Mixability made efficient: Fast online multiclass logistic regression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Mixability has been shown to be a powerful tool to obtain algorithms with optimal regret. However, the resulting methods often suffer from high computational complexity which has reduced their practical applicability. For example, in the case of multiclass logistic regression, the aggregating forecaster (Foster et al. (2018)) achieves a regret of $O(\log(Bn))$ whereas Online Newton Step achieves $O(e^B \log(n))$ obtaining a double exponential gain in B (a bound on the norm of comparative functions). However, this high statistical performance is at the price of a prohibitive computational complexity $O(n^{37})$.

In this paper, we use quadratic surrogates to make aggregating forecasters more efficient. We show that the resulting algorithm has still high statistical performance for a large class of losses. In particular, we derive an algorithm for multi-class logistic regression with a regret bounded by $O(B\log(n))$ and a computational complexity of only $O(n^4)$.

1 Introduction

In online learning, a learner sequentially interacts with an environment and tries to learn based on data observed on the fly (Cesa-Bianchi and Lugosi (2006), Hazan et al. (2016)). More formally, at each iteration $t \geq 1$, the learner receives an input x_t in some space $\mathcal{X}$; makes a prediction $\widehat{y}_t$ in a decision domain $\widehat{\mathcal{Y}}$ and the environment reveals the output $y_t \in \mathcal{Y}$. The inputs x_t and the outputs y_t are sequentially chosen by the environment and can be arbitrary. No stochastic assumption (except boundedness) on the data sequence $(x_t, y_t)_{1\leq t\leq n}$ is made. The accuracy of a prediction $\widehat{y}_t \in \widehat{\mathcal{Y}}$ at instant $t \geq 1$ for the outcome $y_t \in \mathcal{Y}$ is measured through a loss function $\ell : \widehat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}$. The learner aims at minimizing his cumulative regret

$$R_n(f) = \sum_{t=1}^{n} \ell\big(\widehat{y}_t, y_t\big) - \sum_{t=1}^{n} \ell\big(f(x_t), y_t\big)\,, \tag{1}$$

with respect to any function $f \in \mathcal{F}$ in a reference class. All along this paper, we will consider parametric class of functions f_θ indexed by $\theta \in \Theta \subset \mathbb{R}^d$. We will also assume convexity of the loss according to index θ, which we denote $\ell_{x,y} : \theta \mapsto \ell(f_\theta(x), y)$ for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$. In this general context, many algorithms have been designed based on different assumptions and obtaining different trade-offs for regret. We review below the most relevant ones for our purpose.

Assuming only convexity of the loss ℓ, a well-known strategy for the learner is Online Gradient Descent (Zinkevich (2003)) with an optimal regret of order $O(\sqrt{n})$. However, if the loss is α-mixable, the learner may achieve the faster rate $O(\frac{1}{\alpha}\log(n))$ by using an aggregating forecaster (see Vovk (2001), Van Erven et al. (2015)). Yet, such an algorithm is not constructive in general and when it is, the computational complexity is often very high (a notable exception is the least-squares setting). To

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Algorithm	OGD	ONS	Foster et al. (2018)	**GAF** (Ours)
Regret	$B\sqrt{n}$	$dKe^B \log(n)$	$dK \log(Bn)$	$dK(B^2 + B\log(n))$
Total complexity	ndK	nd^3K^3	$B^6n^{25}(Bn + dK)^{12}$	$nK^3d^2 + K^2n^4$

Table 1: Regret bounds and computational complexities (in $O(\cdot)$) of relevant algorithms for logistic regression with K classes.

reduce it, a stronger assumption has been introduced by Hazan et al. (2007) with β-exp concavity. Under the latter hypothesis, Online Newton Step (ONS) has a regret bounded by $O(\frac{1}{\beta}\log(n))$ and a computational complexity of $O(n)$. However this efficiency comes at the price of deteriorating statistical performance. Indeed, β-exp concavity implies α-mixability for $\beta \geq \alpha$ and in some cases, the gap between β and α can be very large.

The most spectacular case of this phenomenon occurs for logistic regression. In this setting, the loss is defined as

$$\ell(\widehat{y}_t, y_t) = -\log(\sigma(\widehat{y}_t)_{y_t}) \qquad \text{where} \qquad \sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

and the regret is computed with respect to linear functions $\mathcal{F} = \{x \mapsto \theta^\top \Phi(x), \theta \in \Theta\}$, where $\Theta = \mathcal{B}(\mathbb{R}^d, B)$ is the ℓ_2 bounded ball of radius $B > 0$. On this subset, the logistic loss is β-exp concave *only* for $\beta \leq e^{-B}$. The regret of Online Newton Step can thus be of order $O(e^B \log(n))$. On the other hand, as remarked by Foster et al. (2018), the logistic loss is 1-mixable and an aggregating algorithm may achieve $O(\log(nB))$. Nevertheless, the algorithm relies on Monte Carlo methods and must sample from log-concave probability distributions which is extremely computationally expensive (see Table 1). Therefore, in this framework, the exp-concavity assumption leads to an efficient algorithm with low dependence on B while mixability yields an inefficient algorithm with much better statistical performance.

Between these two extremes, other algorithms and trade-offs have been analyzed. First, it has been shown that in some situations, Follow The Regularized Leader (FTRL) can achieve a fast rate without an exponential constant (Bach et al. (2010), Marteau-Ferey et al. (2019), Ostrovskii et al. (2021)). However, several additional assumptions are necessary to achieve these rates. Unfortunately, they are essentially unavoidable. Indeed, Hazan et al. (2014) showed a polynomial lower bound for the proper algorithm (*i.e.,* with linear prediction function) in the regime $B = \log(n)$. This prevents algorithms like ONS and Follow The Regularized Leader from reaching logarithmic regret without an exponential constant in B. This result motivates the search for improper algorithms with better computational complexity than Foster et al. (2018).

For binary logistic regression, two efficient improper algorithms have been proposed in the literature. First, in the i.i.d. framework, Mourtada and Gaïffas (2019) have proposed the Sample Minmax Predictor (SMP). It achieves an excess risk of order $O((B^2 + d)/n)$ with computational complexity equivalent to Follow The Regularized leader. In the online framework, Jézéquel et al. (2020) proposed AIOLI, which is based on quadratic approximations of the logistic loss as well as virtual labels to regularize. The regret is upper-bounded $O(dB\log(n))$ and the computational complexity is $O(n(d^2 + \log(n)))$.

These previous works left open the question of achieving the same type of performance in a setting other than binary logistic regression. In particular, for multi-class logistic regression, no other algorithm than Foster et al. (2018) is known to achieve logarithmic regret without an exponential constant.

Contributions We introduce in Section 3 a new generic online learning algorithm, that we call Gaussian Aggregating Forecaster (GAF) . GAF achieves logarithmic regret with a small multiplicative constant for a large class of convex loss functions (see Theorem 1). The latter includes several popular loss functions such as squared loss, binary and multi-class logistic loss. Our assumptions on the loss functions are slightly stronger than α-mixability but generally weaker than assumptions widely used in the statistical framework such as generalized self-concordance (see Bach et al. (2010)).

In the particular but significant setting of multi-class logistic regression, GAF has a regret bounded by $O(dKB^2 + dKB\log(n))$ and a total computational complexity of $O(nK^3d^2 + K^2n^4)$, thus

significantly improving on the $O(B^6 n^{25}(Bn + dK)^{12})$ complexity of Foster et al. (2018), which was the best known to date for algorithms without exponential dependence on B. Table 1 summarizes the regrets and computational cost obtained by the relevant algorithms in this framework. It is worth pointing out that, by using standard online-to-batch conversion Helmbold and Warmuth (1995), this paper also provides new results on the excess risk in the statistical i.i.d. Indeed, even in this extensively used framework, the only existing improper algorithms without exponential constant in B so far are GAF and Foster et al. (2018). Jézéquel et al. (2020) and Mourtada and Gaïffas (2019) are in fact restricted to binary outputs only.

Our new algorithm is inspired by the aggregating forecaster of Vovk (2001) that we cleverly apply to quadratic approximations of the loss, in order to make it more efficient. The high-level idea is that these quadratic approximations replace complicated log-concave distributions with Gaussian distributions, from which it is much easier to sample and thereby drastically reducing the complexity of the algorithm. We believe that this approach can be applied to much broader contexts than those analyzed here.

2 Setting

We recall here the setting and introduce the main notations and assumptions that will be used throughout the paper.

Setting and notation Our framework is formalized as a sequential game between a learner and an environment. At each forecasting instance $t \geq 1$, the learner is given an input $x_t \in \mathcal{X}$; forms a prediction $\widehat{y}_t \in \widehat{\mathcal{Y}} \subseteq \mathbb{R}^K$ (possibly based on the current input x_t and on the past information $x_1, y_1, \ldots, x_{t-1}, y_{t-1}$). Note that the prediction space $\widehat{\mathcal{Y}} \subseteq \mathbb{R}^K$ may be uni-dimensional ($K = 1$) in some settings (least-square regression) or multi-dimensional ($K \geq 1$) in some cases (e.g., vector-valued regression, multi-class classification). Then, the environment chooses $y_t \in \mathcal{Y}$; reveals it to the learner which incurs the loss $\ell(\widehat{y}_t, y_t)$. Finally, the performance of the learner is assessed by the regret

$$R_n(\theta) = \sum_{t=1}^{n} \ell(\widehat{y}_t, y_t) - \sum_{t=1}^{n} \ell_{x_t, y_t}(\theta)\,,$$

with respect to all $\theta \in \Theta \subset \mathbb{R}^d$. Here, $\ell_{x,y}(\theta) = \ell(f_\theta(x), y)$ where $\mathcal{F} = \{f_\theta : \mathcal{X} \to \mathcal{Y}\}$ is the reference class of functions. For simplicity of notation, we also define $\ell_t(\cdot) = \ell_{x_t, y_t}(\cdot)$ for all $t \geq 1$.

In all specific examples considered in this work, the learner will be compared to linear functions $f_\theta : x \mapsto \theta^\top \Phi(x)$, where $\theta \in \Theta = \mathcal{B}(\mathbb{R}^d, B)$ and Φ is a function from $\mathcal{X}$ to $\mathbb{R}^{d \times K}$ such that for any $x \in \mathcal{X}$ and $i \in [K]$, $\|\Phi(x)_{\cdot,i}\| \leq R$. We introduce and discuss below our main assumptions on the losses.

Assumptions We assume that, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the losses $\ell_{x,y}$ are convex, $\mathcal{C}^2$ and satisfy the following assumptions.

(A1) The loss function ℓ is α-mixable. In other words, for all (Gaussian) probability distributions π over $\mathbb{R}^d$ and input $x \in \mathcal{X}$, there exists $\widehat{y} \in \mathbb{R}^K$ such that for all $y \in \mathcal{Y}$,

$$\ell(\widehat{y}, y) \leq -\frac{1}{\alpha} \log \left(\mathbb{E}_{\theta \sim \pi} e^{-\alpha \ell_{x,y}(\theta)} \right)\,.$$

(A2) The exists $\zeta > 0$ such that, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $\theta_1, \theta_2 \in \mathbb{R}^d$,

$$\ell_{x,y}(\theta_1) \leq \ell_{x,y}(\theta_2) + \nabla \ell_{x,y}(\theta_2)^\top (\theta_1 - \theta_2) + e^{\zeta \|\theta_1 - \theta_2\|^2} \|\theta_1 - \theta_2\|^2_{\nabla^2 \ell_{x,y}(\theta_2)}$$

(A3) There exists $\beta > 0$ such that, for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $(\theta_1, \theta_2) \in \Theta \times \mathbb{R}^d$,

$$\ell_{x,y}(\theta_1) \geq \ell_{x,y}(\theta_2) + \nabla \ell_{x,y}(\theta_2)^\top (\theta_1 - \theta_2) + \frac{\beta}{2} (\theta_1 - \theta_2)^\top \nabla^2 \ell_{x,y}(\theta_2)(\theta_1 - \theta_2)$$

(A4) ℓ is γ-smooth *i.e.,* for all $x \in \mathcal{X}, y \in \mathcal{Y}, \theta \in \mathbb{R}^d, \nabla^2 \ell_{x,y}(\theta) \leq \gamma I$.

In general, these assumptions are rather weak and related to standard assumptions of (online) convex optimization. (A1) follows from instance from exp-concavity. As we illustrate in Section 4, all are

satisfied for broadly used loss functions such as squared loss, binary and multi-class logistic loss. We provide more details on these assumptions below.

Mixability (A1) is a standard assumption in online learning to achieve fast rates for the regret (Van Erven et al., 2015), which was introduced by Vovk (2001). It is a weaker condition than exp-concavity because (A1) holds with $\widehat{y} = \mathbb{E}_{\theta\sim\pi}[\theta]$ for α-exp concave loss functions.

Assumption (A2) basically prevents the Hessian from changing too quickly on θ. This condition is new but not very restrictive because, as we prove in Lemma 6, it is weaker than other widely accepted assumptions such as generalized self-concordance (Bach et al., 2010).

The lower bound (A3) is the most original assumptions and is crucial to make aggregating forecaster efficient. This bound is close to the one needed in the analyses of ONS, which can be derived from β-exp-concavity. However, the key difference is that (A3) uses the Hessian in the quadratic term and not the outer product of the gradient. This makes the right-hand-side closer to the third Taylor series approximation of the loss and allows much larger values for β. For example, as we will show in the section 4.2, this change allows us to remove the exponential constant for the logistic regression setting. In this case, (A3) is verified with $\beta \simeq B^{-1}$, while β-exp-concavity would require $\beta \simeq e^{-B}$.

3 Algorithm and regret bound

GAF is a new sequential forecasting rule inspired by the aggregating forecaster of Vovk (2001). It may be implemented if the loss function ℓ is mixable (A1) and ensures fast performance guarantees on the regret under Assumptions (A1)-(A4) as shown in Theorem 1. GAF requires the following hyper-parameters: a regularization parameter $\lambda > 0$ and $\alpha, \beta > 0$ such that the loss ℓ satisfies assumptions (A1)-(A4).

Algorithm (GAF) At each time step, the prediction $\widehat{y}_t$ is formed by GAF by following a two steps procedure. A first estimator $\theta_t \in \mathbb{R}^d$ is computed by solving

$$\theta_t = \operatorname*{argmin}_{\theta\in\mathbb{R}^d} \left\{ \sum_{s=1}^{t-2} \tilde{\ell}_s(\theta) + \ell_{t-1}(\theta) + \lambda\|\theta\|_2^2 \right\}. \tag{2}$$

Basically, θ_t follows a regularized leader where the losses ℓ_s for $s \leq t-2$ are substituted with quadratic approximations $\tilde{\ell}_s$. Then, GAF computes the quadratic approximation of ℓ_{t-1} at point θ_t. For any $\theta \in \mathbb{R}^d$, it defines

$$\tilde{\ell}_{t-1}(\theta) = \ell_{t-1}(\theta_t) + \nabla\ell_{t-1}(\theta_t)^\top(\theta-\theta_t) + \frac{\beta}{2}(\theta-\theta_t)^\top\nabla^2\ell_{t-1}(\theta_t)(\theta-\theta_t). \tag{3}$$

Finally, GAF predicts $\widehat{y}_t \in \widehat{\mathcal{Y}}$ such that for all $y \in \mathcal{Y}$,

$$\ell(\widehat{y}_t, y) \leq -\frac{1}{\alpha}\log\left(\mathbb{E}_{\theta\sim\tilde{P}_{t-1}} e^{-\alpha\ell_{x_t,y}(\theta)}\right), \qquad \text{where} \qquad \tilde{P}_{t-1}(\theta) = \frac{e^{-\alpha\tilde{L}_{t-1}(\theta)}}{\int_{\mathbb{R}^d} e^{-\alpha\tilde{L}_{t-1}(\theta)}d\theta}, \tag{4}$$

with $\tilde{L}_{t-1}(\theta) = \sum_{s=1}^{t-1}\tilde{\ell}_s(\theta) + \lambda I$. Such a prediction $\widehat{y}_t$ exists as soon as Assumption (A1) is true. In Section 4, we present some specific cases where $\widehat{y}_t$ can be computed in a closed form.

We now state our main theoretical result, which is an upper bound on the regret suffered by GAF.

Theorem 1. *Let $d, n \geq 1$, $B > 0$, and $\Theta \subset \mathcal{B}(\mathbb{R}^d, B)$. Let $(x_1, y_1), ..., (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ be an arbitrary sequence of observations and ℓ a loss function that verifies Assumptions (A1)-(A4) with $\alpha, \zeta, \beta, \gamma > 0$. GAF* (4)*, run with regularization parameter $\lambda \geq 8\zeta$, satisfies the following upper-bound on the regret*

$$R_n(\theta) \leq \lambda\|\theta\|^2 + d\left[\frac{1}{2\alpha} + \frac{\sqrt{3}2^{\frac{d}{2\lambda}+1}}{\beta}\right]\log\left(1 + \frac{n\beta\gamma}{\lambda}\right), \qquad \forall\theta \in \Theta.$$

In particular, the choice $\lambda = 8d\zeta$ yields

$$R_n(\theta) \leq 8d\zeta B^2 + d\left[\frac{1}{2\alpha} + \frac{4}{\beta}\right]\log\left(1 + \frac{n\beta\gamma}{d\zeta}\right), \qquad \forall\theta \in \Theta.$$

Theorem 1 states that GAF has a logarithmic regret in the number of samples with a multiplicative constant proportional to $\alpha^{-1} + \beta^{-1}$. Recalling that α is the mixability parameter, α^{-1} is the optimal multiplicative constant that can be obtained using an (computationally expensive) aggregating forecaster. The constant β is the curvature parameter in the quadratic lower-bound assumption (A3). Having β^{-1} as a multiplicative constant can be seen as similar to the regret bound of ONS. However, it is important to note the crucial difference that, unlike ONS, it is the Hessian that is used in the quadratic substitutes of Assumption (A3) and not the gradient outer product. Thus, in some cases, it is possible to have a much larger value for β, than would be attainable using the exp-concavity. For example, for multiclass logistic regression, the loss is 1-mixable (Proposition 1, Foster et al., 2018) but only e^{-B}-exp concave. We prove in Lemma 4 that this loss verifies Assumption (A3) with parameter $\beta \simeq (\log K + BR)^{-1}$. Therefore, GAF achieves, in the logistic case, a regret upper bounded by $O(dB^2R^2 + d(\log(K) + BR)\log(n))$. More details on this specific case are given in Section 4.2.

We provide below a sketch of the proof of Theorem 1 that highlights the key steps in the proof. The full proof is available in Appendix A.

Sketch of proof. The proof starts from the definition (4) of the prediction $\widehat{y}_t$ formed by the algorithm. It satisfies the mixability assumption (A1) for all $y \in \mathcal{Y}$

$$\ell(\widehat{y}_t, y) \leq -\frac{1}{\alpha}\log\left(\mathbb{E}_{\theta\sim\tilde{P}_{t-1}(\theta)} e^{-\alpha\ell_t(\theta)}\right), \quad \text{where} \quad \tilde{P}_{t-1}(\theta) = \frac{e^{-\alpha\tilde{L}_{t-1}(\theta)}}{\int_{\mathbb{R}^d} e^{-\alpha\tilde{L}_{t-1}(\theta)}d\theta}.$$

Next, we decompose the term on the right side of the inequality into two terms: an approximation error that will be small for well-chosen parameters; and a term, which telescopes and is standard for aggregating forecaster's analysis (Vovk, 2001), except that here it is written with quadratic surrogate losses. We have,

$$\begin{aligned}\ell(\widehat{y}_t, y_t) &\leq -\frac{1}{\alpha}\log\left(\frac{\int_{\mathbb{R}^d} e^{-\alpha(\ell_t(\theta)-\tilde{\ell}_t(\theta))-\alpha\tilde{L}_t(\theta)}d\theta}{\int_{\mathbb{R}^d} e^{-\alpha\tilde{L}_{t-1}(\theta)}d\theta}\right)\\ &= \underbrace{-\frac{1}{\alpha}\log\left(\mathbb{E}_{\theta\sim\tilde{P}_t} e^{-\alpha[\ell_t(\theta)-\tilde{\ell}_t(\theta)]}\right)}_{\Omega_t} + \underbrace{\frac{1}{\alpha}\log\left(\frac{\int_{\mathbb{R}^d} e^{-\alpha\tilde{L}_{t-1}(\theta)}d\theta}{\int_{\mathbb{R}^d} e^{-\alpha\tilde{L}_t(\theta)}d\theta}\right)}_{\Psi_t}.\end{aligned}$$

Now, the core idea is that since $\tilde{L}_t$ is quadratic, the integrals inside Ψ_t are Gaussian, which allows closed form formulas for the analysis (and fast computational time). Using that $\nabla\tilde{\ell}_t(\theta_{t+1}) = \nabla\ell_t(\theta_{t+1})$ and the expression of Gaussian integrals, it is possible to show that

$$\Psi_t = \tilde{L}_t(\theta_{t+1}) - \tilde{L}_{t-1}(\theta_t) + \frac{1}{2\alpha}\log\left(\frac{|A_t|}{|A_{t-1}|}\right),$$

where A_t is the Hessian of $\tilde{L}_t/2$. Therefore, summing over $t = 1, \ldots, n$ and using $\tilde{L}_0(\theta_1) = 0$, it yields

$$\sum_{t=1}^{n} \ell(\widehat{y}_t, y_t) - \tilde{L}_n(\theta_{n+1}) \leq \sum_{t=1}^{n} \Omega_t + \frac{1}{2\alpha}\log\left(\frac{|A_n|}{|A_0|}\right).$$

But, from Assumption (A3) together with the definition (2) of θ_{t+1}, for any $\theta \in \Theta$, $L_t(\theta) \geq \tilde{L}_t(\theta) \geq \tilde{L}_t(\theta_{t+1})$. Thus, the regret can be bounded for any $\theta \in \Theta$ by

$$R_n(\theta) = \sum_{t=1}^{n} \ell(\widehat{y}_t, y_t) - L_n(\theta) \leq \lambda\|\theta\|^2 + \sum_{t=1}^{n} \Omega_t + \frac{1}{2\alpha}\log\left(\frac{|A_n|}{|\lambda I|}\right). \tag{5}$$

Finally, Assumption (A2) allows to bound the approximation error Ω_t by a telescopic term close to the usual one but with $\frac{1}{\beta}$ as multiplicative constant instead of $\frac{1}{\alpha}$,

$$\Omega_t \leq \frac{C}{\beta}\log\left(\frac{|A_t|}{|A_{t-1}|}\right).$$

The proof is concluded by applying Assumption (A4). □

4 Specific settings: squared loss and multi-class logistic

In this section we show how our general framework cover several interesting settings. In particular, we prove that our Assumption (A1)-(A4) are satisfied and provide concrete implementations of GAF in those contexts.

4.1 Squared loss

For linear regression with squared loss, we recover classical results by Vovk (2001). This framework is defined, as a special case of our generic setting of Section 2, by setting: input domain $\mathcal{X} = \mathcal{B}(\mathbb{R}^d, R)$, output domain $\mathcal{Y} = [-Y, Y]$ with $Y > 0$, decision domain $\widehat{\mathcal{Y}} = \mathbb{R}$, loss function $\ell(\widehat{y}_t, y_t) = (\widehat{y}_t - y_t)^2$ and input feature $\Phi(x) = x$. Now we can verify that all assumptions are true:

- (A1) By Lemma 2 and Lemma 3 of Vovk (2001), the squared loss is mixable with parameter $\alpha = 1/(2Y^2)$.
- (A2) It suffices to note that the inequality is an equality for $\zeta = 0$. However, we supposed $\zeta > 0$ so we will take $\zeta = (8dB^2)^{-1}$.
- (A3) It holds with $\beta = 1$ because the Taylor expansion of order 2 of a quadratic function is an equality.
- (A4) It is true with $\gamma = R^2$ since $\nabla^2 \ell_{x,y}(\theta) = 2xx^\top \leq 2R^2 I$.

In particular, in this particular case the quadratic approximations are exact $\tilde{\ell}_t = \ell_t$ for all $t \geq 1$. Therefore, GAF reduces to the classical non-linear Ridge regression of Vovk (2001) and Azoury and Warmuth (2001) that predicts

$$\widehat{y}_t = \widehat{\theta}_t^\top x_t \quad \text{with} \quad \widehat{\theta}_t = \operatorname{argmin}_{\theta \in \mathbb{R}^d} \left\{ L_{t-1}(\theta) + \theta^\top x_t + \lambda \|\theta\|^2 \right\}.$$

A direct consequence of Theorem 1, by substituting the constants derived above, yields

$$R_n(\theta) \leq 1 + d(Y^2 + 4) \log\left(1 + 16nB^2R^2\right).$$

This result essentially recovers the existing regret bound for the non-linear Ridge regression (see e.g., Theorem 11.8 of Cesa-Bianchi and Lugosi (2006)).

4.2 Logistic regression

The most interesting application of our general framework is logistic regression. In this section, we show the validity of our assumptions and the concrete implementation of GAF in this context. We present the theoretical results obtained and discuss the computational complexity.

The goal of logistic regression is to form a K-dimensional prediction $\widehat{y} \in \widehat{\mathcal{Y}} = \mathbb{R}^K$ of a categorical label $y \in \mathcal{Y} = \{1, ..., K\}$ from the observation of an input label $x \in \mathcal{X} = \mathcal{B}(\mathbb{R}^{d'}, R)$, $d' \geq 1$. The performance of $\widehat{y}$ is measured by the logistic loss defined by

$$\ell(\widehat{y}, y) = -\log(\sigma(\widehat{y})_y) \qquad \text{where} \quad \sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}.$$

Defining the input feature and the linear predictions respectively as

$$\Phi(x)_{i,j} = \begin{cases} x_{i-d'j} & \text{if } d'j \leq i < d'(j+1) \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \quad f_\theta(x) = \theta^\top \Phi(x), \quad \text{for } \theta \in B(\mathbb{R}^d, B),$$

one can check that our setting recovers the standard multi-class logistic regression setting (see Foster et al. (2018) for an equivalent convention). We check below Assumptions (A1)–(A4).

- (A1) It holds with $\alpha = 1$, since by Proposition 1 of Foster et al. (2018), the logistic loss is 1-mixable. Indeed, given a distribution π on $\mathbb{R}^K$, the choice $\widehat{y}_\pi = \sigma^+(\mathbb{E}_{\widehat{y} \sim \pi} \sigma(\widehat{y}))$, where $\sigma^+(z)_k := \log(z_k)$, satisfies

$$\mathbb{E}_{\widehat{y} \sim \pi} \exp(-\ell(\widehat{y}, y)) = \mathbb{E}_{\widehat{y} \sim \pi} \sigma(\widehat{y})_y = \sigma(\widehat{y}_\pi)_y = \exp(-\ell(\widehat{y}_\pi, y)),$$

  for any $y \in [K]$.
- (A2) It is true for $\zeta = 4R^2$ by Lemma 6 in Appendix B.
- (A3) It holds with $\beta = (\log(K)/2 + BR + 1)^{-1}$ by Lemma 4 in Appendix B applied with the choices $a = \theta_1^\top \Phi(x)$ and $b = \theta_2^\top \Phi(x) \in [-BR, BR]^K$.
- (A4) It is valid with $\gamma = R^2$. Indeed, $\nabla^2 \ell_{x,y}(\theta) = \Phi(x)(\operatorname{diag}(p) - pp^\top)\Phi(x)^\top \leq \Phi(x)\Phi(x)^\top \leq R^2 I$, where $p = \sigma(\theta^\top \Phi(x))$.

Parameters $\lambda, \mu, \beta, \varepsilon, \delta > 0, m \geq 1$
initialize $A_0 = \lambda I, b_0 = 0, \theta_1 = 0$
for $t = 1, ..., n$ **do**
  receive $x_t \in \mathbb{R}^d$
  sample independently $\omega_1, ..., \omega_m \sim \mathcal{N}(\theta_t^\top \Phi(x_t), \Phi(x_t)^\top A_{t-1}^{-1} \Phi(x_t))$
  predict $\tilde{y}_t = \sigma^+(\mathrm{smooth}_\mu(\frac{1}{m}\sum_{i=1}^m \sigma(\omega_i)))$ where $\sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$, $\sigma^+(z)_i = \log(z_i)$ and $\mathrm{smooth}_\mu(z) := (1-\mu)z + \mu \mathbf{1}/K$.
  receive $y_t \in \{1, ..., K\}$
  update $\theta_{t+1} = \mathrm{argmin}_{\theta \in \mathbb{R}^{Kd}} b_{t-1}^\top \theta + \theta^\top A_{t-1} \theta + \ell_{x_t, y_t}(\theta)$
    $b_t = b_{t-1} + \nabla \ell_{x_t, y_t}(\theta_{t+1}) - \beta \nabla^2 \ell_{x_t, y_t}(\theta_{t+1}) \theta_t$
    $A_t = A_{t-1} + \frac{\beta}{2} \nabla^2 \ell_{x_t, y_t}(\theta_{t+1})$
end

Algorithm 1: Efficient-GAF for K-class logistic regression

In particular, the proof of the mixability assumption (A1) provides us a concrete expression for Equation (4) of GAF by setting

$$\widehat{y}_t = \sigma^+(\mathbb{E}_{\theta \sim \tilde{P}_{t-1}(\theta)}(\sigma(\theta^\top x_t))) \,. \tag{6}$$

Substituting the above parameters into Theorem 1 yields the following corollary.

Corollary 2. *Let $R, B > 0$ and $d', n, K \geq 1$. Let $(x_1, y_1), ..., (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ be an arbitrary sequence of observations and $\ell(\widehat{y}, y) = -\log(\sigma(\widehat{y})_y)$. GAF, run with $\lambda = 32 d' K' R^2$, $\beta = (\log(K)/2 + BR + 1)^{-1}$, and $\alpha = 1$, satisfies*

$$R_n(\theta) \lesssim d' K B^2 R^2 + d' K \left[\log(K) + BR\right] \log\left(1 + \tfrac{n}{d' K (\log(K) + BR)}\right), \tag{7}$$

for all $\theta \in \mathcal{B}(\mathbb{R}^d, B)$, where $\lesssim$ denotes an approximate inequality which is up to universal multiplicative and additive constants.

Computationally efficient approximation Yet, a key difficulty remains to compute exactly $\widehat{y}_t$ in Equation (6): the calculation of the expectation. In general, there is no closed-form expression and an approximation algorithm, such as Monte Carlo sampling, must be used. We provide now a fully implementable approximated version of GAF, which satisfies the same regret guarantees up to negligible additive constants. It is described in Algorithm 1.

First, to ensure that our approximation of the expectation produces forecasts close to those of the exact algorithm, we must smooth the function σ^+ in Equation (6). Following the idea of Foster et al. (2018), we thus define for some $\mu \in [0, \frac{1}{2}]$, the smoothing operator $\mathrm{smooth}_\mu : \Delta_K \to \Delta_K$ by

$$\mathrm{smooth}_\mu(p) = (1-\mu)p + \mu \mathbf{1}/K, \qquad \forall p \in \Delta_k \,.$$

Now, we can show that $\sigma^+(\mathrm{smooth}_\mu(\cdot))$ is μ-Lispchitz. Denoting the smoothed approximation of $\widehat{y}_t$ by $\bar{y}_t = \sigma^+(\mathrm{smooth}_\mu(\mathbb{E}_{\theta \sim \tilde{P}_{t-1}}(\sigma(\theta x_t)))$, Lemma 16 of Foster et al. (2018) shows that using $\bar{y}_t$ instead of $\widehat{y}_t$ worsens the regret bound only by an additional constant μn; hence our choice $\mu = n^{-1}$. The last step of the approximated algorithm consists in using Monte Carlo sampling to approximate the expectation by a finite sum

$$\tilde{y}_t = \sigma^+\left(\mathrm{smooth}_\mu\left(\frac{1}{m}\sum_{i=1}^m \sigma(\omega_i)\right)\right) \qquad \text{where} \quad \omega_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\theta_t^\top \Phi(x_t), \Phi(x_t)^\top A_{t-1}^{-1} \Phi(x_t)) \,.$$

Using Chernoff's inequality, we show below that $\tilde{y}_t$ concentrates well to $\bar{y}_t$ which entails the following guarantee.

Proposition 3. *Let $\delta > 0$. Efficient-GAF, run with parameters $\lambda = 32 d' K' R^2$, $\beta = (\log(K)/2 + BR + 1)^{-1}$, $m \geq 1$, and $\mu = (\log(n/\delta)/m)^{1/3}$ satisfies with probability $1 - \delta$ the regret bound*

$$R_n(\theta) \lesssim d' K B^2 R^2 + d' K \left[\log(K) + BR\right] \log\left(1 + \frac{n}{d' K (\log(K) + BR)}\right) + n \left(\frac{K}{m} \log\left(\frac{n}{\delta}\right)\right)^{\frac{1}{3}}$$

with a computational cost $O(nK^3 d^2 + K^2 n m)$. In particular, the choice $m = n^3 \log(n/\delta)$, yields the regret bound (7) with probability $1 - \delta$ and a total computational time of order $O\big(nK^3 d^2 + K^2 n^4 \log(n/\delta)\big)$.

Proof. We first prove the computational complexity upper-bound. At each iteration $t = 1, \dots, n$, the algorithm performs the following computations:

(i) Update A_t^{-1}: since the rank of $A_t - A_{t-1}$ is K, the inverse A_t^{-1} can be updated in $O(K^3 d^2)$ operations.

(ii) Update θ_{t+1}: by Theorem 4 of Jézéquel et al. (2020), this can be done in $O(\log(n) + K^2 d^2)$.

(iii) Compute $\tilde{y}_t$: to do so, it must sample m times from a Gaussian of dimension K. The cost is thus $O(mK^2)$.

Therefore, the overall time complexity of the algorithm is $O\big(nK^3 d^2 + K^2 nm\big)$.

We now prove the corresponding regret bound. First, we define

$$\tilde{p}_t := \mathrm{smooth}_\mu\left(\tfrac{1}{m}\textstyle\sum_{i=1}^m \sigma(\omega_i)\right) \qquad \text{and} \qquad \bar{p}_t := \mathbb{E}[\tilde{p}_t] = \mathrm{smooth}_\mu\left(\mathbb{E}_{\theta\sim\tilde{P}_{t-1}}(\sigma(\theta^\top \Phi(x_t))\right).$$

With this notation, $\tilde{y}_t = \sigma^+(\tilde{p}_t)$ and $\bar{y}_t = \sigma^+(\bar{p}_t)$. Using that $\sigma(\sigma^+(p)) = p$ for any $p \in \Delta_K$ yields for all $y_t \in [K]$

$$\ell(\tilde{y}_t, y_t) = -\log\big(\sigma(\tilde{y}_t)_{y_t}\big) = -\log\big(\sigma(\sigma^+(\tilde{p}_t))_{y_t}\big) = -\log(\tilde{p}_{t,y_t})\,,$$

where $\tilde{p}_{t,i}$ denotes the i-th component of $\tilde{p}_t$. Similarly, $\ell(\bar{y}_t, y_t) = -\log(\bar{p}_{t,y_t})$. Therefore, fixing some $\varepsilon > 0$, we have

$$\mathbb{P}\big[\ell(\tilde{y}_t, y_t) - \ell(\bar{y}_t, y_t) > \varepsilon\big] = \mathbb{P}\left[-\log\left(\tfrac{\tilde{p}_{t,y_t}}{\bar{p}_{t,y_t}}\right) > \varepsilon\right] = \mathbb{P}\left[\tilde{p}_{t,y_t} - \bar{p}_{t,y_t} < (e^{-\varepsilon} - 1)\bar{p}_{t,y_t}\right].$$

Using that $e^{-\varepsilon} - 1 \le 1 - 2\varepsilon$ for $\varepsilon \in (0, \frac{1}{2})$ together with the multiplicative Chernoff's bound (see e.g., Theorem 1.10.5 of Doerr (2020)), entails for all $0 < \varepsilon < \frac{1}{2}$,

$$\mathbb{P}\left[\ell(\tilde{y}_t, y_t) - \ell(\bar{y}_t, y_t) > \varepsilon\right] \le \mathbb{P}\left[\tilde{p}_{t,y_t} < (1 - 2\varepsilon)\bar{p}_{t,y_t}\right] \le e^{-2\varepsilon^2 m \bar{p}_{t,y_t}} \le e^{-\varepsilon^2 m\mu/K}\,, \tag{8}$$

where the last inequality is because $\mathrm{smooth}_\mu(p)_i \ge \mu/K$ for all $i \in [K]$. Taking $\varepsilon = \sqrt{\frac{K}{\mu m}\log(n/\delta)}$, we have

$$\mathbb{P}\left[\ell(\tilde{y}_t, y_t) - \ell(\bar{y}_t, y_t) > \sqrt{\tfrac{K}{\mu m}\log\left(\tfrac{n}{\delta}\right)}\right] \le \tfrac{\delta}{n}\,. \tag{9}$$

Furthermore, denoting $\widehat{p}_t = \mathbb{E}_{\theta\sim\tilde{P}_{t-1}}(\sigma(\theta^\top\Phi(x_t))$ so that $\bar{p}_t = \mathrm{smooth}_\mu(\widehat{p}_t)$ and following Lemma 16 of Foster et al. (2018), we get

$$\begin{aligned}\ell(\bar{y}_t, y_t) - \ell(\widehat{y}_t, y_t) &= \log\left(\frac{\widehat{p}_{t,y_t}}{\bar{p}_{t,y_t}}\right) = \log\left(\frac{\widehat{p}_{t,y_t}}{\mathrm{smooth}_\mu(\widehat{p}_t)_{y_t}}\right) = \log\left(\frac{\widehat{p}_{t,y_t}}{(1-\mu)\widehat{p}_{t,y_t} + \frac{\mu}{K}}\right)\\ &= \log\left(\frac{1}{1 - \mu\big(1 - \frac{1}{K\widehat{p}_{t,y_t}}\big)}\right) \le 2\mu\big(1 - \frac{1}{K\widehat{p}_{t,y_t}}\big) \le 2\mu\,. \end{aligned}\tag{10}$$

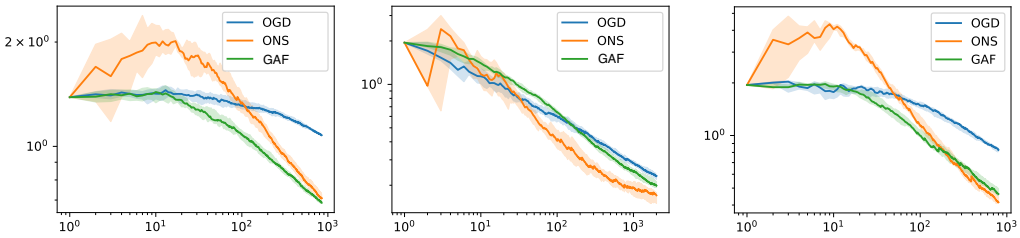
Combining (9) and (10), and using a union bound yields with probability at least $1 - \delta$

$$\textstyle\sum_{t=1}^n \ell(\tilde{y}_t, y_t) - \ell(\widehat{y}_t, y_t) \le n\sqrt{\frac{K}{\mu m}\log\left(\frac{n}{\delta}\right)} + \mu n\,.$$

Optimizing $\mu = (\log(n/\delta)K/m)^{1/3}$ and using Corollary 3 concludes the proof. □

Using Gaussian distribution improves significantly the computation time with respect to log-concave distribution as considered by Foster et al. (2018). Indeed, it allows to get exact and efficient samples from $\tilde{P}_{t-1}$ which leads to a computation time of order $O(n^4)$. On the contrary, since it is not possible to draw an exact sample from a log-concave distribution, Foster et al. (2018) must resort to expensive random walks to approximately sample from it. Using the method from Bubeck et al. (2018), as suggested by Foster et al. (2018), leads to a $O(B^6 n^{24}(Bn + dK)^{12})$ computation time per iteration (see their Example 3 with the choices $\varepsilon = n^{-2}$ and $\mu = n^{-1}$).

Note that our analysis shows a clean trade-off between the computational time and the regret bound. One could set a smaller value for m at the price of a larger regret. In particular, it is worth pointing out that our analysis consider the worst case scenario. In the experiments that we considered, it seems that much less samples are sufficient to reach a good accuracy. This is partly because the lower-bound $\bar{p}_{t,y_t} \ge \mu \approx 1/n$ in Inequality (8) may be very coarse for many t. If the $\bar{p}_{t,y_t}$ were of order $\Omega(1)$, one could choose $m = O(n^2)$ instead of $m = O(n^3)$. Another possible explanation comes from the fact that, if A_t is large enough, the variance of the samples is small and the convergence is much faster. In our experiments, the choice of $m = 100$ already provides a good approximation. We leave to future work the possibility to improve the complexity in favorable scenarios.



(a) Vehicle (b) Shuttle (c) Segment

Figure 1: Averaged losses over time incurred by ONS, GAF, and OGD. The experiments were repeated 20 times and the empirical quantiles 0.25, 0.5, and 0.75 are reported.

5 Experiments

Although GAF is primarily theoretically motivated in a worst-case analysis, here we study its performance on real data sets. We consider three datasets (vehicle, shuttle, and segmentation taken from LIBSVM Data [1]) and compare the performance of GAF with two well-used algorithms: Online Gradient Descent (OGD) (Zinkevich, 2003) and Online Newton Step (ONS) (Hazan et al., 2007). The algorithm of Foster et al. (2018) is not considered because of prohibitive computational complexity. Concerning the hyper-parameters, the values suggested by the theory are generally too conservative. We thus choose the best ones in a grid for each algorithm ($\lambda, \beta \in [0.01, 0.03, 0.1, 0.3, 1, 3, 10]$).

The averaged losses over time are reported in Figure 1. We can remark that the performance of GAF is similar to the one of ONS when the number of samples is high. However, the learning of GAF seems more stable than ONS which leads sometimes (vehicle and segment) to better performance when there are few samples. This is not surprising as aggregating forecasters are hedging against worst-case scenario. Those results show that GAF is not only a theoretical oriented algorithm but could also be used successfully in practice. However, we expect it to perform best in a hard adversarial regime (close to the one described in Hazan et al. (2014)).

Conclusion and future work

We have shown how to leverage both mixability and quadratic approximations to design an algorithm GAF which achieves high statistical performance while being more efficient than existing algorithms. In particular, it achieves a new trade-off for multiclass logistic regression.

Some interesting questions are still remaining and left for future work. The linear dependence in B although better than what can be achieved by any proper algorithm is still sub-optimal compared to Foster et al. (2018) (logarithmic dependence). It may be possible to improve it by using other surrogates than quadratics. The essential point would be to prove an equivalent of Lemma 4 for those surrogates. The computational complexity of computing Equation (6) should also remain low. The computational complexity may also be improved. Like previous point, other surrogates could be used to make Equation (6) easier to compute. Finally, we believe that GAF should easily be extended to kernels. One would need to adapt the analysis to depend on the effective dimension (Jézéquel et al., 2019) of the RKHS instead of d. It would of interest to see how classical approximation algorithms, like Nystrom and random Fourier features, deteriorate statistical performance. Finally, we have only provided two examples of loss functions (multi-class logistic loss and squared loss). An intriguing question would be to see if the algorithm can be used to improve existing results for other losses such as Huber loss.

[1] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html`

References

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):211–246.

Bach, F. et al. (2010). Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414.

Bubeck, S., Eldan, R., and Lehec, J. (2018). Sampling from a log-concave distribution with projected langevin monte carlo. *Discrete & Computational Geometry*, 59(4):757–783.

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

Doerr, B. (2020). Probabilistic tools for the analysis of randomized optimization heuristics. In *Theory of Evolutionary Computation*, pages 1–87. Springer.

Foster, D. J., Kale, S., Luo, H., Mohri, M., and Sridharan, K. (2018). Logistic regression: The importance of being improper. *arXiv preprint arXiv:1803.09349*.

Hazan, E., Agarwal, A., and Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192.

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

Hazan, E., Koren, T., and Levy, K. Y. (2014). Logistic regression: Tight bounds for stochastic and online optimization. In *Conference on Learning Theory*, pages 197–209. PMLR.

Helmbold, D. P. and Warmuth, M. K. (1995). On weak learning. *Journal of Computer and System Sciences*, 50(3):551–573.

Jézéquel, R., Gaillard, P., and Rudi, A. (2019). Efficient online learning with kernels for adversarial large scale problems. *arXiv preprint arXiv:1902.09917*.

Jézéquel, R., Gaillard, P., and Rudi, A. (2020). Efficient improper learning for online logistic regression. In *Conference on Learning Theory*, pages 2085–2108. PMLR.

Marteau-Ferey, U., Ostrovskii, D., Bach, F., and Rudi, A. (2019). Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. *arXiv preprint arXiv:1902.03046*.

Mourtada, J. and Gaïffas, S. (2019). An improper estimator with optimal excess risk in misspecified density estimation and logistic regression. *arXiv preprint arXiv:1912.10784*.

Ostrovskii, D. M., Bach, F., et al. (2021). Finite-sample analysis of m-estimators using self-concordance. *Electronic Journal of Statistics*, 15(1):326–391.

Van Erven, T., Grunwald, P., Mehta, N. A., Reid, M., Williamson, R., et al. (2015). Fast rates in statistical and online learning.

Vovk, V. (2001). Competitive on-line statistics. *International Statistical Review*, 69(2):213–248.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936.

NeurIPS 2021 Paper Checklist

The NeurIPS Paper Checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] There are not potential negative societal impacts of this work. This is mostly a theoretical paper.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] The experiments were run on a personal laptop and are not computationally expensive.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]