39] proved similar bound to LASER by extending [40] work.

Recently, [41] 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, [42] 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, [43] proposed an online regression algorithm. However, they did not study the bounds.

We now proceed and formalise the problem, i.e. we present a linear benchmark function against which we will compete to achieve the upper bound on the cumulative loss of the type stated in inequality (6) in a similar setting as AAR.

3. Problem formulation

Considering a sequence of instances and outcomes $(x_1, y_1), ..., (x_t, y_t)$. Letting $w_t \in \Theta = \mathbb{R}^n$ to denote the decision strategy at time t and let $w_{t,i}$, for $i = 1, ...n$ denote the $i-$th component of the decision vector at time t.

We assume that the input is taken from the $L_\infty-$ ball $\{x_t \in \mathbb{R}^n : ||x||_\infty \leq R\}$ of radius R and the vector w_t is indexed by $\Theta = \{w_t \in \mathbb{R}^n : C \leq ||w_t||_1 \leq P\}$. Also, assuming the prediction on trial t is given by $w_t' x_t$.

We define the following quantities:

$$b_t := \sum_{s=1}^{t} y_s x_s \in \mathbb{R}^n \tag{7}$$

$$A_t := \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s' \right) \in \mathbb{R}^{n \times n},\ a > 0 \tag{8}$$

where

$$D_{w_{t-1}} = \text{diag}(|w_{t-1,1}|, ..., |w_{t-1,n}|) \tag{9}$$

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

$$L_t(w_t) := \sum_{s=1}^{t} (y_s - w_{s-1}' x_s)^2 \tag{10}$$

we denote $\triangledown f(w_t)$ for first derivative and $H \triangledown f(w_t)$ (H is for Hessian matrix) for second derivative with respect to w_t.

We aim to compete against IRR, which was suggested as an approximate solution for the following problem:

$$\inf_{w_t \in \mathbf{R}^n} (L_t(w_t) + a||w_t||_1) \tag{11}$$

where $a > 0$. The problem (11) 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. So, we compete with the approximation of dual vector [44, 6, 7]. The method is based on the following approximation:

$$|w_{t,i}| \approx \frac{(w_{t,i})^2}{|w_{t-1,i}|} \tag{12}$$

for $i = 1, 2, ..., n$. Substituting (12) into (11) gives an expression similar to ridge regression, which is as follows (see Equation (22) in [44]):

$$w_t = \left(\mathbf{X}'\mathbf{X} + aD_{w_{t-1}}^{-1} \right)^{-1} \mathbf{X}' y \tag{13}$$

where $\mathbf{X}$ is the design matrix and $y \in \mathbb{R}^n = (y_1, y_2, ..., y_n)$. The derivation of our algorithm will imply that (13) is a solution to the following optimisation

problem:

$$\inf_{w_t \in \mathbf{R}^n} \left(L_t(w_t) + a||D_{w_{t-1}}^{-\frac{1}{2}} w_t||^2 \right) \tag{14}$$

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 we will do latter, but for simplicity let's keep this restriction. Interestingly (13) coincides with a more recent algorithm known as Shrinkage via Limit of Gibbs Sampler (SLOG) presented by [41] who show that under mild assumptions, SLOG converges to LASSO. The SLOG algorithm is inspired by the Bayesian LASSO [45], which focuses on estimating the posterior mean of coefficients using Gibbs sampling. The high time complexity of the Gibbs sampler makes the full Bayesian implementation of the LASSO less attractive for practitioners [10]. SLOG improves the computational aspect of Bayesian LASSO without affecting the performance.

In our work, we allow CIRR and OSLOG to make only one pass over the data without making use of the Gibbs sampler. We perform theoretical analysis in both game-theoretic and Bayesian settings. In order to make our results more interpretive, we use Cauchy-Schwartz inequality and compete against the following:

$$\inf_{w_t \in \mathbf{R}^n} \left(L_t(w_t) + a||D_{w_{t-1}}^{-\frac{1}{2}} w_t||^2 \right) \leq \inf_{w_t \in \mathbf{R}^n} \left(L_t(w_t) + aSS||w_t||^2 \right) \tag{15}$$

where SS is the sum of squares of the diagonal matrix $D_{w_{t-1}}$ elements. We show in Section 4 that inequality (15) holds. However, our bound will imply for (14). We use (15) for ease of interpretation.

Now that we have formulated the problem, we present next our novel algorithm CIRR along with its theoretical analysis.

4. Formulation of CIRR

In this section the derivation and analysis of CIRR algorithm are presented. Inspired by AAR algorithm we propose a novel algorithm Competitive Iterative Ridge Regression (CIRR). We follow the approach of competitive analysis done in [16, 40, 39, 30] .To summarise:

- Lemma 1 is the derivation of the weights updating equation and Lemma 2 generalises the weights updating equation
- By using Lemma 1, Theorem 1 derives CIRR prediction at any given trial.
- Lemma 3 is used later in Theorem 2, which discusses the upper bound on cumulative square loss of the CIRR algorithm.
- Theorem 3 compares the bound of CIRR with AAR under similar conditions.

Protocol 1 shows the framework under which CIRR work. In this protocol, we notice that the learner does not know the label at the time of prediction, but it knows the moves made by the decision pool $w_t \in \mathbb{R}^n$ at each trial t and prediction, $w_t' x_t$, is computed. It is worth noting that our strategy interacts with the decision pool twice. In contrast to AAR, the learner does not need to interact with the decision pool explicitly.

Protocol 1 : Learning strategy of CIRR

1: **for** $t = 1, 2, ...$ **do**
2: Nature chooses $x_t \in \mathbb{R}^n$
3: Learner chooses $w_t \in \Theta$
4: Learner predicts $\hat{y}_t \in \mathbb{R}$
5: Nature chooses $y_t \in [-Y, Y]$
6: Learner chooses $w_t \in \Theta$
7: **end for**

The following Lemma gives the best decision strategy from the decision pool.

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

$$w_t = A_t^{-1} b_t \quad and \quad f(w_t) = \sum_{s=1}^{t} y_s^2 - b_t A_t^{-1} b_t$$

such that none of the elements of the weight vector has its absolute value at any step equal to zero. The definition of b_t, A_t, $D_{w_{t-1}}^{-\frac{1}{2}}$ *and* $L_t(w_t)$ *is given in (7), (8), (9) and (10) respectively.*

Proof. From the definition

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

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

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

$$f(w_t) = \sum_{s=1}^{t} y_s^2 - 2w_t' b_t + w_t' A_t w_t \tag{16}$$

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

We proceed by differentiating with respect to w_t (we treat w_{t-1} as a constant when differentiating with respect to w_t)

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

and

$$H \nabla f(w_t) = 2 \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s' \right)$$

Since $\nabla f(w_t) = 0 - 2b_t + 2A_t w_t$ and $H \nabla f(w_t) = 2A_t \Rightarrow f$ is convex, so to attain the minimal point w_t, setting $\nabla f(w_t) = 0$ i.e.

$$w_t = b_t' A_t^{-1}$$

Thus,

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

$$f(w_t) = \sum_{s=1}^{t} y_s^2 - b_t A_t^{-1} b_t \tag{17}$$

this concludes the proof. □

Lemma 1 gives us the prediction of the best decision strategy. Hence, the prediction of the learning algorithm is given as follows:

Theorem 1. *CIRR predicts* $\hat{y}_t = b'_{t-1}A_t^{-1}x_t$ *at trial* $t = 1, 2, ...$

Proof. To complete Protocol 1, we use Lemma 1 and write:

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

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

$$= \arg \inf_{\hat{y}_t \in \mathbb{R}} \sup_{y_t \in [-Y,Y]} \left(\sum_{s=1}^{t} (y_s - \hat{y}_s)^2 - \sum_{s=1}^{t} y_s^2 + b_{t-1} A_t^{-1} b_{t-1} \right.$$
$$\left. + 2 y_t b'_{t-1} A_t^{-1} x_t + y_t^2 x'_t A_t^{-1} x_t \right) \quad (18)$$

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

$$= \arg \inf_{\hat{y}_t \in \mathbb{R}} \sup_{y_t \in [-Y,Y]} \left(2 y_t (b'_{t-1} A_t^{-1} x_t - \hat{y}_t) + y_t^2 (x'_t A_t^{-1} x_t) + \hat{y}_t^2 \right) \quad (19)$$

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

$$2Y \left(b_{t-1} A_t^{-1} x_t - \gamma_t \right) + \gamma_t^2 \quad (20)$$

(20) 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 (20) 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 (20) 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 $\hat{y}_t = b_{t-1}A_t^{-1}x_t$ (18) attains minimum. □

Before we upper bound the cumulative loss of the CIRR, we state two simple Lemmas to help our cause.

Lemma 2. *For all* $t = 1, 2, ...,\ a > 0$

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

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

Proof. From the properties of a diagonal matrix, we write

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

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

□

By incorporating the prediction of the learner we have an explicit algorithm for CIRR:

Algorithm 1 : The CIRR algorithm

Initialise: $a > 0, \Sigma = \mathbf{0}^{n\times n}$, $b = \mathbf{0}^{n\times 1}$ and $w_0 = \mathbf{1} \in \mathbb{R}^{n\times 1}$.
for $t = 1, 2, ...,$ **do**
  Read $x_t \in \mathbb{R}^n$
  $D_{w_{t-1}} = \text{diag}(|w_{t-1,1}|, ..., |w_{t-1,n}|)$
  $\Sigma = \Sigma + x_t x_t'$
  $A^{-1} = \sqrt{D_{w_{t-1}}} \left(a\mathbf{I} + \sqrt{D_{w_{t-1}}} A \sqrt{D_{w_{t-1}}}\right)^{-1} \sqrt{D_{w_{t-1}}}$ (Lemma 2)
  $w_t = A^{-1}b$ (Lemma 1)
  $\hat{y}_t = w_t' x_t$
  Read $y_t \in \mathbb{R}$
  $b = b + y_t x_t$
  Update $w_t = A^{-1}b$
end for

Remark 1. *The algorithm performs sequentially by processing each data point at each trial. At trial $t = 1$ the algorithm receives $x_1 \in \mathbb{R}^n$, and outputs $\hat{y}_1 = w_0' x_1$, then computes $D_{w_{t-1}} = diag(|w_{0,1}|, ..., |w_{0,n}|)$. This is a diagonal matrix, on the diagonal is $w_0 \in \mathbb{R}^n$. The algorithm than computes the co-variance matrix $\mathbb{R}^{n \times n}$ i.e. $x_1 x_1'$ (where x_1 is a vector of $n \times 1$, so its transpose is $1 \times n$ and results in $n \times n$ matrix), followed by inversion and multiplication by $D_{w_{t-1}}$. Notice, the co-variance matrix is symmetric-semi-positive definite, but the inversion is done for positive definite (addition of $a\mathbf{I}$ ensures that the inverse exist for all trials). Now, the algorithm receives the actual observation after which it updates w and the process continues for $t = 2, 3, ...$*

Algorithm 1 is applicable even when there are components that are exactly zero in the weight vector w_t.

Lemma 3. *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, we have:

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

□

Remark 2. *For $n = m$ in Lemma 3*

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

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 diagonal elements. This justifies the Inequality (15).

We now prove the upper loss bound on Algorithm 1.

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

$$L_T(CIRR) \leq \inf_{w_T \in \mathbb{R}^n} \left(L_T(w_T) + a||D_{w_{t-1}}^{-\frac{1}{2}} w_T||_2^2 \right) + Y^2 \ln \det \left(\frac{1}{a} A_T \right) \quad (21)$$

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

$$L_T(CIRR) \leq L_T(w_T) + aP^2C^{-1} + Y^2 n \ln \left(\frac{a + CTR^2}{aC} \right) \quad (22)$$

Proof. From Theorem 1 in [16] and Theorem 3 in [30] the upper bound on the cumulative loss of AAR is as follows:

$$L_T(AAR) \leq \inf_{w_T \in \mathbb{R}^n} \left(w_T' B_T w - 2w' b_T + \sum_{t=1}^{T} y_t^2 \right) + Y^2 \ln \det \left(\frac{1}{a} B_T \right) \quad (23)$$

where $B_T = \left(a\mathbf{I} + \sum_{t=1}^{T} x_t x_t' \right)$ and $w_T' B_T w_T - 2w_T' b_T + \sum_{t=1}^{T} y_t^2 = L_T(w_T) + ||w_T||_2^2$, here b_T and $L_T(w_T)$ are defined as (7) and (10) respectively. Notice (23) is only true for positive definite matrices and A_T is positive definite so, we replace $Y^2 \ln \det \left(\frac{1}{a} B_T \right)$ with $Y^2 \ln \det \left(\frac{1}{a} A_T \right)$. To elaborate further, expanding and performing some algebraic manipulation on the function $f(w_t)$ in Lemma 1 to obtain:

$$y_t^2 x_t' A_{t-1}^{-1} x_t + b_{t-1} (A_t^{-1} x_t x_t' A_t^{-1} - A_{t-1}^{-1} + A_t^{-1}) b_{t-1} \quad (24)$$

Since, $A_{t-1} - A_t = x_t x_t'$, so $A_{t-1}^{-1} - A_t^{-1} = A_t^{-1} x_t x_t' A_{t-1}^{-1}$ and consequently $A_{t-1}^{-1} - A_t^{-1} - A_t^{-1} x_t x_t' A_t^{-1} = A_t^{-1} x_t x_t' A_t^{-1} x_t x_t' A_t^{-1}$. Thus, (24) can be written as:

$$y_t^2 x_t' A_t^{-1} x_t - (x_t' A_{t-1}^{-1} x_t) b_{t-1}' A_t^{-1} x_t x_t' A_t^{-1} b_{t-1} \quad (25)$$

It is easy to see that the term $(x_t' A_{t-1}^{-1} x_t) b_{t-1}' A_t^{-1} x_t x_t' A_t^{-1} b_{t-1}$ in (25) can be written as $(x_t' A_{t-1}^{-1} x_t) \hat{y}_t^2$ and,

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

summing over $t = 1, 2, ..., T$ leads to the following expression:

$$L_T(CIRR) - \inf_{w_T \in \mathbb{R}^n} \left(L_T(w_T) + a||D_{w_{t-1}}^{-\frac{1}{2}} w_T||_2^2 \right) \leq Y^2 \sum_{t=1}^{T} x_t A_t^{-1} x_t$$

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 $x_t' A_t^{-1} x_t \leq \ln\frac{\det A_t}{\det A_{t-1}}$. For $x_t = 0$, clearly $x_t' A_t x_t < 1$ 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 [46] 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 (21), when A_T is positive definite.

We use the definition of (7), (8), (9) and (10) to write the following:

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

To prove (22) we first need to show the following holds:

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

From Lemma 3, we know that:

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

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

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

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

This concludes the proof. □

We may compare (22) with the following [16]:

$$L_T(AAR) \leq L_T(w_T) + aP^2 + nY^2 \ln\left(1 + \frac{TR^2}{a}\right) \tag{26}$$

$$L_T(ORR) \leq L_T(w_T) + aP^2 + 4nY^2 \ln\left(1 + \frac{TR^2}{a}\right) \tag{27}$$

It is worth noting that AAR considers

$$\inf_{w_t \in \mathbb{R}^n} \left(L_t(w_t) + a||w_t||_2^2 \right) \tag{28}$$

whereas CIRR considers (14). Notice we are scaling w_t, i.e.

Lemma 4. *For all $t = 1, 2, ..., T$, then*

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

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

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

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

Theorem 3. *If, $||x_t||_\infty \leq R$ and $C \leq ||w_t||_1 \leq P$ 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. Let $R_T^*(AAR) = L_T^U(AAR) - L_T(w_T)$ and $R_T^*(CIRR) = L_T^U(CIRR) - L_T(w_T)$. We proceed by showing $R_T^*(CIRR) \leq R_T^*(AAR)$ i.e.

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

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

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

5. Formulation of OSLOG

In this section we discuss Bayesian variant of Algorithm 1. We show that the Bayesian interpretation of CIRR leads to SLOG in online setting, hence we call the Bayesian version of CIRR as OSLOG. We proceed by briefly discussing SLOG.

SLOG is a batch learning algorithm that makes multiple passes over the data until convergence to LASSO. We now allow SLOG to make only one pass over the data. To obtain prediction at time $T+1$ we multiply SLOG's weight updating equation (see Equation (6) in [41]) at time T by x_{T+1}

$$\left(\sum_{t=1}^{T} x_t y_t\right)' \left(D_{w_{T-1}}^{\frac{1}{2}} \left(a\mathbf{I} + D_{w_{T-1}}^{\frac{1}{2}} \left(\sum_{t=1}^{T} x_t x_t'\right) D_{w_{T-1}}^{\frac{1}{2}}\right)^{-1} D_{w_{T-1}}^{\frac{1}{2}}\right) x_{T+1}$$

where $D_{w_{T-1}} = \text{diag}(|w_{T-1,1}|, ..., |w_{T-1,n}|)$. To show that SLOG and OSLOG's weight updating equation is the same, we derive OSLOG's updating equation by Protocol 2. The difference between Protocol 1 and Protocol 2 is that the prediction in Protocol 1 is made before updating the weight, whereas the learner in Protocol 2 only needs to interact with the decision pool once.

Protocol 2 : Learning strategy of OSLOG

1: **for** $t = 1, 2, ...$ **do**
2: Nature chooses $x_t \in \mathbb{R}^n$
3: Learner prediction $w'x_t \in \mathbb{R}$
4: Nature chooses $y_t \in [-Y, Y]$
5: Learner chooses weights $w_t \in \Theta$
6: **end for**

For better readability the overview of the following Lemmas, Theorems and Corollaries is as follows:

- Lemma 5 derives the weight updating equation for OSLOG.
- Theorem 4 presents OSLOG's upper bound on cumulative square loss.
- Theorem 5 compares OSLOG's guarantee with AAR's.

- Corollary 1 highlights an advantage of CIRR over OSLOG.

Lemma 5. *If an algorithm follows Bayesian strategy with likelihood of* $p(y_t|w_t) \sim N(w_t'x_t, \frac{1}{2\eta})$*, where prior over* w_t *is as follows:*

$$p(w_t) = \exp\left(-a\eta w_t' D_{w_{t-1}}^{-1} w_t\right)$$

such that $|w_{t-1,1}|, ..., |w_{t-1,n}| \neq 0, w_0$ *is initialised with uniform distribution and* $a, \eta > 0$*, then following holds:*

$$w_{t+1} = \left(\sum_{s=1}^{t} x_s y_s\right)' \left(D_{w_{t-1}}^{\frac{1}{2}} \left(a\mathbf{I} + D_{w_{t-1}}^{\frac{1}{2}} \left(\sum_{s=1}^{t} x_s x_s'\right) D_{w_{t-1}}^{\frac{1}{2}}\right)^{-1} D_{w_{t-1}}^{\frac{1}{2}}\right) \tag{29}$$

Proof. The Bayesian strategy implies the posterior to be proportional to the likelihood times the prior i.e.

$$p(w|y) \propto p(y|w)p(w)$$

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

$$p(w_t|y_t) \propto \exp\left(-a\eta w_t' D_{w_{t-1}}^{-1} w_t - \eta L_t(w_t)\right)$$

$$e^{-a\eta w_t' D_{w_{t-1}}^{-1} w_t - \eta L_t(w_t)} \propto e^{-\eta w_t' \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s'\right) w_t + 2w_t' \eta \left(\sum_{s=1}^{t} y_s x_s\right)}$$

$$-a\eta w_t' D_{w_{t-1}}^{-1} w_t - \eta L_t(w_t) \propto -\eta w_t' \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s'\right) w_t + 2w_t' \eta \left(\sum_{s=1}^{t} y_s x_s\right)$$

By cancelling η and multiplying with the negative sign on both sides we write the above expression as follows:

$$aw_t' D_{w_{t-1}}^{-1} w_t + \eta L_t(w_t) \propto w_t' \left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s'\right) w_t - 2w_t \left(\sum_{s=1}^{t} y_s x_s\right) \tag{30}$$

Minimising the right hand side of the proportionality (30) gives:

$$\inf_{w_t} \left(||aD_{w_{t-1}}^{-\frac{1}{2}}||_2^2 + L_t(w_t)\right)$$

and minimising the expression on the left hand side of the proportionality (30) gives:

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

$$\implies \nabla f(w_t) = -\left(\sum_{s=1}^{t} 2y_s x_s\right) + 2w_t'\left(aD_{w_{t-1}}^{-1} + \sum_{s=1}^{t} x_s x_s'\right) \tag{31}$$

Setting $\nabla f(w_t) = 0$ and applying Lemma 2, we obtain (29). □

Algorithm 2 : The OSLOG algorithm

1: **Initialise:** $a > 0, \Sigma = \mathbf{0}^{n\times n}$, $b = \mathbf{0}^{n\times 1}$ and $w_0 = \mathbf{1} \in \mathbb{R}^{n\times 1}$
2: **for** $t = 1, 2, ...$ **do**
3: Read $x_t \in \mathbb{R}^n$
4: $\hat{y}_t = w' x_t$
5: $D_{w_{t-1}} = \text{diag}(|w_{t,1}|, ..., |w_{t,n}|)$
6: $\Sigma = \Sigma + x_t x_t'$
7: $A^{-1} = \sqrt{D_{w_{t-1}}}\left(a\mathbf{I} + \sqrt{D_{w_{t-1}}}\Sigma\sqrt{D_{w_{t-1}}}\right)^{-1}\sqrt{D_{w_{t-1}}}$ (Lemma 2)
8: Read $y_t \in \mathbb{R}$
9: $b = b + y_t x_t$
10: Update $w = A^{-1}b$ (Lemma 5)
11: **end for**

Remark 3. *The algorithm performs sequentially by processing each data point at each trial see Remark 1. However, notice after arrival of $x_1 \in \mathbb{R}^n$, prediction is $\hat{y}_1 = 0$. So, at the first trial the algorithm does not make use of x_1. This philosophically and mathematically differs from CIRR. Here the algorithm updates after prediction.*

Theorem 4. *For any point in time $t = 1, 2, ..., T$*

$$L_T(OSLOG) \leq \inf_{w_T \in \mathbb{R}^n}\left(L_T(w_T) + aSS||w_T||_2^2\right) + 4Y^2 \ln\det\left(\frac{1}{a}A_{T-1}\right) \tag{32}$$

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

$$L_T(OSLOG) \leq L_T(w_T) + aP^2C^{-1} + 4Y^2 n \ln\left(C^{-1} + \frac{TR^2}{a}\right) \tag{33}$$

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

Proof. From Remark 3 in [16], we know updating weights in Algorithm 2 after making the prediction is 4 times worse than updating weights before making the prediction. The rest of the proof follows same arguments as Theorem 2. □

The following Theorem presents a scenario when OSLOG's upper bound is better than ARR and ORR.

Theorem 5. *If* $||x_t||_\infty \leq R$ *and* $C \leq ||w_t||_1 \leq P$ *such that* $C > 1$, $a > 0$, *and* n *is some positive integer, then* $\forall t$ *the following holds:*

$$L_T^U(OSLOG) < L_T^U(AAR)$$

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

Proof. The proof is analogous to Theorem 3. □

Remark 4. *The regret of OSLOG is smaller than that of the regret of AAR when* $C > 1$.

It is also possible to write an explicit relationship between OSLOG and CIRR. Using Sherman-Morrison formula as used in [33] leads to the following corollary

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

$$\gamma_T = \frac{s_T}{1 + x_T' D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T}$$

where γ_T *denotes the prediction of CIRR and* s_T *denotes the prediction of OSLOG at time* T.

Proof. We proceed as follows:

$$\gamma_T = \left(\sum_{t=1}^{T-1} y_t x_t \right)' D_{w_{T-1}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-1}}^{\frac{1}{2}} \sum_{t=1}^{T} x_t x_t' D_{w_{T-1}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-1}}^{\frac{1}{2}} x_T$$

$$= \left(\sum_{t=1}^{T-1} y_t x_t \right)' D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T$$

$$- \left(\sum_{t=1}^{T-1} y_t x_t \right)' \frac{\left(D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T \right)}{1 + x_T' D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T}$$

$$\times \frac{\left(D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T \right)'}{1 + x_T' D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T} x_T$$

Noticing,

$$s_T = \left(\sum_{t=1}^{T-1} y_t x_t \right)' D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T$$

and by some simple algebraic manipulation, we obtain

$$\gamma_T = \frac{s_T}{1 + x_T' D_{w_{T-2}}^{\frac{1}{2}} \left(a\mathbb{I} + D_{w_{T-2}}^{\frac{1}{2}} \sum_{t=1}^{T-1} x_t x_t' D_{w_{T-2}}^{\frac{1}{2}} \right)^{-1} D_{w_{T-2}}^{\frac{1}{2}} x_T} \tag{34}$$

□

Remark 5. *Corollary 1 gives an expression that connects CIRR and OSLOG. It also gives the limiting behaviour of their predictions. As* $||x_T|| \to \infty$, $\gamma_T \to 0$ *whereas* $s_T \to \infty$. *The advantage of CIRR over OSLOG is that CIRR is less likely to overfit. AAR has similar advantage over ORR, but CIRR and AAR might be more likely to underfit.*

6. Empirical study

Having explained CIRR and OSLOG algorithms and their upper bound loss, in the following we perform experiments to investigate their performance. In particular, we check the performance against statistically optimal linear model. To illustrate better the discussion, we use two real-world data sets for which the model is linear in the parameters (coefficients/weights) and a third one which is synthetic.

Before delving into the details, we make some comments on the time complexity of the statistical and online models. The most computationally intensive calculation in the computation of the statistically optimal regression model

is the calculation of design matrix with its transpose, which can be done in $\mathcal{O}(n^2m)$ for an $n \times m$ matrix. The application of statistically optimal model to sequentially arriving data requires an addition of data point in $\mathbb{R}^n$ at each trial. Moreover, the statistically optimal model has no performance guarantee. In contrast, the most computationally intensive computation (by application of Sherman-Morrison formula) of the proposed online regression algorithms is $\mathcal{O}(n^2)$ and they do not need to row bind data points at each trial. Thus, they are efficient is terms of memory and time, while possessing a performance guarantee.

Linear regression assumes that the model is linear in parameters and does not contain influential outliers that affect the prediction quality. They also assume normality in residuals and homoscedasticity which is not about the predictive performance, but rather the correctness of the inference [1]. In our work, we consider datasets that present sparse adaptive regression problem.

We perform all experiments by splitting data in training (25%) and testing (75%). We tune the parameter a on training data by grid search, than fix the parameter a for testing. Both training and testing is done in online mode. The explanation given under synthetic data heading will discusses the process of training and testing in more detail. The code of the mentioned algorithms is part of the online machine learning library SOLMA[2]. To reproduce the results, please see Github link[3].

Synthetic data

We perform experiments to illustrate the usefulness of the proposed algorithms, OSLOG and CIRR. We use an adaptive artificial data set described in [48, 49]. The attributes $x_{t,1}, x_{t,2}, ..., x_{t,10}$ are generated independently and

[1] In [47] a comprehensive empirical study was conducted (using 42 datasets) showing that the linear methods (Least squares, LASSO, etc.) outperform many of the modern methods (Regression trees, ANN, etc.) on linear heteroscedastic data, despite the fact that homoscedasticity is an explicit assumption in the linear methods.

[2] `https://github.com/proteus-h2020/proteus-solma`

[3] `https://github.com/JamilWaqas/CR`

uniformly over $[0, 1]$ for 40768 data points. The value of the output y is:

$$y_t = 10\sin(\pi x_{t,1} x_{t,2}) + 20(x_{t,3} - 0.5)^2 + 10x_{t,4} + 5x_{t,5} + \epsilon_t \tag{35}$$

where $\epsilon_t \sim \mathcal{N}(0, 1)$. The Friedman function (35) is one of the most used function for data generation as it contains linear and non-linear relations between input and output (it is still linear in parameters). Also, only half of the attributes contribute towards the correlation with the output. We fit six online regression algorithms (CIRR, OSLOG, AAR, ORR, OGD, ONS) on the data and compare their prediction results against the Best Linear Unbiased Estimator (BLUE) $x_t' w^{BLUE}$ where:

$$w^{BLUE} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \tag{36}$$

Notice that the BLUE solution considers the entire data $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{Y} \in \mathbb{R}^n$, which is not possible when forecasting. Table 2 reports RMSE, R^2, MAE and error quantiles denoted by LQE (25%), MQE (50%) and UQE (75%) for 30576 data points. We use 25% (10192 data points) of the dataset to tune the parameter $a > 0$ require for initialisation of Algorithm 1 and 2. The tuning is done by performing grid search in the online mode. The length of the grid is selected arbitrarily, meaning that the selection is made by, for instance, assigning the size of the grid to 1 to 10 with 0.1 increment. If 1.5 comes out as the optimum value, then the grid's size is set to 0.1 to 1.5 with increment of 0.01 and so on. The value of a resulting in the least square loss on 25% of the data is fixed for the 75% of the data. Also, during the tuning stage, we observe that on the 10192th (final trial of 25%) trial CIRR, OSLOG and AAR give more importance to $x_{t,1}, x_{t,2}, x_{t,4}$ and $x_{t,5}$, and attributes weights are (approximately) $w_{10192,1} = 7, w_{10192,2} = 7, w_{10192,4} = 10$ and $w_{10192,5} = 5$. For this data ORR assigns very high weights, due to the outliers, the minimum weight ORR assigns is of 30004. We Observe RMSE of AAR, OSLOG and CIRR on the 25% of the data and notice that despite assigning similar weights, RMSE of AAR is much higher than CIRR and OSLOG. Theorem 3 and 5 indicates when $C \leq ||w||_1 \leq P$ and $C > 1$ CIRR and OSLOG outperform ORR and AAR. So, the CIRR, OSLOG and AAR correctly identify significant weights except for

the 3rd attribute weight. None of the attributes are set to 0. So, we do not remove any attributes to avoid the effect on the forecasting quality. Since ϵ in (35) and attributes $x_{t,3}, x_{t,6}, x_{t,7}, x_{t,8}, x_{t,9}$ and $x_{t,10}$ in (35) are in $[0, 1]$, they are somewhat helpful for the adjustment of the noise. Also, $x_{t,3}$ is part of the model see (35). However, if the computational efficiency is a priority, then less significant attributes can be removed, which will lead to inversion of 4×4 matrix at each trial instead of inverting a 10×10 matrix at each trial. By removal of less significant attributes, we obtain an RMSE of 3.41 for CIRR and 21.12 for OSLOG. The OGD and ONS perform a bit worse than CIRR and OSLOG when significant attributes are removed.

Table 2: Comparison of the algorithms using synthetic data.

Algorithm	**RMSE**	**R^2**	**MAE**	**LQE**	**MQE**	**UQE**
AAR	1.19×10^5	9.69×10^{-4}	1.02×10^5	−146305.37	−93971.93	−46701.88
ORR	4.73×10^5	1.69×10^{-2}	3.70×10^5	−518833.1	−303995	−145757.1
CIRR	2.65	0.72	2.04	−1.53	0.12	1.76
OSLOG	2.63	0.72	2.03	−1.56	0.09	1.73
OGD	2.91	0.67	2.26	−1.75	0.08	1.93
ONS	2.66	0.72	2.06	−1.55	0.11	1.77
BLUE	2.63	0.72	2.03	−1.56	0.10	1.75

In the above data we place a couple of anomalies. By doing so, the RMSE of OSLOG, OGD and ONS shoots up, while the RMSE of CIRR changes to 3.022, suggesting that CIRR is less sensitive to anomalies. For the above data L_1 and L_2 do not perform[4] as well as BLUE (linear model). We used R `glmnet`: Lasso and Elastic-Net Regularised Generalised Linear Models [51, 52] to fit L_1 and L_2 regression.

[4]Since we use the full data, adding regularisation leads to higher RMSE, R^2 and MAE. Regularisation helps reduce the over-fitting [50].

Real-world data

In order to further investigate the performance of the proposed algorithm, we use two real-world datasets: Istanbul Stock Exchange[5] (ISE) and Ailerons[6] ($F-16$) data. Both datasets present an adaptive regression prediction problem. The ISE data has 536 observations with 8 attributes: S&P 500 Index, Deutscher Aktien Index, FTSE 100 Index, Nikkel Index, Bovespa Index, Bovespa Index, MSCI Europe Index and MSCU Emerging Markets Index. The goal is to make the prediction of ISE in USD (which is the output variable). On the other hand, $F-16$ data consists of 13750 observations with a total of 40 attributes that describe the status of the $F-16$ and the goal is to predict control action on the ailerons of the $F-16$ data.

Table 3: Comparison of the algorithms using real-world data.

Algorithm	**RMSE**	**R^2**	**MAE**	**LQE**	**MQE**	**UQE**
Data: ISE						
OGD	0.19×10^{-1}	5.49×10^{-1}	0.14×10^{-1}	-8.40×10^{-3}	2.52×10^{-3}	1.24×10^{-2}
ONS	0.19×10^{-1}	5.50×10^{-1}	0.14×10^{-1}	-8.42×10^{-3}	2.50×10^{-3}	1.22×10^{-2}
AAR	1.87×10^{-2}	3.61×10^{-1}	1.39×10^{-2}	-8.46×10^{-3}	2.49×10^{-3}	1.18×10^{-2}
ORR	5.67×10^{-2}	1.69×10^{-1}	2.41×10^{-1}	-1.26×10^{-1}	5.57×10^{-4}	1.28×10^{-2}
CIRR	6.30×10^{-3}	9.00×10^{-1}	4.56×10^{-3}	-3.85×10^{-3}	2.95×10^{-4}	3.22×10^{-3}
OSLOG	1.05×10^{-2}	7.46×10^{-1}	4.44×10^{-3}	-3.72×10^{-3}	2.91×10^{-4}	2.80×10^{-3}
BLUE	4.81×10^{-3}	9.35×10^{-1}	3.74×10^{-3}	-3.11×10^{-2}	-5.55×10^{-6}	4.99×10^{-3}
Data: F$-$16						
OGD	5.32×10^{-4}	1.73×10^{-1}	3.84×10^{-4}	-4.96×10^{-4}	-2.33×10^{-4}	2.22×10^{-5}
ONS	2.73×10^{05}	2.48×10^{-4}	3.22×10^{04}	-1.05×10^{-3}	-1.52×10^{-3}	6.51×10^{-3}
AAR	7.61×10^{-1}	5.25×10^{-5}	2.71×10^{-1}	-9.12×10^{-2}	8.29×10^{-3}	1.05×10^{-1}
ORR	2.58×10^{07}	2.49×10^{-4}	2.00×10^{06}	-2.24×10^{04}	3.60×10^{03}	8.99×10^{04}
CIRR	1.97×10^{-4}	7.89×10^{-1}	1.41×10^{-4}	-8.57×10^{-5}	2.23×10^{-5}	1.16×10^{-5}
OSLOG	2.81×10^{00}	1.78×10^{-5}	3.80×10^{-2}	-7.35×10^{-5}	3.00×10^{-5}	1.23×10^{-4}
BLUE	1.69×10^{-4}	8.41×10^{-1}	1.25×10^{-4}	-9.08×10^{-5}	2.79×10^{-6}	9.64×10^{-5}

Table 3 compiles the results of the six algorithms when using the real-world data. For ISE during the tuning stage SLOG and CIRR set $w_{t,i} = 0$ for the

[5] `https://archive.ics.uci.edu/ml/datasets/ISTANBUL+STOCK+EXCHANGE`

[6] `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`

Nikkel Index, so we removed this attribute for CIRR and OSLOG. All algorithms, except CIRR, fail to perform well on the ISE data. In the case of F-16 data, OSLOG and CIRR assign $w_{t,1}, ..., w_{t,10} = 0$ to all attributes other than the first $1 - 10$ and 39th attributes. ORR and AAR assign near zero weight to 26th,30th, 32nd and 34th during the tuning phase, so we remove these attributes and run the algorithms on 75% of the data. OGD and ONS does not shrink any attributes weights close enough to zero.

In this experimental setting, CIRR and OSLOG select appropriate attributes during the tuning phase on these datasets, but on testing phase when the tuning parameter is fixed, OSLOG predictive ability is not as good as CIRR, meaning that OSLOG is more sensitive to the tuning parameter like AAR, ORR, ONS and OGD. Also based on these experiments, AAR, ORR, OGD and ONS's ability to perform shrinkage is no where near as impressive as CIRR and OSLOG's.

7. Conclusions

In this paper, we proposed two novel online algorithms, called CIRR and OSLOG. The theoretical analysis shows that CIRR and OSLOG have a better upper bound on the cumulative loss than AAR and ORR under certain circumstances and CIRR has a better upper bound on the cumulative loss than OSLOG under all circumstances. Moreover, the presented algorithms have similar type of bound as ONS (see Table 1), but without bounding the losses.

The empirical analysis indicates that OSLOG and CIRR have a different learning path and better computational efficiency in comparison to ORR, ONS, OGD and AAR. Furthermore, OSLOG and CIRR are able to perform model selection. In particular, CIRR is able to predict well under wider range of circumstances comparatively.

In the future, we will investigate the possibility of extending OSLOG and CIRR to the non-stationary case. From this piece of work, we were unable to find any circumstances where OSLOG can outperforms its competitors. However, it is difficult to make any conclusive remarks based on this work, a more insightful

theoretical and empirical study is required. Perhaps, empirically, one could try to answer the previous question by using different settings and different types of data sets, and theoretically studying lower bounds might be beneficial. The adjustment of the tuning parameter in online manner for these regression algorithms remains an important open question, along with the tightness of the upper loss bounds.

Acknowledgements

The European Commission supports Waqas Jamil and Abdelhamid Bouchachia under the Horizon 2020 Grant 687691 related to the project *PROTEUS: Scalable Online Machine Learning for Predictive Analytic and Real-Time Interactive Visualization*

References

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

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

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

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

[5] D. Golovin, B. McMahan, D. Sculley, Online learning with maximal no-regret l_1 regularization, in: NIPS workshop on Optimization for Machine Learning, 2016.

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

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

[8] N. Cesa-Bianchi, G. Lugosi, Prediction, learning, and games, Cambridge university press, 2006.

[9] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014) 44:1–44:37.

[10] R. Sambasivan, S. Das, S. K. Saha, A bayesian perspective of statistical machine learning for big data, arXiv preprint arXiv:1811.04788.

[11] A. Rakhlin, K. Sridharan, A. Tewari, Online learning via sequential complexities, The Journal of Machine Learning Research 16 (1) (2015) 155–186.

[12] T. Cover, A. Shenhar, Compound bayes predictors for sequences with apparent markov structure, IEEE Transactions on Systems, Man, and Cybernetics 7 (6) (1977) 421–424.

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

[14] V. Vovk, A game of prediction with expert advice, in: Proceedings of the eighth annual conference on Computational learning theory, ACM, 1995, pp. 51–60.

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

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

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

[18] A. R. Karlin, M. S. Manasse, L. Rudolph, D. D. Sleator, Competitive snoopy caching, Algorithmica 3 (1-4) (1988) 79–119.

[19] A. P. Dawid, The well-calibrated bayesian, Journal of the American Statistical Association 77 (379) (1982) 605–610.

[20] A. P. Dawid, Present position and potential developments: Some personal views: Statistical theory: The prequential approach, Journal of the Royal Statistical Society. Series A (General) (1984) 278–292.

[21] A. DeSantis, G. Markowsky, M. N. Wegman, Learning probabilistic prediction functions, in: Foundations of Computer Science, 1988., 29th Annual Symposium on, IEEE, 1988, pp. 110–119.

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

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

[24] V. Vovk, Aggregating strategies, in: Proc. Third Workshop on Computational Learning Theory, Morgan Kaufmann, 1990, pp. 371–383.

[25] K. Crammer, A. Kulesza, M. Dredze, Adaptive regularization of weight vectors, in: Advances in neural information processing systems, 2009, pp. 414–422.

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

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

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

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

[30] J. Forster, On relative loss bounds in generalized linear regression, in: Fundamentals of Computation Theory, Springer, 1999, pp. 829–829.

[31] K. S. Azoury, M. K. Warmuth, Relative loss bounds for on-line density estimation with the exponential family of distributions, Machine Learning 43 (3) (2001) 211–246.

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

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

[34] F. Orabona, N. Cesa-Bianchi, C. Gentile, Beyond logarithmic bounds in online learning, in: Artificial Intelligence and Statistics, 2012, pp. 823–831.

[35] R. P. Monti, C. Anagnostopoulos, G. Montana, A framework for adaptive regularization in streaming lasso models, arXiv preprint arXiv:1610.09127.

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

[37] K. Crammer, M. Dredze, F. Pereira, Exact convex confidence-weighted learning, in: Advances in Neural Information Processing Systems, 2009, pp. 345–352.

[38] N. Vaits, K. Crammer, Re-adapting the regularization of weights for non-stationary regression., in: ALT, Springer, 2011, pp. 114–128.

[39] Y. Kalnishkan, An upper bound for aggregating algorithm for regression with changing dependencies, in: International Conference on Algorithmic Learning Theory, Springer, 2016, pp. 238–252.

[40] S. Busuttil, Y. Kalnishkan, Online regression competitive with changing predictors, in: Algorithmic Learning Theory, Springer, 2007, pp. 181–195.

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

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

[43] P. Garrigues, L. E. Ghaoui, An homotopy algorithm for the lasso with online observations, in: Advances in neural information processing systems, 2009, pp. 489–496.

[44] M. Schmidt, Least squares optimization with l_1-norm regularization, CS542B Project Report (2005) 14–18.

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

[46] E. Beckenbach, R. Bellman, Inequalities, Springer-Verlag, 1961.

[47] S. J. Gelfand, Understanding the impact of heteroscedasticity on the predictive ability of modern regression methods, Master Thesis at Simon Fraser University.

[48] L. Breiman, Bagging predictors, Machine learning 24 (2) (1996) 123–140.

[49] J. H. Friedman, Multivariate adaptive regression splines, The annals of statistics (1991) 1–67.

[50] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Vol. 1, Springer series in statistics New York, 2001.

[51] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of Statistical Software 33 (1) (2010) 1–22.
URL http://www.jstatsoft.org/v33/i01/

[52] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for cox's proportional hazards model via coordinate descent, Journal of Statistical Software 39 (5) (2011) 1–13.
URL http://www.jstatsoft.org/v39/i05/