Online PAC-Bayes Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Most PAC-Bayesian bounds hold in the batch learning setting where data is collected at once, prior to inference or prediction. This somewhat departs from many contemporary learning problems where data streams are collected and the algorithms must dynamically adjust. We prove new PAC-Bayesian bounds in this online learning framework, leveraging an updated definition of regret, and we revisit classical PAC-Bayesian results with a batch-to-online conversion, extending their remit to the case of dependent data. Our results hold for bounded losses, potentially *non-convex*, paving the way to promising developments in online learning.

1 Introduction

Batch learning is somewhat the dominant learning paradigm in which we aim to design the best predictor by collecting a training dataset which is then used for inference or prediction. Classical algorithms such as SVMs [see Cristianini et al., 2000, among many others] or feedforward neural networks [Svozil et al., 1997] are popular examples of efficient batch learning. While the mathematics of batch learning constitute a vivid and well understood research field, in practice this might not be aligned with the way practitionners collect data, which can be sequential when too much information is available at a given time (*e.g.* the number of micro-transactions made in finance on a daily basis). Indeed batch learning is not designed to properly handle dynamic systems.

Online learning (OL) [Zinkevich, 2003, Shalev-Shwartz, 2012, Hazan, 2016] fills this gap by treating data as a continuous stream with a potentially changing learning goal. OL has been studied with convex optimisation tools and the celebrated notion of regret which measures the discrepancy between the cumulative sum of losses for a specific algorithm at each datum and the optimal strategy. It led to many fruitful results comparing the efficiency of prediction for optimisation algorithms such that Online Gradient Descent (OGD), Online Newton Step through static regret [Zinkevich, 2003, Hazan et al., 2007]. OL is flexible enough to incorporate external expert advice onto classical algorithms with the optimistic point of view that such advices are useful for training [Rakhlin and Sridharan, 2013a,b] and then having optimistic regret bounds. Modern extensions also allow to compare to moving strategies through dynamic regret [see e.g. Yang et al., 2016, Zhang et al., 2018, Zhao et al., 2020]. However, this notion of regret has been challenged recently: for instance, Wintenberger [2021] chose to control an expected cumulative loss through PAC inequalities in order to deal with the case of stochastic loss functions.

Statements holding with arbitrarily large probability are widely used in learning and especially within the PAC-Bayes theory. Since its emergence in the late 90s, the PAC-Bayes theory (see the seminal works of Shawe-Taylor and Williamson, 1997, McAllester, 1998, 1999 and the recent surveys by Guedj, 2019, Alquier, 2021) has been a powerful tool to obtain generalisation bounds and to derive efficient learning algorithms. Classical PAC-Bayes generalisation bounds help to understand how

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

a learning algorithm may perform on future similar batches of data. More precisely, PAC-Bayes learning exploits the Bayesian paradigm of explaining a learning problem through a meaningful distribution over a space of candidate predictors [see e.g. Maurer, 2004, Catoni, 2007, Tolstikhin and Seldin, 2013, Mhammedi et al., 2019]. An active line of research in PAC-Bayes learning is to overcome classical assumptions such that data-free prior, bounded loss, iid data [see Lever et al., 2010, 2013, Alquier and Guedj, 2018, Holland, 2019, Rivasplata et al., 2020, Haddouche et al., 2021, Guedj and Pujol, 2021] while remaining in a batch learning sprit. Finally, a pioneering line of work led by [Seldin et al., 2012a,b] on PAC-Bayes learning for martingales and independently developed by [Gerchinovitz, 2011, Foster et al., 2015, Li et al., 2018] boosted PAC-Bayes learning by providing sparsity regret bound, adaptive regret bounds and online algorithms for clustering.

Our contributions. Our goal is to provide a general online framework for PAC-Bayesian learning. Our main contribution (Thm. 2.2 in Sec. 2) is a general bound which is then used to derive several online PAC-Bayesian results (as developed in Secs. 3 and 4). More specifically, we derive two types of bounds, *online PAC-Bayesian training and test bounds*. Training bounds exhibit online procedures while the test bound provide efficiency guarantees. We propose then several algorithms with their associated training and test bounds as well as a short series of experiments to evaluate the consistency of our online PAC-Bayesian approach. Our efficiency criterion is not the classical regret but an expected cumulative loss close to the one of Wintenberger [2021]. We emphasize that our PAC-Bayesian results only require a bounded loss to hold: no assumption is made on the data distribution, priors can be data-dependent and we do not require any convexity assumption on the loss, as commonly assumed in the OL framework. Finally, we acknowledge the recent and inspiring work of Neu and Lugosi [2022], which we see as the counterpart of our contribution. Indeed, they use tools from convex analysis to obtain learning theory results while we use PAC-Bayes learning, which is a branch of learning theory, to derive OL algorithms.

Outline. Sec. 2 introduces the theoretical framework as well as our main result. Sec. 3 presents an online PAC-Bayesian algorithm and draws links between PAC-Bayes and OL results. Sec. 4 details online PAC-Bayesian disintegrated procedures with reduced computational time and Sec. 5 gathers supporting experiments. We include reminders on OL and PAC-Bayes in Appendixes A and B. All proofs are deferred to Appendix C.

2 An online PAC-Bayesian bound

We establish a novel PAC-Bayesian theorem (which in turn will be particularised in Sec. 3) which overcomes the classical limitation of data-independent prior and iid data.

We call our main result an *online PAC-Bayesian bound* as it allows to consider a sequence of priors which may depend on the past and a sequence of posteriors that can dynamically evolve as well. Indeed, we follow the online learning paradigm which considers a continous stream of data that the algorithm has to process on the fly, adjusting its outputs at each time step w.r.t. the arrival of new data and the past. In the PAC-Bayesian framework, this paradigm translates as follows: from an initial (still data independent) prior $Q_1 = P$ and a data sample $S = (z_1, ..., z_m)$, we design a sequence of posterior $(Q_i)_{1 \leq i \leq m}$ where $Q_i = f(Q_1, ..., Q_{i-1}, z_i)$.

Framework. Consider a data space $\mathcal{Z}$ (which can be only inputs or pairs of inputs/outputs). We fix an integer $m > 0$ and our data sample $S \in \mathcal{Z}^m$ is drawn from an unknown distribution μ. We do not make any assumption on μ. We set a sequence of priors, starting with $P_1 = P$ a data-free distribution and $(P_i)_{i \geq 2}$ such that for each i, P_i is $\mathcal{F}_{i-1}$ measurable where $(\mathcal{F}_i)_{i \geq 0}$ is an adapted filtration to S. We also denote by Q_i our sequence of candidate posteriors. There is no restriction on what Q_i could be. In what follows we fix a filtration $(\mathcal{F}_i)_{i \geq 0}$ and we denote by KL the Kullback-Leibler divergence between two distributions.

We consider a predictor space $\mathcal{H}$ and a loss funtion $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$ bounded by a real constant $K > 0$. This loss defines the (potentially moving) learning objective. We denote by $\mathcal{M}_1(\mathcal{H})$ the set

of all probability distributions on $\mathcal{H}$. In particular, for all i, $P_i \in \mathcal{M}_1(\mathcal{H})$. In this work, the sequence (P_i) satisfies the following conditions.

Definition 2.1. *We say that a sequence of distributions $(P_i)_{i=1..m}$ is an* **online predictive sequence** *if (i) for all $i \geq 1$, P_i is $\mathcal{F}_{i-1}$ measurable and (ii) for all $i \geq 2$, $P_i \gg P_{i-1}$ where $Q \gg P$ denotes the absolute continuity of Q w.r.t. P.*

Note that (ii) implies that for all i, $P_i \gg P_1$ with P_1 a data-free measure (yet a classical prior in the PAC-Bayesian theory). We can now state our main result.

Theorem 2.2. *For any distribution μ over $\mathcal{Z}^m$, any $\lambda > 0$ and any online predictive sequence (used as priors) (P_i), for any posterior sequence (Q_i) the following holds with probability $1-\delta$ over the sample $S \sim \mu$:*

$$\sum_{i=1}^{m} \mathbb{E}_{h_i \sim Q_i} \left[\mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] \right] \leq \sum_{i=1}^{m} \mathbb{E}_{h_i \sim Q_i} \left[\ell(h_i, z_i) \right] + \frac{\mathrm{KL}(Q_i \| P_i)}{\lambda} + \frac{\lambda m K^2}{2} + \frac{\log(1/\delta)}{\lambda}.$$

The proof is deferred to Appendix C.1

A batch to online conversion. First, we remark that our bound slightly exceeds the OL framework: indeed, it would require our posterior sequence to be an online predictive sequence as well, which is not the case here (for any i, Q_i can depend on the whole dataset). This is a consequence of our proof method (see Appendix C.1), which is classically denoted as a "batch to online" conversion (in opposition to the "online to batch" procedures as in Dekel and Singer, 2005). In other words, we exploited PAC-Bayesian tools designed for a fixed batch of data to obtain a dynamic result. This is why we refer to our bound as online as it allows to consider sequences of priors and posteriors that can dynamically evolve.

Analysis of the different terms in the bound. Our PAC-Bayesian bound formally differs in many points from the classical ones. On the left-hand side of the bound, the sum of the averaged expected loss conditioned to the past appears. Having such a sum of expectations instead of a single one is necessary to assess the quality of all our predictions. Indeed, because data may be dependent, one can not consider a single expectation as in the iid case. We also stress that taking an online predictive sequence as priors leads to control losses conditioned to the past, which differs from classical PAC-Bayes results designed to bound the expected loss. This term, while original in the PAC-Bayesian framework (to the best of our knowledge) recently appeared (in a modified form) in Wintenberger [2021, Prop 3].

On the right hand-side of the bound, online counterparts of classical PAC-Bayes terms appear. At time i, the measure Q_i has a tradeoff to achieve between an overfitted prediction of z_i (the case $Q_i = \delta_{z_i}$ where δ is a Dirac measure) and a too weak impact of the new data with regards to our prior knowledge (the case $Q_i = P_i$). The quantity $\lambda > 0$ can be seen as a regulariser to adjust the relative impact of both terms.

Influence of λ. The quantity λ also plays a crucial role on the bound as it is involved in an explicit tradeoff between the KL terms, the confidence term $\log(1/\delta)$ and the residual term $mK^2/2$. This idea of seeing λ as a trading parameter is not new [Thiemann et al., 2017, Germain et al., 2016]. However, the results from Thiemann et al. [2017] stand w.p. $1-\delta$ for any λ while ours and the ones from Germain et al. [2016] hold for any λ w.p. $1-\delta$ which is weaker and implies to discretise $\mathbb{R}^+$ onto a grid to estimate the optimal λ.

We now move on to the design of online PAC-Bayesian algorithms.

3 An online PAC-Bayesian procedure

OL algorithms (we refer to Hazan, 2016 an introduction to the field) are producing sequences of predictors by progressively updating the considered predictor (see Appendix A for an example).

Recall that, in the OL framework, an algorithm outputs at time i a predictor which is $\mathcal{F}_{i-1}$-measurable. Here, our goal is to design an online procedure derived from Thm. 2.2 which outputs an online predictive sequence.

Online PAC-Bayesian (OPB) training bound. We state a corollary of our main result which paves the way to an online algorithm. This constructive procedure motivates the name *Online PAC-Bayesian training bound* (OPBTRAIN in short).

Corollary 3.1 (OPBTRAIN)**.** *For any distribution μ over $\mathcal{Z}^m$, any $\lambda > 0$ and any online predictive sequences $\hat{Q}, P$, the following holds with probability $1-\delta$ over the sample $S \sim \mu$:*

$$\sum_{i=1}^{m} \mathbb{E}_{h_i \sim \hat{Q}_{i+1}} \left[\mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}]\right] \leq \sum_{i=1}^{m} \mathbb{E}_{h_i \sim \hat{Q}_{i+1}} \left[\ell(h_i, z_i)\right] + \frac{\mathrm{KL}(\hat{Q}_{i+1} \| P_i)}{\lambda} + \frac{\lambda m K^2}{2} + \frac{\log(1/\delta)}{\lambda}.$$

The proof consists in applying Thm. 2.2 with for all i, $Q_i = \hat{Q}_{i+1}$ and P_i. Note that in this case, our posterior sequence is an online predictive sequence in order to fit with the OL framework.

Corollary 3.1 suggests to design $\hat{Q}$ as follows, assuming we have drawn a dataset $S = \{z_1, ..., z_m\}$, fixed a scale parameter $\lambda > 0$ and an online predictive sequence P_i:

$$\hat{Q}_1 = P, \quad \forall i \geq 1 \; \hat{Q}_{i+1} = \mathrm{argmin}_Q \, \mathbb{E}_{h_i \sim Q} \left[\ell(h_i, z_i)\right] + \frac{\mathrm{KL}(Q \| P_i)}{\lambda} \tag{1}$$

which leads to the explicit formulation

$$\frac{d\hat{Q}_{i+1}}{dP_i}(h) = \frac{\exp\left(-\lambda \ell(h, z_i)\right)}{\mathbb{E}_{h \sim P_i}\left[\exp\left(-\lambda \ell(h, z_i)\right)\right]}. \tag{2}$$

Thus, the formulation of Eq. (2), which has been highlighted by Catoni [2003, Sec. 5.1] shows that our online procedure produces Gibbs posteriors. So, PAC-Bayesian theory provides sound justification for the somewhat intuitive online procedure in Eq. (1): at time i, we adjust our new measure $\hat{Q}_{i+1}$ by optimising a tradeoff between the impact of the newly arrived data z_i and the one of prior knowledge $\hat{Q}_i$.

Notice that $\hat{Q}$ is an online predictive sequence: $\hat{Q}_i$ is $\mathcal{F}_{i-1}$-measurable for all i as it depends only on $\hat{Q}_{i-1}$ and z_{i-1}. Furthermore, one has $\hat{Q}_i \gg \hat{Q}_{i-1}$ for all i as $\hat{Q}_i$ is defined as an argmin and the KL term is finite if and only it is absolutely continuous w.r.t. $\hat{Q}_{i-1}$.

Remark 3.2. *In Corollary 3.1, while the right hand-side is the reason we considered Eq. (1), the left hand side still needs to be analysed. It expresses how the posterior $\hat{Q}_{i+1}$ (designed from $\hat{Q}_i, z_i$) generalises well on average to any new draw of z_i. More precisely, this term measures how much the training of $\hat{Q}_{i+1}$ is overfitting on z_i. A low value of it ensures our online predictive sequence, which is obtained from a single dataset, is robust to the randomness of S, hence the interest of optimising the right hand side of the bound. This is a supplementary reason we refer to Corollary 3.1 as an* OPBTRAIN bound as it provide robustness guarantees for our training.

Online PAC-Bayesian (OPB) test bound. However, Corollary 3.1 does not say if $\hat{Q}_{i+1}$ will produce good predictors to minimise $\ell(., z_{i+1})$, which is the objective of $\hat{Q}_{i+1}$ in the OL framework (we only have access to the past to predict the future). We then need to provide an *Online PAC-Bayesian (OPB) test bound* (OPBTEST bound) to quantify our prediction's accuracy. We now derive an OPBTEST bound from Thm. 2.2.

Corollary 3.3 (OPBTEST)**.** . *For any distribution μ over $\mathcal{Z}^m$, any $\lambda > 0$, and any online predictive sequence $(\hat{Q}_i)$, the following holds with probability $1-\delta$ over the sample $S \sim \mu$:*

$$\sum_{i=1}^{m} \mathbb{E}_{h_i \sim \hat{Q}_i} \left[\mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}]\right] \leq \sum_{i=1}^{m} \mathbb{E}_{h_i \sim \hat{Q}_i} \left[\ell(h_i, z_i)\right] + \frac{\lambda m K^2}{2} + \frac{\log(1/\delta)}{\lambda}.$$

The proof consists in applying Thm. 2.2 with for all i, $Q_i = \hat{Q}_i = P_i$.

Corollary 3.3 quantifies how efficient will our predictions be. Note that in this corollary the involved λ can differ from the scale parameter of Eq. (1), it is now a way to compensate for the tradeoff between the two last terms of the bound and the optimum λ yields an upper bound of $\sqrt{\log(1/\delta)K^2 m/2}$. The strength of this bound is that since $\hat{Q}$ is an online predictive sequence, the Kullback-Leibler terms vanished, leaving terms depending only on hyperparameters.

Links with previous approaches

We now present a specific case of Corollary 3.1 where we choose as priors the online predictive sequence $\hat{Q}$ (*i.e.* in Thm. 2.2, we choose $Q_i = \hat{Q}_{i+1}, P_i = \hat{Q}_i$). The reason we focus on this specific case is that it enables to build strong links between PAC-Bayes and OL.

We then adapt our OPBTRAIN bound (Corollary 3.1). The online procedure becomes:

$$\hat{Q}_1 = P, \quad \forall i \geq 1 \; \hat{Q}_{i+1} = \operatorname{argmin}_Q \mathbb{E}_{h_i \sim Q} \left[\ell(h_i, z_i)\right] + \frac{\mathrm{KL}(Q \| \hat{Q}_i)}{\lambda}, \tag{3}$$

which leads to the explicit formulation

$$\frac{d\hat{Q}_{i+1}}{d\hat{Q}_i}(h) = \frac{\exp\left(-\lambda \ell(h, z_i)\right)}{\mathbb{E}_{h \sim \hat{Q}_i}\left[\exp\left(-\lambda \ell(h, z_i)\right)\right]}.$$

Links with classical PAC-Bayesian bounds. We denote that the optimal predictor in this case is such that at any time i, $d\hat{Q}_{i+1}(h) \propto \exp(-\lambda \ell(h, z_i)) d\hat{Q}_i(h)$ hence $d\hat{Q}_{m+1}(h) \propto \exp\left(-\lambda \sum_{i=1}^m \ell(h, z_i)\right) d\hat{Q}_1(h)$. One recognises, up to a multiplicative constant, the optimised predictor of Catoni [2007, Th 1.2.6] which solves $\operatorname{argmin}_Q \mathbb{E}_{h \sim Q}\left[\frac{1}{m}\sum_{i=1}^m \ell(h, z_i)\right] + \frac{\mathrm{KL}(Q \| \hat{Q}_1)}{\lambda}$, thus one sees that in this case, the output of our online procedure after m steps coincides with Catoni's output. This shows consistency of our general procedure which recovers classical result within an online framework: when too many data are available, treating data sequentially until time m leads to the same Gibbs posterior than if we were treating the whole dataset as a batch.

Analogy with Online Gradient Descent (OGD). We propose an analogy between the procedure Eq. (3) and the celebrated OGD algorithm (see Appendix A for a recap). First we remark that our minimisation problem is equivalent to $\operatorname{argmin}_Q \lambda \mathbb{E}_{h_i \sim Q}\left[\ell(h_i, z_i)\right] + \mathrm{KL}(Q \| \hat{Q}_i)$. Thus an analogy between the scale parameter λ and the step size η in OGD can be derived. Furthermore we point an analogy between the expected term in Eq. (3) and the gradient term in OGD: both of these terms are accounting for the influence of the new point. We can push the analogy further when the candidate posteriors Q is characterised by a real-valued vector μ (thus $Q = q_\mu$): Shalev-Shwartz [2012] proposed the approximation $\mathbb{E}_{h_i \sim q_\mu}[\ell(h_i, z_i)] := \bar{L}_i(\mu) \approx \mu^T \nabla \bar{L}_i(\mu_i)$ where μ is their considered online predictive sequence. Thus, updating μ_i with this approximation involves directly the gradient $\nabla \bar{L}_i(\mu_i)$. Finally, we remark that the optimum rate in Corollary 3.3 is a $O(\sqrt{m})$ which is comparable to the best rate of Shalev-Shwartz [2012, Eq (2.5)] (see proposition A.2).

Comparison with previous work. We acknowledge that the procedure of Eq. (3) already appeared in literature. Li et al. [2018, Alg. 1] propose a Gibbs procedure somewhat similar to ours, the main difference being the addition of a surrogate of the true loss at each time step. Within the OL literature, the idea of updating measures online has been recently studied for instance in Chérief-Abdellatif et al. [2019]. More precisely, our procedure is similar to their Streaming Variational Bayes (SVB) algorithm. A slight difference is that they approximated the expected loss similarly to Shalev-Shwartz [2012]. The guarantees Chérief-Abdellatif et al. [2019] provided for SVB hold for Gaussian priors and posteriors and are valid for iid data, which is a particular case of our work. Their bounds are deterministic and are using tools and assumptions from convex optimisation (such that convex expected losses) while ours are probabilistic and are using measure theory tools which allow to relax these assumptions.

Strength of our result. We emphasize two points. First, to the best of our knowledge, Corollary 3.1 is the first bound which theoretically suggests Eq. (3) as a learning algorithm. Second, we stress

that Eq. (3) is a particular case of Corollary 3.1 and our result can lead to other fruitful routes. For instance, we consider the idea of adding noise to our measures at each time step to avoid overfitting (this idea has been used *e.g.* in Neelakantan et al., 2015 in the context of deep neural networks): if our online predicitve sequence $(\hat{Q}_i)$ can be defined through a sequence of parameter vectors $\hat{\mu}$, then we can define P_i by adding a small noise on $\hat{\mu}_i$ and thus giving more freedom through stochasticity.

Thus, we see that our procedure led us to the use of the Gibbs posteriors of Catoni. However, in practice, Gaussian distributions are preferred [*e.g.* Dziugaite and Roy, 2017, Rivasplata et al., 2019, Perez-Ortiz et al., 2021b,a, Pérez-Ortiz et al., 2021]). That is why we focus next on new online PAC-Bayesian algorithms involving Gaussian distributions.

4 Disintegrated online algorithms for Gaussian distributions.

We dig deeper in the field of disintegrated PAC-Bayesian bounds, originally explored by Catoni [2007], Blanchard and Fleuret [2007], further studied by Alquier and Biau [2013], Guedj and Alquier [2013] and recently developed by Rivasplata et al. [2020], Viallard et al. [2021] (see Appendix B for a short presentation of the bound we adapted and used). The strength of the disintegrated approach is that we have directly guarantees on the random draw of a single predictor, which avoids to consider expectations over the predictor space. This fact is particularly significant in our work as the procedure precised in Eq. (2), require the estimation of an exponential moment to be efficient, which may be costful. We then show that disintegrated PAC-Bayesian bounds can be adapted to the OL framework, and that they have the potential to generate proper online algorithms with weak computational cost and sound efficiency guarantees.

Online PAC-Bayesian disintegrated (OPBD) training bounds. We present a general form for *online PAC-Bayes disintegrated (OPBD) training bounds*. The terminology comes from the way we craft those bounds: from PAC-Bayesian disintegrated bounds we use the same tools as in Thm. 2.2 to create the first online PAC-Bayesian disintegrated bounds. OPBD training bounds have the following form.

For any online predictive sequences $\hat{Q}, P$, any $\lambda > 0$ w.p. $1-\delta$ over $S \sim \mu$ and $(h_1, ..., h_m) \sim \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}$:

$$\sum_{i=1}^{m} \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] \leq \sum_{i=1}^{m} \ell(h_i, z_i) + \Psi(h_i, \hat{Q}_{i+1}, P_i) \ + \Phi(m), \tag{4}$$

with Ψ, Φ being real-valued functions. Ψ controls the global behaviour of Q_{i+1} w.r.t. the $\mathcal{F}_{i-1}$-measurable prior P_i. If one has no dependency on h_i this behaviour is global, otherwise it is local. Note that those functions may depend on λ, δ. However, since they are fixed parameters, we do not make these dependencies explicit. Similarly to Corollary 3.1, this kind of bound allows to derive a learning algorithm (cf Algorithm 1) which outputs an online predicitve sequence $\hat{Q}$. Finally we draw $(h_1, ..., h_m) \sim \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}$ (and not $\hat{Q}_1 \otimes ... \otimes \hat{Q}_m$) since an OPBD bound is designed to justify theoretically an OPBD procedure in the same way Corollary 3.1 allowed to justify Eq. (1).

Why focus on Gaussian measures? The reason is that a Gaussian variable $h \sim \mathcal{N}(w, \sigma^2 \mathbf{I}_d)$ can be written as $h = w + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$, and this expression totally defines h ($\mathbf{I}_d$ being the identity matrix).

A general OPBD algorithm for Gaussian measure with fixed variance We use an idea presented in Viallard et al. [2021] which restrict the measure set to Gaussian on $\mathbb{R}^d$ **with known and fixed covariance matrix** $\sigma^2 \mathbf{I}_d$. Then we present in Algorithm 1 a general algorithm (derived from an OPBD training bound) for Gaussian measures with fixed variance which outputs a sequence of gaussian $\hat{Q}_i = \mathcal{N}(\hat{w}_i, \sigma^2 \mathbf{I}_d)$ from a prior sequence $P_i = \mathcal{N}(w_i^0, \sigma^2 \mathbf{I}_d)$ where for each i, w_i^0 is $\mathcal{F}_{i-1}$-measurable. Because the variance is fixed, the distribution is uniquely defined by its mean, thus we identify $\hat{Q}_i$ and $\hat{w}_i$, P_i and w_i^0.

Algorithm 1: A general OPBD algorithm for Gaussian measures with fixed variance.

Parameters : Time m, scale parameter λ
Initialisation : Variance σ^2, Initial mean $\hat{w}_1 \in \mathbb{R}^d$, epoch m

for *each iteration* i *in* $1..m$ **do**
  Observe z_i, w_i^0 and draw $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$
  Update:

$$\hat{w}_{i+1} := \text{argmin}_{w \in \mathbb{R}^d} \ell(w + \varepsilon_i, z_i) + \Psi(w + \varepsilon_i, w, w_i^0)$$

end
Return $(\hat{w}_i)_{i=1..m+1}$

At each time i, Algorithm 1 requires the draw of $\varepsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$. Doing so, we generated the randomness for our h_i (because our bound holds for a single draw of $(h_1, .., h_m) \sim \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}$), we then write $h_i = w + \varepsilon_i$ and we optimise w.r.t. Ψ to find $\hat{w}_{i+1}$.

Bounds of interest. We present two possible choices of pairs (Ψ, Φ) derived from the disintegrated results presented in Appendix B. Doing so, we explicit two ready-to-use declinations of Algorithm 1.

Corollary 4.1. *For any distribution μ over $\mathcal{Z}^m$, any online predictive sequences of Gaussian measures with fixed variance $\hat{Q}_i = \mathcal{N}(\hat{w}_i, \sigma^2 \mathbf{I}_d)$ and $P_i = \mathcal{N}(w_i^0, \sigma^2 \mathbf{I}_d)$, any $\lambda > 0$, w.p. $1 - \delta$ over $S \sim \mu$ and $(h_i = \hat{w}_{i+1} + \varepsilon_i)_{i=1..m} \sim \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}$, the bound of Eq. (4) holds for the two following pairs Ψ, Φ:*

$$\Psi_1(h_i, \hat{w}_{i+1}, w_i^0) = \frac{1}{\lambda}\left(\frac{||\hat{w}_{i+1} + \varepsilon_i - w_i^0||^2 - ||\varepsilon||^2}{2\sigma^2}\right) \quad \Phi_1(m) = \frac{\lambda m K^2}{2} + \frac{\log(1/\delta)}{\lambda}, \tag{5}$$

$$\Psi_2(h_i, \hat{w}_{i+1}, w_i^0)) = \frac{1}{\lambda}\frac{||\hat{w}_{i+1} - w_i^0||^2}{2\sigma^2} \quad \Psi_2(m) = \lambda m K^2 + \frac{3\log(1/\delta)}{2\lambda}. \tag{6}$$

Where the notation $1, 2$ denote whether the functions have been derived from adapted theorems of Rivasplata et al., 2020, Viallard et al., 2021 recalled in Appendix B We then can use algorithm 1 with Eq. (5), Eq. (6).

Proof is deferred to Appendix C.2. Note that in Corollary 4.1, we identified $\hat{Q}_i$ to $\hat{w}_i$ and for the last formula, Ψ has no dependency on h_i.

Comparison with Eq. (1). The main difference with Eq. (1) provided by the disintegrated framework is that the optimisation route does not include an expected term within the optimisation objective. The main advantage is a weaker computational cost when we restrict to Gaussian distributions. The main weakness is a lack of stability as our algorithm now depends at time i on $\ell(h + \varepsilon_i, z_i)$ so on ε_i directly. We denote that Eq. (5) is less stable than Eq. (6) as it involves another dependency on ε_i through Ψ. The reason is that Rivasplata et al. [2020] proposed a bound involving a disintegrated KL divergence while Viallard et al. [2021] proposed a result involving a Rényi divergence avoiding a dependency on ε_i. We refer to Appendix B for a detailed statement of those properties.

OPBD test bounds. Similarly to what we did in Sec. 3, we also provide *OPBD test bounds* to provide efficiency guarantees for online predicitve sequences (e.g. the output of algorithm 1). Our proposed bounds have the following general form.

For any online predictive sequence $\hat{Q}$, any $\lambda > 0$ w.p. $1-\delta$ over S and $(h_1, ..., h_m) \sim \hat{Q}_1 \otimes ... \otimes \hat{Q}_m$:

$$\sum_{i=1}^{m} \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] \leq \sum_{i=1}^{m} \ell(h_i, z_i) + \Phi(m), \tag{7}$$

with Φ being a real-valued function(possibly dependent on λ, δ though it is not explicited here).

Note that our predictors $(h_1, ..., h_m)$ are now drawn from $\hat{Q}_1 \otimes ... \otimes \hat{Q}_m$. Thus, the left-hand side of the bound considers a h_i drawn from an $\mathcal{F}_{i-1}$-measurable distribution evaluated on $\ell(., z_i)$: this is effectively a measure of the prediction performance.

We now state a corollary which gives disintegrated guarantees for any online predicitve sequence.

Corollary 4.2. *For any distribution μ over $\mathcal{Z}^m$, any $\lambda > 0$, and any online predictive sequence $(\hat{Q}_i)$, the following holds with probability $1-\delta$ over the sample $S \sim \mu$ and the predictors $(h_1, ..., h_m) \sim \hat{Q}_1 \otimes ... \otimes \hat{Q}_m$, the bound of Eq. (7) holds with :*

$$\Phi_1(m) = \frac{\lambda m K^2}{2} + \frac{\log(1/\delta)}{\lambda}, \quad \Phi_2(m) = 2\lambda m K^2 + \frac{\log(1/\delta)}{\lambda}.$$

Where the notation $1, 2$ denote whether the functions have been derived from adapted theorems of Rivasplata et al., 2020, Viallard et al., 2021 recalled in Appendix B. The optimised λ gives in both cases a $O(\sqrt{m \log(1/\delta)}$.

Proof is deferred to Appendix C.2.

5 Experiments

We adapt the experimental framework introduced in Chérief-Abdellatif et al. [2019, Sec.5] to our algorithms (anonymised code available here). We conduct experiments on several real-life datasets, in classification and linear regression. Our objective is twofold: check the convergence of our learning methods and compare their efficiencies with classical algorithms. We first introduce our experimental setup.

Algorithms. We consider four online methods of interest: the OPB algorithm of Eq. (3) which update through time a Gibbs posterior. We instantiate it with two different priors $\hat{Q}_1$: a Gaussian distribution and a Laplace one. We also implement Algorithm 1 with the functions Ψ_1, Ψ_2 from Corollary 4.1. To assess efficiency, we implement the classical OGD (as described in Alg. 1 of Zinkevich, 2003) and the SVB method of Chérief-Abdellatif et al. [2019].

Binary Classification. At each round i the learner receives a data point $x_i \in \mathbb{R}^d$ and predicts its label $y_i \in \{-1, +1\}$ using $\langle x_i, h_i \rangle$, with $h_i = \mathbb{E}_{h \sim \hat{Q}_i}[h]$ for OPB methods or h_i being drawn under $\hat{Q}_i$ for OPBD methods. The adversary reveals the true value y_i, then the learner suffers the loss $\ell(h_i, z_i) = \left(1 - y_i h_i^T x_i\right)_+$ with $z_i = (x_i, y_i)$ and $a_+ = a$ if $a > 0$ and $a_+ = 0$ otherwise. This loss is unbounded but can be thresholded.

Linear Regression. At each round i, the learner receives a set of features $x_i \in \mathbb{R}^d$ and predicts $y_i \in \mathbb{R}$ using $\langle x_i, h_i \rangle$ with $h_i = \mathbb{E}_{h \sim \hat{Q}_i}[h]$ for SVB and OPB methods or h_i being drawn under $\hat{Q}_i$ for OPBD methods. Then the adversary reveals the true value y_t and the learner suffers the loss $\ell(h_i, z_i) = \left(y_i - h_i^T x_i\right)^2$ with $z_i = (x_i, y_i)$. This loss is unbounded but can be thresholded.

Datasets. We consider four real world dataset: two for classification (Breast Cancer and Pima Indians), and two for regression (Boston Housing and California Housing). All datasets except the Pima Indians have been directly extracted from `sklearn` [Pedregosa et al., 2011]. Breast Cancer dataset [Street et al., 1993] is available here and comes from the UCI ML repository as well as the Boston Housing dataset [Belsley et al., 2005] which can be obtained here. California Housing dataset [Pace and Barry, 1997] comes from the StatLib repository and is available here. Finally, Pima Indians dataset [Smith et al., 1988] has been recovered from this Kaggle repository. Note that we randomly permuted the observations to avoid to learn irrelevant human ordering of data (such that date or label).

Parameter settings. We ran our experiments on a 2021 MacBookPro with an M1 chip and 16 Gb RAM. For OGD, the initialisation point is $\mathbf{0}_{\mathbb{R}^d}$ and the values of the learning rates are set to $\eta = 1/\sqrt{m}$. For SVB, mean is initialised to $\mathbf{0}_{\mathbb{R}^d}$ and covariance matrix to $\mathrm{Diag}(1)$. Step at time i is $\eta_i = 0.1/\sqrt{i}$. For both of the OPB algorithms with Gibbs posterior, we chose $\lambda = 1/m$. As priors, we took respectively a centered Gaussian vector with the covariance matrix $\mathrm{Diag}(\sigma^2)$ ($\sigma = 1.5$) and an iid vector following the standard Laplace distribution. For the OPBD algorithm with Ψ_1, we chose $\lambda = 10^{-4}/m$, the initial mean is $\mathbf{0}_{\mathbb{R}^d}$ and our fixed covariance matrix is $\mathrm{Diag}(\sigma^2)$ with $\sigma = 3.10^{-3}$. For the OPBD algorithm with Ψ_1, we chose $\lambda = 2.10^{-3}/m$, the initial mean is $\mathbf{0}_{\mathbb{R}^d}$

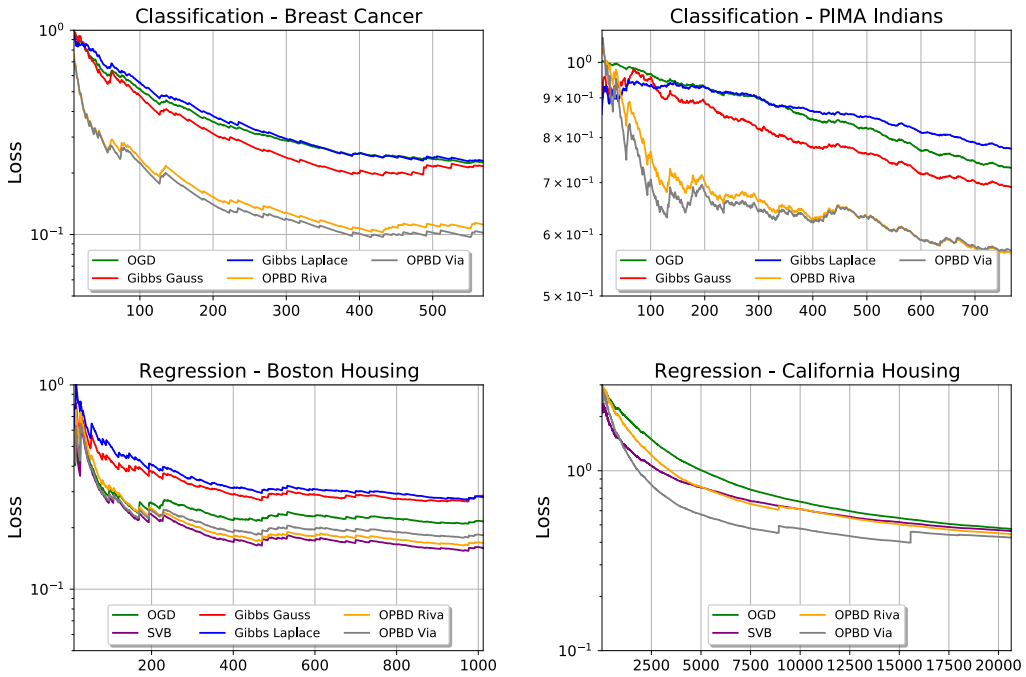

Figure 1: Averaged cumulative losses for all four considered datasets. 'Gibbs Gauss' denotes OPB with Gaussian Prior, 'Gibbs Laplace' denotes OPB with Laplace prior. 'OPBD Riva' denotes OPBD with Ψ_1, 'OPBD Via' denotes OPBD with Ψ_2.

and our covariance matrix is $\mathrm{Diag}(\sigma^2)$ with $\sigma = 10^{-2}$. The reason of those higher scale parameters and variance is that Ψ from Rivasplata et al. [2020] is more stochastic (yet unstable) than the one Viallard et al. [2021].

Experimental results. For each dataset, we plot the evolution of the average cumulative loss $\sum_{i=1}^{t} \ell(h_i, z_i)/t$ as a function of the step $t = 1, \ldots, m$, where m is the dataset size and h_i is the decision made by the learner h_i at step i. The results are gathered in Fig. 1

Empirical findings. OPB with Gaussian prior ('Gibbs Gauss') outperforms OGD on all datasets except California Housing (on which this method is not implemented) while OPB with Laplace prior ('Gibbs Laplace') always fail w.r.t. OGD. OPB methods fail to compete with SVB on the Boston Housing dataset. OPBD methods compete with SVB on regression problems and clearly outperforms OGD on classification tasks. OPBD with Ψ_2 (labeled as 'OPBD Via' in Fig. 1) performs better on the California Housing dataset while OPBD with Ψ_1 (labeled as 'OPBD Riva') is more efficient on the Boston Housing dataset. Both methods performs roughly equivalently on classification tasks. This brief experimental validation shows the consistency of all our online procedures as we observe a visible decrease of the cumulative losses through time. It particularly shows that OPBD procedures improve on OGD on these dataset.

6 Conclusion

We establish links between online learning and PAC-Bayes. We show that PAC-bayesian bounds are useful to derive new OL algorithms. We also prove sound theoretical guarantees for such algorithms. We emphasise that all of our results stand for any general bounded loss, especially no convexity assumption is needed. Having no convexity assumption on the loss paves the way to exciting future practical studies, starting with *Spiking Neural Network* which is investigated in an online fashion (see Lobo et al., 2020 for a recent survey). A follow-up question on the theoretical part is whether we can relax the bounded loss assumption: we leave this for future work.

References

P. Alquier. User-friendly introduction to PAC-Bayes bounds, 2021. URL https://arxiv.org/abs/2110.11216.

P. Alquier and G. Biau. Sparse single-index model. *Journal of Machine Learning Research*, 14(1), 2013.

P. Alquier and B. Guedj. Simpler PAC-Bayesian bounds for hostile data. *Machine Learning*, 107(5): 887–902, 2018. ISSN 1573-0565. URL http://dx.doi.org/10.1007/s10994-017-5690-0.

D. A. Belsley, E. Kuh, and R. E. Welsch. *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons, 2005.

G. Blanchard and F. Fleuret. Occam's hammer. In *International Conference on Computational Learning Theory*, pages 112–126. Springer, 2007.

O. Catoni. A PAC-Bayesian approach to adaptive classification. *preprint*, 840, 2003.

O. Catoni. PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning. Institute of Mathematical Statistics Lecture Notes—Monograph Series 56. *IMS, Beachwood, OH. MR2483528*, 5544465, 2007.

B.-E. Chérief-Abdellatif, P. Alquier, and M. E. Khan. A generalization bound for online variational inference. In *Asian Conference on Machine Learning*, pages 662–677. PMLR, 2019.

N. Cristianini, J. Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

O. Dekel and Y. Singer. Data-driven online to batch conversions. *Advances in Neural Information Processing Systems*, 18, 2005.

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

D. J. Foster, A. Rakhlin, and K. Sridharan. Adaptive Online Learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/19de10adbaa1b2ee13f77f679fa1483a-Paper.pdf.

S. Gerchinovitz. Sparsity Regret Bounds for Individual Sequences in Online Linear Regression. In S. M. Kakade and U. von Luxburg, editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 377–396, Budapest, Hungary, 09–11 Jun 2011. PMLR. URL https://proceedings.mlr.press/v19/gerchinovitz11a.html.

P. Germain, F. Bach, A. Lacoste, and S. Lacoste-Julien. PAC-Bayesian theory meets Bayesian inference. *Advances in Neural Information Processing Systems*, 29, 2016.

M. Gil, F. Alajaji, and T. Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

B. Guedj. A Primer on PAC-Bayesian Learning. In *Proceedings of the second congress of the French Mathematical Society*, 2019.

B. Guedj and P. Alquier. PAC-Bayesian estimation and prediction in sparse additive models. *Electron. J. Statist.*, 7:264–291, 2013. doi: 10.1214/13-EJS771. URL https://doi.org/10.1214/13-EJS771.

B. Guedj and L. Pujol. Still No Free Lunches: The Price to Pay for Tighter PAC-Bayes Bounds. *Entropy*, 23(11), 2021. ISSN 1099-4300. doi: 10.3390/e23111529. URL https://www.mdpi.com/1099-4300/23/11/1529.

M. Haddouche, B. Guedj, O. Rivasplata, and J. Shawe-Taylor. PAC-Bayes unleashed: generalisation bounds with unbounded losses. *Entropy*, 23(10):1330, 2021.

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

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

M. Holland. PAC-Bayes under potentially heavy tails. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2715–2724. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8539-pac-bayes-under-potentially-heavy-tails.pdf.

G. Lever, F. Laviolette, and J. Shawe-Taylor. Distribution-Dependent PAC-Bayes Priors. In M. Hutter, F. Stephan, V. Vovk, and T. Zeugmann, editors, *Algorithmic Learning Theory*, pages 119–133, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-16108-7.

G. Lever, F. Laviolette, and J. Shawe-Taylor. Tighter PAC-Bayes Bounds through Distribution-Dependent Priors. *Theor. Comput. Sci.*, 473:4–28, Feb. 2013. ISSN 0304-3975. URL https://doi.org/10.1016/j.tcs.2012.10.013.

L. Li, B. Guedj, and S. Loustau. A quasi-Bayesian perspective to online clustering. *Electronic Journal of Statistics*, 12(2):3071 – 3113, 2018. doi: 10.1214/18-EJS1479. URL https://doi.org/10.1214/18-EJS1479.

J. L. Lobo, J. Del Ser, A. Bifet, and N. Kasabov. Spiking neural networks and online learning: An overview and perspectives. *Neural Networks*, 121:88–100, 2020.

A. Maurer. A note on the PAC Bayesian theorem. *arXiv preprint cs/0411099*, 2004.

D. A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational Learning Theory*, pages 230–234. ACM, 1998.

D. A. McAllester. PAC-Bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational Learning Theory*, pages 164–170. ACM, 1999.

Z. Mhammedi, P. Grünwald, and B. Guedj. PAC-Bayes Un-Expected Bernstein Inequality. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12202–12213. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9387-pac-bayes-un-expected-bernstein-inequality.pdf.

A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

G. Neu and G. Lugosi. Generalization Bounds via Convex Analysis. In *COLT*, 2022. arXiv preprint arXiv:2202.04985.

R. K. Pace and R. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297, 1997.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

M. Perez-Ortiz, O. Rivasplata, B. Guedj, M. Gleeson, J. Zhang, J. Shawe-Taylor, M. Bober, and J. Kittler. Learning PAC-Bayes priors for probabilistic neural networks. Submitted., 2021a. URL https://arxiv.org/abs/2109.10304.

M. Perez-Ortiz, O. Rivasplata, E. Parrado-Hernandez, B. Guedj, and J. Shawe-Taylor. Progress in self-certified neural networks. In *NeurIPS 2021 workshop Bayesian Deep Learning [BDL]*, 2021b. URL http://bayesiandeeplearning.org/2021/papers/38.pdf.

M. Pérez-Ortiz, O. Rivasplata, J. Shawe-Taylor, and C. Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research*, 22, 2021.

A. Rakhlin and K. Sridharan. Online Learning with Predictable Sequences. In S. Shalev-Shwartz and I. Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 993–1019, Princeton, NJ, USA, 12–14 Jun 2013a. PMLR. URL https://proceedings.mlr.press/v30/Rakhlin13.html.

S. Rakhlin and K. Sridharan. Optimization, Learning, and Games with Predictable Sequences. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013b. URL https://proceedings.neurips.cc/paper/2013/file/f0dd4a99fba6075a9494772b58f95280-Paper.pdf.

O. Rivasplata, V. M. Tankasali, and C. Szepesvári. PAC-Bayes with backprop. *arXiv preprint arXiv:1908.07380*, 2019.

O. Rivasplata, I. Kuzborskij, C. Szepesvári, and J. Shawe-Taylor. PAC-Bayes analysis beyond the usual bounds. *Advances in Neural Information Processing Systems*, 33:16833–16845, 2020.

Y. Seldin, N. Cesa-Bianchi, P. Auer, F. Laviolette, and J. Shawe-Taylor. PAC-Bayes-Bernstein Inequality for Martingales and its Application to Multiarmed Bandits. In D. Glowacka, L. Dorard, and J. Shawe-Taylor, editors, *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*, volume 26 of *Proceedings of Machine Learning Research*, pages 98–111, Bellevue, Washington, USA, 02 Jul 2012a. PMLR. URL https://proceedings.mlr.press/v26/seldin12a.html.

Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer. PAC-Bayesian Inequalities for Martingales. *IEEE Transactions on Information Theory*, 58(12):7086–7093, 2012b. doi: 10.1109/TIT.2012.2211334.

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

J. Shawe-Taylor and R. C. Williamson. A PAC analysis of a Bayes estimator. In *Proceedings of the 10th annual conference on Computational Learning Theory*, pages 2–9. ACM, 1997.

J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, and R. S. Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, page 261. American Medical Informatics Association, 1988.

W. N. Street, W. H. Wolberg, and O. L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–870. SPIE, 1993.

D. Svozil, V. Kvasnicka, and J. Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

N. Thiemann, C. Igel, O. Wintenberger, and Y. Seldin. A strongly quasiconvex PAC-Bayesian bound. In *International Conference on Algorithmic Learning Theory*, pages 466–492. PMLR, 2017.

I. O. Tolstikhin and Y. Seldin. PAC-Bayes-Empirical-Bernstein Inequality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/a97da629b098b75c294dffdc3e463904-Paper.pdf.

P. Viallard, P. Germain, A. Habrard, and E. Morvant. A general framework for the derandomization of PAC-Bayesian bounds. 2021. URL https://arxiv.org/abs/2102.08649.

O. Wintenberger. Stochastic Online Convex Optimization; Application to probabilistic time series forecasting. *arXiv preprint arXiv:2102.00729*, 2021.

T. Yang, L. Zhang, R. Jin, and J. Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, pages 449–457. PMLR, 2016.

L. Zhang, T. Yang, rong jin, and Z.-H. Zhou. Dynamic Regret of Strongly Adaptive Methods. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5882–5891. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/zhang18o.html.

P. Zhao, Y.-J. Zhang, L. Zhang, and Z.-H. Zhou. Dynamic Regret of Convex and Smooth Functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12510–12520. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/939314105ce8701e67489642ef4d49e8-Paper.pdf.

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

Checklist

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] From the abstract to the conclusion, we explicited our main assumption (bounded loss).
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Due to the theoretical nature of our contribution, we do not foresee immediate societal impact. If anything, we certainly hope a better theoretical understanding of online learning algorithms can foster a more responsible use by practitionners.
   (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] for all theorems and corollaries
   (b) Did you include complete proofs of all theoretical results? [Yes] But deferred to Appendix C due to space constraints.
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] We have included the url in Sec. 5, note that this is an anonymous repository.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All details are gathered in Sec. 5 so that all readers can replicate our experiments and validate our fundings.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We chose to consider the classical online gradient descent as main comparison.
   (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)? [Yes] We precised the machine we used in the 'Parameters Settings' paragraph of Sec. 5.
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? [Yes] We precised the origin of our datasets in the 'Datasets' paragraph of Sec. 5.
   (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? [Yes] All the original code is available on the anonymised repository precised in Sec. 5.
   (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]