Online PAC-Bayes Learning

Maxime Haddouche
Inria and University College London
France and UK

Benjamin Guedj
Inria and University College London
France and UK

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 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

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

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. 2.3 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]. More precisely, Sec. 3 propose a stable yet time-consuming Gibbs-based algorithm, while Sec. 4 proposes time efficient yet volatile algorithms. 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.

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.1 and C. Appendix B provide disucssion about our main result. All proofs are deferred to Appendix D.

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. For $P, Q \in \mathcal{M}_1(\mathcal{H})$, the notation $P \ll Q$ indicates that Q is absolutely continuous wrt P (i.e. $Q(A) = 0$ if $P(A) = 0$ for measurable $A \subset \mathcal{H}$). 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 $\mathcal{H}$. We now introduce the notion of *stochastic kernel* [Rivasplata et al., 2020] which formalise properly data-dependent measures within the PAC-Bayes framework. First, for a fixed predictor space $\mathcal{H}$, we set $\Sigma_{\mathcal{H}}$ to be the considered σ-algebra on $\mathcal{H}$.

Definition 2.1 (Stochastic kernels)**.** *A* stochastic kernel *from $\mathcal{S} = \mathcal{Z}^m$ to $\mathcal{H}$ is defined as a mapping $Q : \mathcal{Z}^m \times \Sigma_{\mathcal{H}} \to [0; 1]$ where*

- *For any $B \in \Sigma_{\mathcal{H}}$, the function $S = (z_1, ..., z_m) \mapsto Q(S, B)$ is measurable,*
- *For any $S \in \mathcal{Z}^m$, the function $B \mapsto Q(S, B)$ is a probability measure over $\mathcal{H}$.*

We denote by $\texttt{Stoch}(\mathcal{S}, \mathcal{H})$ the set of all stochastic kernels from $\mathcal{S}$ to $\mathcal{H}$ and for a fixed S, we set $Q_S := Q(S, .)$ the data-dependent prior associated to the sample S through Q.

From now, to refer to a distribution Q_S depending on a dataset S, we introduce a stochastic kernel $Q(.,.)$ such that $Q_S = Q(S,.)$. Note that this notation is perfectly suited to the case when Q_S is obtained from an algorithmic procedure A. In this case the stochastic kernel Q of interest is the learning algorithm A. We use this notion to characterise our sequence of priors.

Definition 2.2. *We say that a sequence of stochastic kernels $(P_i)_{i=1..m}$ is an* **online predictive sequence** *if (i) for all $i \geq 1, S \in \mathcal{Z}^m, P_i(S,.)$ is $\mathcal{F}_{i-1}$ measurable and (ii) for all $i \geq 2$, $P_i(S,.) \gg P_{i-1}(S,.)$.*

Note that (ii) implies that for all $i, P_i(S,.) \gg P_1(S,.)$ with $P_1(S,.)$ a data-free measure (yet a classical prior in the PAC-Bayesian theory).

We can now state our main result.

Theorem 2.3. *For any distribution μ over $\mathcal{Z}^m$, any $\lambda > 0$ and any online predictive sequence (used as priors) (P_i), for any sequence of stochastic kernels (Q_i) we have with probability $1-\delta$ over the sample $S \sim \mu$, the following, holding for the data-dependent measures $Q_{i,S} := Q_i(S,.), P_{i,S} := P_i(S,.)$:*

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

Remark 2.4. *For the sake of clarity, we assimilate in what follows the stochastic kernels Q_i, P_i to the data-dependent distributions $Q_i(S,.), P_i(S,.)$. Then, an online predictive sequence is also assimilated to a sequence of data-dependent distributions. Concretely this leads to the switch of notation $Q_{i,S} \to Q_i$ in Thm. 2.3. The reason of this switch is that, even though stochastic kernel is the right theoretical structure to state our main result, we consider in Secs. 3 and 4 practical algorithmic extensions which focus only on data-dependent distributions, hence the need to alleviate our notations.*

The proof is deferred to Appendix D.1. See Appendix B for context and discussions.

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, the distribution $Q_{i,S}$ can depend on the whole dataset). This is a consequence of our proof method (see Appendix D.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]. See Appendix B.2 for further disucssions.

On the right hand-side of the bound, online counterparts of classical PAC-Bayes terms appear. At time i, the measure Q_i (i.e. $Q_{i,S}$ according to Remark 2.4) 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.1 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. 2.3 which outputs an online predictive sequence (which is assimilated, according to Remark 2.4, to a sequence of distributions).

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 $S \sim \mu$:*

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

Here, λ is seen as a scale parameter as precised below. The proof consists in applying Thm. 2.3 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:

$$\hat{Q}_1 = P_1, \quad \forall i \geq 1 \; \hat{Q}_{i+1} = \underset{Q \in \mathcal{M}_1(\mathcal{H})}{\operatorname{argmin}} \; \mathbb{E}_{h_i \sim Q} \left[\ell(h_i, z_i) \right] + \frac{\mathrm{KL}(Q \| P_i)}{\lambda} \tag{1}$$

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. 2.3.

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 $S \sim \mu$:*

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

Optimising in λ gives $\lambda = \sqrt{2\log(1/\delta)/mK^2}$ and ensure that:

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

The proof consists in applying Thm. 2.3 with for all i, $Q_i = \hat{Q}_i = P_i$.

Corollary 3.3 quantifies how efficient will our predictions be. Indeed, the left hand side of this bound relates for all i, how good $\hat{Q}_i$ is to predict z_i (on average) which is what $\hat{Q}_i$ is designed for. Note that here, 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. 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. 2.3, 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.1 for a recap). First we remark that our minimisation problem is equivalent to $\operatorname{argmin}_Q \lambda \mathbb{E}_{h_i \sim Q} \left[\ell(h_i, z_i)\right] + \mathrm{KL}(Q \| \hat{Q}_i)$. Then we assume that for any $i, \hat{Q}_i = \mathcal{N}(\hat{m}_i, I_d)$ with $\hat{m}_i \in \mathbb{R}^d$ and we set $\mathcal{L}_i(\hat{m}_i) = \mathbb{E}_{h_i \sim \hat{Q}_i} \left[\ell(h_i, z_i)\right]$. The minimisation problem becomes: $\operatorname{argmin}_{\hat{m}} \lambda \mathcal{L}_i(\hat{m}) + \frac{1}{2}\|\hat{m} - \hat{m}_i\|^2$. And so using the first order Taylor expansion, we use the approximation $\mathcal{L}_i(\hat{m}) \approx \mathcal{L}_i(\hat{m}_i) + \langle \hat{m} - \hat{m}_i, \nabla \mathcal{L}_i(\hat{m}_i) \rangle$ which finally transform our argmin into the following optimisation process: $\hat{m}_{i+1} = \hat{m}_i - \lambda \nabla \mathcal{L}_i(\hat{m}_i)$ which is exactly OGD on the loss sequence $\mathcal{L}_i$. We draw an analogy between the scale parameter λ and the step size η in OGD. the KL term translates the influence of the previous point and the expected loss gives the gradient. This analogy has been already exploited in Shalev-Shwartz [2012] where they approximated $\mathbb{E}_{h_i \sim q_\mu}[\ell(h_i, z_i)] := \bar{L}_i(\mu) \approx \mu^T \nabla \bar{L}_i(\mu_i)$ where μ is their considered online predictive sequence.

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 comes at the cost of additional constraints that do not allow to consider any aggregation strategies contrary to what Corollary 3.1 propose. 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 C 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. 2.3 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:

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

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 C. 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 C We then can use algorithm 1 with Eq. (5), Eq. (6).

Proof is deferred to Appendix D.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 C for a detailed statement of those properties.

Comparison with van der Hoeven et al. [2018]. Theorem 3 of van der Hoeven et al. [2018] recovers OGD from the exponential weights algorithm by taking a sequence of moving distributions being Gaussians with fixed variance which is exactly what we consider here. From these, they retrieve the classical OGD algorithm as well as its classical convergence rate. Let us compare our results with theirs.

First, if we fix a single step η in their bound and assume two traditional assumptions for OGD (a finite diameter D of the convex set and an uniform bound G on the loss gradients), we recover

for the OGD (greedy GD in van der Hoeven et al., 2018) a rate of $\frac{D^2}{2\sigma^2\eta} + \frac{\eta\sigma^2 TG^2}{2}$. This is, up to constants and notation changes, exactly our Ψ_i ($i \in \{1, 2\}$). Also, we notice a difference in the way to use Gaussian distributions: Theorem 3 of van der Hoeven et al. [2018] is based on their Lemma 1 which provides guarantees for the expected regret. This is a clear incentive to consider as predictors the mean of the sucessive Gaussians of interest. On the contrary, Corollary 4.1 involves a supplementary level of randomness by considering predictors h_i drawn from our Gaussians. This additional randomness appears in our optimisation process (algorithm 1). Finally, notice that van der Hoeven et al. [2018] based their whole work on the use of a KL divergence while Corollary 4.1 not only exploit a disintegrated KL (Ψ_1) but also a Rényi α-divergence (Ψ_2). Note that we propose a result only for $\alpha = 2$ for the sake of space constraints but any other value of α leads to another optimisation objective to explore.

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 C. The optimised λ gives in both cases a $O(\sqrt{m \log(1/\delta)})$.

Proof is deferred to Appendix D.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.

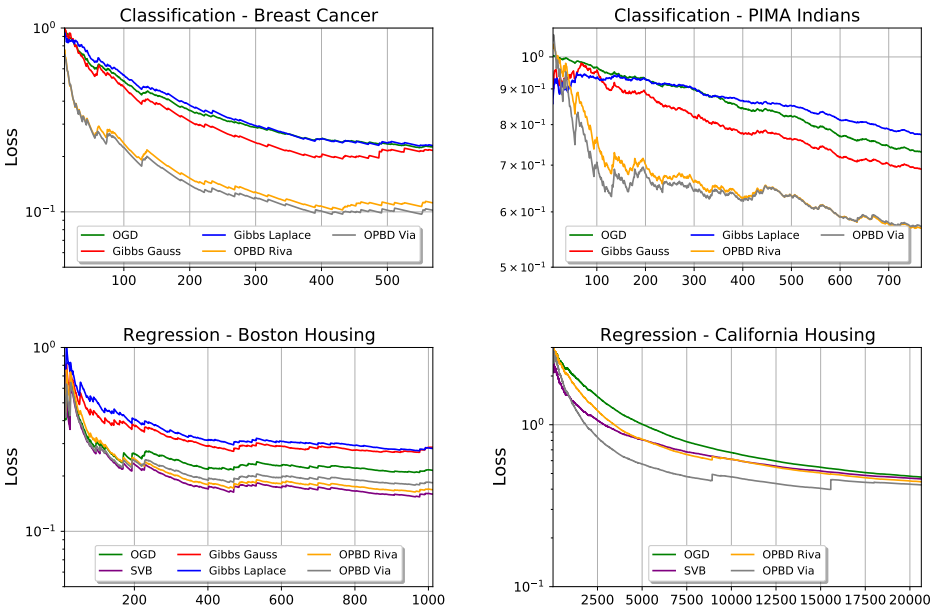

Figure 1: Averaged cumulative losses for all four considered datasets. 'Gibbs Gauss' denotes OPB with Gaussian Prior, 'Gibbs Laplace' denotes OPB with Laplace prior. 'OPBD Riva' denotes OPBD with Ψ_1, 'OPBD Via' denotes OPBD with Ψ_2.

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 $\mathbf{0}_{\mathbb{R}^d}$ 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 $\sum_{i=1}^{t} \ell\left(h_i, z_i\right)/t$ 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. We refer to Appendix E for additional table gathering the error bars of our OPBD methods.

Why do we perform better than OGD? As stated in Sec. 4, OGD can be recovered as a Gaussian approximation of the exponential weights algorithm (EWA). Thus, a legitimate question is why do we perform better than OGD as our OPBD methods are also based on a Gaussian surrogate of EWA? van der Hoeven et al. [2018] only used Gaussians distributions with fixed variance as a technical tool when the considered predictors are the Gaussian means. In our work, we exploited a richer characteristic of our distributions in the sense our predictors are points sampled from our Gaussians and not only the means. This also has consequences in our learning algorithm as at time i of our algorithm 1, our optimisation step involves a noise $\varepsilon_i \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. Thus, we believe that OPBD methods should perform at least as well as OGD. We write 'at least' as we think that the higher flexibility due to this additional level of randomness might result in slightly better empirical performances, as seen on the few datasets in Fig. 1.

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.

Acknowledgements

We thank several anonymous reviewers as well as the Area Chair of the Neurips comitee who provided insigthful comments and suggestions which greatly enhanced the quality of our comparisons with literature and of the discussions around our theorems.

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.

P. Alquier, J. Ridgway, and N. Chopin. On the properties of variational approximations of gibbs posteriors. *Journal of Machine Learning Research*, 17(236):1–41, 2016. URL http://jmlr.org/papers/v17/15-290.html.

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.

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.

D. van der Hoeven, T. van Erven, and W. Kotłowski. The many faces of exponential weights in online learning. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 2067–2092. PMLR, 06–09 Jul 2018. URL https://proceedings.mlr.press/v75/hoeven18a.html.

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 D 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]

A Background

A.1 Reminder on Online Gradient Descent

For the sake of completeness we re-introduce the projected Online Gradient Descent (OGD) on a convex set $\mathcal{K}$. This is a first example of online learning philosophy. It may be the algorithm that applies to the most general setting of online convex optimization. This algorithm, which is based on standard gradient descent from offline optimization, was introduced in its online form by Zinkevich [2003]. In each iteration, the algorithm takes a step from the previous point in the direction of the gradient of the previous cost. This step may result in a point outside of the underlying convex set. In such cases, the algorithm projects the point back to the convex set, i.e. finds its closest point in the convex set. We precise this algorithm works with the assumptions of a convex set $\mathcal{K}$ bounded in diameter by D and of bounded gradients (by a certain G). We also assume here to have a dataset $S = (z_t)_{t=1..T}$ and to be coherent with the online learning philosophy, we assume that for each $t > 0$, we possess a loss function ℓ_t depending on the points $(z_1, ...z_t)$. We present OGD in algorithm 2

Algorithm 2: Projected OGD onto a convex $\mathcal{K}$ with fixed step η.

Parameters : Epoch T, step-size (η)
Initialisation : Convex set $\mathcal{K}$, Initial point $\theta_0 \in \mathcal{K}$, T, step sizes $(\eta_t)_t$

for *each iteration* t *in* $1..T$ **do**
  Compute $f'(\theta_n)$
  Play (observe) θ_t and compute the cost $f_t(\theta_t)$ Update and project

$$\zeta_t = \theta_{t-1} - \eta \nabla \ell_t(\theta_{t-1})$$
$$\theta_t = \Pi_{\mathcal{K}}(\zeta_t)$$

end
Return θ_T

One now defines the notion of regret which is the classical quantity to evaluate the performance of an online algorithm.

Definition A.1. *One defines the* regret *of a decision sequence* (θ_t) *at time* T *w.r.t. a point* θ *as:*

$$Regret_T(\theta) := \sum_{t=1}^{T} \ell_t(\theta_t) - \sum_{t=1}^{T} \ell_t(\theta)$$

Now we state a regret bound which can be found in [Shalev-Shwartz, 2012, Eq 2.5] although we slightly modified the result, which uses additional hypotheses from Hazan [2016].

Proposition A.2. *Assume that* $\mathcal{K}$ *has a fixed diameter* D *and that the gradients of any point is bounded by* G*. Then for any* $\theta \in \mathcal{K}$*, the regret of projected OGD with fixed step* η *satisfies:*

$$Regret_T(\theta) \leq \frac{D^2}{2\eta} + \eta T G^2$$

A.2 About PAC-Bayes learning

We provide here more information about PAC-Bayes learning. We propose a local framework; then expose what APC-Bayes lerning ais to do and finally state two celebrated PAC-Bayes theorems.

An usual framework We first state the following framework:

- $\mathcal{H}$ is a space of considered predictors
- $\mathcal{Z}$ is a data space. z can be an unlabeled data x or a couple (x, y) of a point with its label. We assume that μ is a distribution over $\mathcal{Z}$ which rules the distribution of our data.
- $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$ is a loss function i.e. the learning objective we want to minimise.

- $S = (z_1, ...z_m)$ an iid dataset following μ.
- The generalisation risk for $h \in \mathbb{H}$: $R(h) = \mathbb{E}_{z\sim\mu}[\ell(h, z)]$.
- The empirical risk $R_m(h) = \frac{1}{m}\sum_{i=1}^m \ell(h, z_i)$.

What is PAC-Bayes learning? PAC-Bayes learning is about learning a meaningful data-dependent posterior distribution Q from a (classically data-free) prior P without necessarily exploiting the Bayes formula. PAC-Bayes learning controls the *expected generalisation error*:

$$\mathbb{E}_{h\sim Q}[R(h)] := \mathbb{E}_{h\sim Q}\mathbb{E}_{z\sim\mu}[\ell(h, z)],$$

which is the averaged error that would make a predictor drawn from our posterior distribution on a new point (usually this objective holds under an iid assumption on our dataset).

Two classical theorems. We state below two celebrated PAC-Bayesian results: the McAllester bound enriched with Maurer's remark as stated in [Guedj, 2019, Thm.1] and Catoni's bound (Catoni, 2007, Thm 1.2.6). Those theorems both holds with the assumptions of iid data and loss bounded by 1. Note that in Thm. B.1, we also proposed another bound which is a corollary of Catoni's one.

Theorem A.3 (McAllester' bound)**.** *For any prior distribution P, we have with probability $1-\delta$ over the m-sample S, for any posterior distribution Q such that $Q \ll P$:*

$$\mathbb{E}_{h\sim Q}\left[R(h)\right] \leq \mathbb{E}_{h\sim Q}\left[R_m(h)\right] + \sqrt{\frac{KL(Q,P) + \log(2\sqrt{m}/\delta)}{2m}},$$

where KL is the Kullback-Leibler divergence.

Theorem A.4. *For any prior distribution P, any $\lambda > 0$, we have with probability $1-\delta$ over the m-sample S, for any posterior distribution Q such that $Q \ll P$:*

$$\mathbb{E}_{h\sim Q}\left[R(h)\right] \leq \frac{1 - \exp\left\{-\frac{\lambda\mathbb{E}_{h\sim Q}[R_m(h)]}{m} - \frac{KL(Q||P)-\log(\delta)}{m}\right\}}{1-\exp\left(-\frac{\lambda}{m}\right)},$$

where KL is the Kullback-Leibler divergence.

B Discussion about Thm. 2.3

B.1 Comparison with classical PAC-Bayes

The goal of this section is to show how good Thm. 2.3 compared to a naive approach which consists in applying classical PAC-Bayes results sequentially. The interest of this section is twofold:

- First, presenting a classical PAC-Bayes result extracted and adapted from Alquier et al. [2016] which is formally close to what we propose.
- Second, showing that a naive (yet natural) approach to obtain online PAC-Bayes bound leads to a deteriorated bound.

We first state our PAC-Bayes bound of interest.

Theorem B.1 (Adapted from Alquier et al. [2016], Thm 4.1)**.** *Let $S = (z_1, ..., z_m)$ be an iid sample from the same law μ_0. For any data-free prior P, for any loss function ℓ bounded by K, any $\lambda > 0, \delta \in]0; 1[$, one has with probability $1-\delta$ for any posterior $Q \in \mathcal{M}_1(\mathcal{H})$:*

$$\mathbb{E}_{h\sim Q}\mathbb{E}_{z\sim\mu_0}[\ell(h, z)] \leq \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{h\sim Q}[\ell(h, z_i)] + \frac{\mathrm{KL}(Q\|P) + \log(1/\delta)}{\lambda} + \frac{\lambda K^2}{2m}$$

Remark B.2. *Two remarks about this result:*

- *Thm. B.1 is a particular case of the original theorem from Alquier et al. [2016] as we take the case of a bounded loss which implies the subgaussianity of the random variables $\ell(., z_i)$ and then allows us to recover the factor $\frac{\lambda K^2}{m}$*

- *This theorem is derived from Catoni [2007] and constitutes a good basis to compare ourselves with as it similar formally similar.*

Naive approach A naive way to obtain OPB bounds is to apply m times Thm. B.1 (one per data) on batches of size 1 and then summing up the associated bounds. Thus one has the benefits of classical PAC-Bayes bound without having no more the need of data-free priors nor the iid assumption. The associated result is stated below:

Theorem B.3. *For any distributions $\mu_1, ..., \mu_m$ over $\mathcal{Z}$ (such that $z_i \sim \mu_i$), any $\lambda > 0$ and any online predictive sequence (used as priors) (P_i), the following holds with probability $1-\delta$ over the sample $S \sim \mu$ for any posterior sequence (Q_i) :*

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

Recall that here again we assimilate the stochastic kernels Q_i, P_i to the data-dependent distributions $Q_i(S, .), P_i(S, .)$

Proof. First of all, for any i, we apply Thm. B.1 m to the batch $\{z_i\}$. This allows us to consider P_i as a prior as it does not depend on the current data. We then have, taking $\delta' = \delta/m$, for any $i \in \{1..m\}$ with probability $1 - \delta/m$:

$$\mathbb{E}_{h_i \sim Q_i}\left[\mathbb{E}_{z_i \sim \mu_i}[\ell(h_i, z_i)]\right] \leq \mathbb{E}_{h_i \sim Q_i}\left[\ell(h_i, z_i)\right] + \frac{\mathrm{KL}(Q_i \| P_i)}{\lambda} + \frac{\lambda K^2}{2} + \frac{\log(m/\delta)}{\lambda}$$

Then, taking an union bound on those m events ensure us that with probability $1-\delta$, for any $i \in \{1..m\}$:

$$\mathbb{E}_{h_i \sim Q_i}\left[\mathbb{E}_{z_i \sim \mu_i}[\ell(h_i, z_i)]\right] \leq \mathbb{E}_{h_i \sim Q_i}\left[\ell(h_i, z_i)\right] + \frac{\mathrm{KL}(Q_i \| P_i)}{\lambda} + \frac{\lambda K^2}{2} + \frac{\log(m/\delta)}{\lambda}$$

Finally, summing those m inequalities ensure us the final result with probability $1-\delta$.

□

Comparison between Thm. 2.3 and Thm. B.3 Three points are noticeable between those two theorems:

- First of all, the main issue with Thm. B.3 is that has a strongly deteriorated rate of $O\left(\frac{m \log(m/\delta)}{\lambda}\right)$ instead of the rate in $O\left(\frac{\log(1/\delta)}{\lambda}\right)$ proposed in Thm. 2.3. More precisely, the problem is that we do not have a sublinear bound: one cannot ensure any learning through time. This point justifies the need of the heavy machinery exploited in Thm. 2.3 proof as it allows a tighter convergence rate.
- The second point point lies in the controlled quantity on the left hand-side of the bound. Thm. B.3 controls $A := \sum_{i=1}^{m} \mathbb{E}_{h_i \sim Q_i}\left[\mathbb{E}_{z_i \sim \mu_i}[\ell(h_i, z_i)]\right]$ instead of $B := \sum_{i=1}^{m} \mathbb{E}_{h_i \sim Q_i}\left[\mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}]\right]$.
  A is a less dynamic quantity than B in the sense that it does not imply any evolution through time, it just considers global expectations. Doing so, A does not take into account that at each time step we have acces to all te past data to predict the future, this may explain the deteriorated convergence rate. Thus B, which appears to be a suitable quantity to control to perform online PAC-Bayes (see Appendix B.2 for additional explanations)
- Finally, an interesting point is that in Thm. B.3 the bound, while looser, holds unformly for any posterior sequence contrary to Thm. 2.3 which holds only for a specific posterior sequence. This point will have a consequence for optimisation. We will come back later on this in Appendix B.3.

B.2 A deeper analysis of Thm. 2.3

This section includes discussion about our proof technique and why all the assumptions made are necessary. We also propose a short discussion about the benefits and limitations of an online PAC-Bayesian framework as well as a deeper reflexion about the new term our bound introduce.

Why do we need an online predictive sequence as priors? This condition is fully exploited when dealing with the exponential moment ξ_m in the proof (see Lemma D.2 proof). Indeed, the fact of having P_i being $\mathcal{F}_{i-1}$-measurable is essential to apply conditional Fubini (Lemma D.3). Note that the condition $\forall i, P_{i-1} \ll P_i$ is not necessary as the weaker condition $\forall i, P_1 \ll P_i$ would suffice here. However, note that when we particularise our theorem, for instance if we choose in Corollary 3.1 $P_i = \hat{Q}_i$, one recovers the condition $\hat{Q}_i \ll \hat{Q}_{i+1}$ to have finite KL divergences. Hence the interest of taking directly an online predictive sequence.

About the boundedness assumption The only moment where we invoke the boundedness assumption is in Lemma D.2's proof where we apply the conditionnal Hoeffding lemma. This lemma actually translates that the sequence of r.v. $(\ell(., z_i)_{i=1..m}$ is *conditionally subgaussian* wrt the past i.e for any i, $h_i \in \mathcal{H}; \lambda \in \mathbb{R}$:

$$\mathbb{E}[\exp(\lambda \tilde{\ell}_i(h_i, z_i)) \mid \mathcal{F}_{i-1}] \leq \exp\left(\frac{\lambda^2 K^2}{2}\right)$$

where $\tilde{\ell}_i(h_i, z_i) = \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] - \ell(h_i, z_i)$.

This condition is the one truly involved in our heavy machinery. However, we chose to restrict ourselves to the stronger assumption of bounded loss function for the sake of clarity. However, an interesting open direction is to find whether there exists concrete classes of unbounded losses which may satisfy either conditional subgaussianity or others conditions (such as conditional Bernstein condition for instance).

Reflections about the left hand side of Thm. 2.3. We study in this paragraph the following term

$$B := \sum_{i=1}^{m} \mathbb{E}_{h_i \sim Q_i} \left[\mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}]\right]$$

has naturally arisen in our work as the right term to compare our empirical loss with to perform the conditional Hoeffding lemma. Taking a broader look, we now interpret this term as the right quantity to control if one wants to perform online PAC-Bayes learning. Indeed this term is a 'best of both world' quantity bridging PAC-Bayes and online learning:

- From the PAC-Bayes point of view one keeps the control on average (cf the conditional expectation in B) on a novel data drawn at each time step. This point is crucial in the PAC-Bayes literature as our posteriors are designed to generalise well to unseen data.
- From the Online Learning point of view, one keeps the control of a sequence of points generated from an online algorithm. Because an online learning algorithm generate a prediction for future points while having access to past data, the conditional expectation in B translates this.

Finally this conditional expectation appears to be a good tradeoff between the classical expectation on data appearing in the PAC-Bayes literature (see e.g. Thm. B.1) and the local control that we have in online learning by only dealing with the performance of a sequence of points generated from a learning algorithm (see e.g. proposition A.2)

About the interest of an Online PAC-Bayesian framework The main shift our work does with classical online learning literature is that it does not consider the celebrated regret but instead focuses on B which is a cumulative expected loss conditionned to the past. This shift does not invalidate our work but put some relief to hte guarantees Online PAC-Bayes learning can provide that Online Learning cannot and reversely.

- Online PAC-Bayes ensure a good potential for generalisation as it deals with the control of conditional expectation. This can be useful if one wants to deal with a periodic process for instance.
- Online Learning through the regret compares the studied sequence of predictors (typically generated from an online learning algorithm) and tries to compare it to the best fixed strategy (static regret) or the best dynamic one (dynamic regret). In this way, OL algorithms want to ensure that their predictions are closed from the optimal solution. This point is not guaranteed by our online PAC-Bayesian study.
- However the limitations of online learning can arise if the studied problem has a huge variance (for instance micro-transactions in finance). In this case those algorithms can follow an unpredictable optimisation route while PAC-Bayes still ensure a good performance on average (knowing the past) in this case.
- Finally, we want to emphasize that PAC-Bayesian learning circumvent a problem of *memoryless learning* which appears in classical OL algorithms. For instance, the OGD algorthm (see Appendix A.1) uses once a data and do not memorise it for further use. This problem does not happen in Online PAC-Bayes learning. Indeed, we take the example of the procedure Eq. (3) which generates Gibbs posterior which keep in mind the influence of past data.

B.3 Thm. 2.3 and optimisation

In this section we discuss about the way Thm 2.2 can be thought in the framework of an optimisation process as we did in Secs. 3 and 4.

A significant change compared to classical PAC-Bayes Thm. 2.3 holds 'for any posterior sequence (Q_i) the following holds with probability $1-\delta$ over the sample $S \sim \mu$ ' while most classical PAC-Bayesian results such that Thm. B.1 holds 'with probability $1-\delta$ over the sample $S \sim \mu$ for any posterior Q'. This change is significant as our theorem does not control simultaneaously all possible sequences of posteriors but only holds for one. Thus, Thm. 2.3 has to be seen as a local or pointwise theorem and not as a global one. In classical PAC-Bayes, this local behavior is a brake on the optimisation process. But as we develop below, it is not the case in our online framework.

Thm. 2.3 is compatible with online optimisation We first recall that classically, an online algorithm like OGD (see Appendix A.1) performs one optimisation step per arriving data. Thus, at time m, such algorithm will perform m optimisation steps and generate m predictors. Similarly the OPB algorithm of Eq. (1) generates m distribution in m time steps.

We insist on the fact that, Thm. 2.3 **and all its corollaries throughout our paper are valid for a sequence of m posteriors and not only a single one.** A key point is that whatever the number m of data, our theoretical guarantee wil still be valid for m posterior distributions with the approximation term $\log(1/\delta)$ (and not $\log(m/\delta)$ as an union bound would provide for a classical PAC-Bayes theorem).

For this reason, given an online PAC-Bayes algorithm, Thm. 2.3 is suited for optimisation. Indeed, having a bound valid for a sequence of posteriors ensures guarantees for a single run of our OPB algorithm. This point is crucial to bridge a link with online learning as regret bounds (e.g. proposition A.2) also provide guarantees for a single sequence of predictors. In online learning however, those guarantees are mainly deterministic (because based on convex optimisation properties) but not totally: the recent work of Wintenberger [2021] proposed PAC regret bounds for its general Stochastic Online Convex Optimisation framework.

An interesting open challenge is to overcome the pointwise behavior of our theorem, for that, we need to rethought [Rivasplata et al., 2020, Thm 2.1] as this basis is pointwise itself. Given we consider a sequence of data-dependent priors one cannot apply the classical change of measure inequality to ensure guarantees holding uniformly on posterior sequences.

A crucial point: having an explicit OPB/OPBD algorithm In our previous paragraph we said that our bound were suitable for optimisation given an OPB/OPBD algorithm. We now provide some precision about this point. All the procedures provided in the paper (i.e. Eq. (1), algorithm 1) take into account an update phase implying an argmin. Luckily for our procedures, this argmin is explicit:

- For the OPB algorithm of Eq. (1), the argmin is solved thanks to the variational formulation of the Gibbs posterior
- For OPBD algorithms, given the explicit choices of Ψ given in Corollary 4.1, argmin becomes explicit when one has a derivable loss function.

In both cases, this explicit argmin ensure our procedure of interest generates explictly a single posterior per time step: we have a well-defined sequence of m posteriors at time m. Doing so the guarantees of Thm. 2.3 holds for this sequence.

C A reminder on PAC-Bayesian disintegrated bounds

We present two PAC-Bayesian disintegrated bounds valid with data-dependent priors (i.e. any stochastic kernels).

- The first one is Th. 1) i) from Rivasplata et al. [2020] which provides a disintegrated version of Thm. 2.3.
- The second one is Thm 2. from Viallard et al. [2021] which involves Rényi divergence instead of the classical KL. Note that this bound has originally been stated for data-indepedent prior, which is why we revisit the proof to adapt it to the stochastic kernel framework.

Proposition C.1 (Th 1) i) Rivasplata et al. [2020]). *Let $P \in \mathcal{M}_1(\mathcal{S})$, $Q^0 \in \texttt{Stoch}(\mathcal{Z}^m, \mathcal{F})$. Let $f : \mathcal{S} \times \mathcal{H} \to \mathbb{R}$ be any measurable function. Then for any $Q \in \texttt{Stoch}(\mathcal{Z}^m, \mathcal{F})$ and any $\delta \in (0,1)$, with probability at least $1-\delta$ over the random draw of $S \sim P$ and $h \sim Q_S$, we have:*

$$f(S,h) \leq \log\left(\frac{dQ_S}{dQ_S^0}(h)\right) + \log(\xi_m/\delta).$$

where $\xi_m := \int_{\mathcal{S}} \int_{\mathcal{H}} e^{f(s,h)} Q_s^0(dh) P(ds)$ and $\frac{dQ_S}{dQ_S^0}$ is the Radon Nykodym derivative of Q_S w.r.t. Q_S^0.

Proposition C.2 (Adapted from Th. 2 of Viallard et al. [2021]). *Let $\mu \in \mathcal{M}_1(\mathcal{S})$, $Q^0 \in \texttt{Stoch}(\mathcal{Z}^m, \mathcal{F})$. Let $\alpha > 1$ and $f : \mathcal{S} \times \mathcal{H} \to \mathbb{R}^+$ be any measurable function.*

Then for any $Q \in \texttt{Stoch}(\mathcal{Z}^m, \mathcal{F})$ such that for any $S \in \mathcal{Z}^m, Q_S >> Q_S^0,\ Q_S^0 >> Q_S$ and any $\delta \in (0,1)$, with probability at least $1-\delta$ over the random draw of $S \sim \mu$ and $h \sim Q_S$, we have:

$$\frac{\alpha}{\alpha-1}\log(f(S,h)) \leq \frac{2\alpha-1}{\alpha-1}\log\frac{2}{\delta} + D_\alpha\left(Q_S \| Q_S^0\right) + \log\left(\mathop{\mathbb{E}}_{S'\sim\mu} \mathop{\mathbb{E}}_{h'\sim Q_{S'}^0} f\left(S', h'\right)^{\frac{\alpha}{\alpha-1}}\right)$$

where $D_\alpha(Q\|P) = \frac{1}{\alpha-1}\log\left(\mathbb{E}\left[\mathbb{E}_{h\sim P}\left(\frac{dQ}{dP}(h)\right)^\alpha\right]\right)$ is the Rényi diverence of order α.

Note that Viallard et al. original bound only stand for data-free priors and i.i.d data. However it appears their proof works with any stochastic kernel as prior and any distribution over the dataset. We propose below an adaptation of their proof below to fit with those more general assumptions.

C.1 Proof of proposition C.2

Proof. For any sample S and any stochastic kernel Q, note that $f(S,h)$ is a non-negative random variable. Hence, from Markov's inequality we have

$$\mathop{\mathbb{P}}_{h\sim Q_S}\left[f(S,h) \leq \frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S} f\left(S,h'\right)\right] \geq 1-\frac{\delta}{2} \iff \mathop{\mathbb{E}}_{h\sim Q_S} \mathbb{1}\left[f(S,h) \leq \frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S} f\left(S,h'\right)\right] \geq 1-\frac{\delta}{2}$$

Taking the expectation over $S \sim \mu$ to both sides of the inequality gives

$$\mathop{\mathbb{E}}_{S\sim\mu} \mathop{\mathbb{E}}_{h\sim Q_S} \mathbb{1}\left[f(S,h) \leq \frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S} f(S,h')\right] \geq 1-\frac{\delta}{2}$$
$$\iff \mathop{\mathbb{P}}_{S\sim\mu, h\sim Q_S}\left[f(S,h) \leq \frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S} f(S,h')\right] \geq 1-\frac{\delta}{2}.$$

Taking the logarithm to both sides of the equality and multiplying by $\frac{\alpha}{\alpha-1} > 0$, we obtain

$$\mathop{\mathbb{P}}_{S\sim\mu, h\sim Q_S}\left[\frac{\alpha}{\alpha-1}\log(f(S,h)) \leq \frac{\alpha}{\alpha-1}\log\left(\frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S} f(S,h')\right)\right] \geq 1-\frac{\delta}{2}.$$

We develop the right side of the inequality in the indicator function and make the expectation of the hypothesis over Q_S^0 our "prior" stochadtic kernel appears. Indeed, because for any $S \in \mathcal{S}, Q_S >> Q_S^0$ and $Q_S^0 >> Q_S$ one can write properly $\frac{dQ_S}{dQ_S^0}$ and $\frac{dQ_S^0}{dQ_S} = \left(\frac{dQ_S}{dQ_S^0}\right)^{-1}$ the Radon-Nykodym derivatives. Thus we have

$$\begin{aligned}
&\frac{\alpha}{\alpha-1}\log\left(\frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S} f(S,h')\right) \\
&\qquad = \frac{\alpha}{\alpha-1}\log\left(\frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S}\frac{dQ_S}{dQ_S^0}(h')\frac{dQ_S^0}{dQ_S}(h')f(S,h')\right) \\
&\qquad\qquad = \frac{\alpha}{\alpha-1}\log\left(\frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S^0}\frac{dQ_S}{dQ_S^0}(h')f(S,h')\right).
\end{aligned}$$

Remark that $\frac{1}{r}+\frac{1}{s}=1$ with $r=\alpha$ and $s=\frac{\alpha}{\alpha-1}$. Hence, we can apply Hölder's inequality:

$$\mathop{\mathbb{E}}_{h'\sim Q_S^0}\frac{dQ_S}{dQ_S^0}(h')f(S,h') \leq \left[\mathop{\mathbb{E}}_{h'\sim Q_S^0}\left(\frac{dQ_S}{dQ_S^0}(h')\right)^{\alpha}\right]^{\frac{1}{\alpha}}\left[\mathop{\mathbb{E}}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}}\right]^{\frac{\alpha-1}{\alpha}}.$$

Then, by taking the logarithm; adding $\log\left(\frac{2}{\delta}\right)$ and multiplying by $\frac{\alpha}{\alpha-1} > 0$ to both sides of the inequality, we obtain

$$\begin{aligned}
&\frac{\alpha}{\alpha-1}\log\left(\frac{2}{\delta}\mathop{\mathbb{E}}_{h'\sim Q_S^0}\frac{dQ_S}{dQ_S^0}(h')f(S,h')\right) \\
&\qquad \leq \frac{\alpha}{\alpha-1}\log\left(\frac{2}{\delta}\left[\mathop{\mathbb{E}}_{h'\sim Q_S^0}\left(\frac{dQ_S}{dQ_S^0}(h')\right)^{\alpha}\right]^{\frac{1}{\alpha}}\left[\mathop{\mathbb{E}}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}}\right]^{\frac{\alpha-1}{\alpha}}\right) \\
&\qquad = \frac{1}{\alpha-1}\log\left(\mathop{\mathbb{E}}_{h'\sim Q_S^0}\left[\frac{dQ_S}{dQ_S^0}(h')\right]^{\alpha}\right) + \frac{\alpha}{\alpha-1}\log\frac{2}{\delta} + \log\left(\mathop{\mathbb{E}}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}}\right) \\
&\qquad\qquad = D_\alpha\left(Q_S\|Q_S^0\right) + \frac{\alpha}{\alpha-1}\log\frac{2}{\delta} + \log\left(\mathop{\mathbb{E}}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}}\right)
\end{aligned}$$

From this inequality, we can deduce that

$$\begin{aligned}
\mathop{\mathbb{P}}_{S\sim\mu, h\sim Q_S}&\left[\frac{\alpha}{\alpha-1}\log(f(S,h)) \leq D_\alpha\left(Q_S\|Q_S^0\right) + \frac{\alpha}{\alpha-1}\log\frac{2}{\delta} + \log\left(\mathop{\mathbb{E}}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}}\right)\right] \\
&\geq 1-\frac{\delta}{2}. \quad (8)
\end{aligned}$$

Note that $\mathbb{E}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}}$ is a non-negative random variable, hence, we apply Markov's inequality to have

$$\mathop{\mathbb{P}}_{S\sim\mu}\left[\mathop{\mathbb{E}}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}} \leq \frac{2}{\delta}\mathop{\mathbb{E}}_{S'\sim\mu}\mathop{\mathbb{E}}_{h'\sim Q_S^0} f(S',h')^{\frac{\alpha}{\alpha-1}}\right] \geq 1-\frac{\delta}{2}.$$

Since the inequality does not depend on the random variable $h \sim Q_S$, we have

$$\mathbb{P}_{S\sim\mu}\left[\mathbb{E}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}} \leq \frac{2}{\delta}\mathbb{E}_{S'\sim\mu}\mathbb{E}_{h'\sim Q_S^0} f(S',h')^{\frac{\alpha}{\alpha-1}}\right]$$
$$= \mathbb{E}_{S\sim\mu}\mathbb{1}\left[\mathbb{E}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}} \leq \frac{2}{\delta}\mathbb{E}_{S'\sim\mu}\mathbb{E}_{h'\sim Q_S^0} f(S',h')^{\frac{\alpha}{\alpha-1}}\right]$$
$$= \mathbb{E}_{S\sim\mu}\mathbb{E}_{h\sim Q_S}\mathbb{1}\left[\mathbb{E}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}} \leq \frac{2}{\delta}\mathbb{E}_{S'\sim\mu}\mathbb{E}_{h'\sim Q_S^0} f(S',h')^{\frac{\alpha}{\alpha-1}}\right]$$
$$= \mathbb{P}_{S\sim\mu,h\sim Q_S}\left[\mathbb{E}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}} \leq \frac{2}{\delta}\mathbb{E}_{S'\sim\mu}\mathbb{E}_{h'\sim Q_S^0} f(S',h')^{\frac{\alpha}{\alpha-1}}\right].$$

Taking the logarithm to both sides of the inequality and adding $\frac{\alpha}{\alpha-1}\log\frac{2}{\delta}$ give us

$$\mathbb{P}_{S\sim\mu,h\sim Q_S}\left[\mathbb{E}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}} \leq \frac{2}{\delta}\mathbb{E}_{S'\sim\mu}\mathbb{E}_{h'\sim Q_S^0} f(S',h')^{\frac{\alpha}{\alpha-1}}\right] \geq 1-\frac{\delta}{2} \iff$$
$$\mathbb{P}_{S\sim\mu,h\sim Q_S}\left[\frac{\alpha}{\alpha-1}\log\frac{2}{\delta} + \log\left(\mathbb{E}_{h'\sim Q_S^0} f(S,h')^{\frac{\alpha}{\alpha-1}}\right) \leq \right.$$
$$\left.\frac{2\alpha-1}{\alpha-1}\log\frac{2}{\delta} + \log\left(\mathbb{E}_{S'\sim\mu}\mathbb{E}_{h'\sim Q_S^0} f(S',h')^{\frac{\alpha}{\alpha-1}}\right)\right] \geq 1-\frac{\delta}{2}. \quad (9)$$

Combining Equation Eq. (8) and Eq. (9) with a union bound gives us the desired result. □

D Proofs

D.1 Proof of Thm. 2.3

Background We first recall [Rivasplata et al., 2020, Thm 2].

Theorem D.1. *Let $\mu \in \mathcal{M}_1(\mathcal{S})$, $Q^0 \in \texttt{Stoch}(\mathcal{S},\mathcal{F})$. Let k be a positive integer, any $A : \mathcal{S}\times\mathcal{H} \to \mathbb{R}^k$ a measurable function and $F : \mathbb{R}^k \to \mathbb{R}$ be a convex function . Then for any $Q \in \texttt{Stoch}(\mathcal{S},\mathcal{F})$ and any $\delta \in (0,1)$, with probability at least $1-\delta$ over the random draw of $S \sim \mu$ we have*

$$F\left(Q_S\left[A_S\right]\right) \leq \mathrm{KL}\left(Q_S \| Q_S^0\right) + \log(\xi_m/\delta).$$

where $\xi_m := \int_{\mathcal{S}}\int_{\mathcal{H}} e^{f(s,h)} Q_s^0(dh)P(ds)$ and $Q_S[A_S] := Q_S[A(S,.)] = \int_{\mathcal{H}} A(S,h)Q_S(dh)$.

Proof of Thm. 2.3. To fully exploit the generality of Thm. D.1, we aim to design a m-tuple of probabilities. Thus, our predictor set of interest is $\mathcal{H}_m := \mathcal{H}^{\otimes m}$ and then, our predictor h is a tuple $(h_1,..,h_m) \in \mathcal{H}$. Throughout our study, our stochastic kernels Q, Q^0 will belong to the specific class $\mathcal{C}$ defined below:

$$\mathcal{C} := \{Q \mid \exists (Q_i)_{i=1..m} \text{s.t. } \forall S,\ Q(S,.) = Q_1(S,.)\otimes ... \otimes Q_m(S,.)\} \quad (10)$$

Thus our kernels are such that conditionally to a given sample, our predictors $(h_1,...,h_m)$ are drawn independently.

We now apply Thm. D.1. To do so, we consider the following function $A : \mathcal{S}\times\mathcal{H}_m \to \mathbb{R}^2$ such that $\forall S = (z_i)_{i=1..m}, h = (h_i)_{i=1..m} \in \mathcal{S}\times\mathcal{H}_m$:

$$A(S,h) = \left(\sum_{i=1}^m \mathbb{E}[\ell(h_i,z_i) \mid \mathcal{F}_{i-1}], \sum_{i=1}^m \ell(h_i,z_i)\right)$$

A is indeed measurable in both of its variables. For a fixed $\lambda > 0$, we set the function F to be $F(x,y) = \lambda(x-y)$.

The only thing left to set up is our stochastic kernels. To do so, let $P = (P_1, ...P_m)$ be an online predictive sequence, we then define $Q^0 \in \mathcal{C}$ (defined in Eq. (10)) s.t. for any sample S, $Q^0_S = P_1(S,.) \otimes ... \otimes P_m(S,.)$. We also fix $Q_1, ..., Q_m$ to be any (posterior) stochastic kernels and similarly we define the stochastic kernel $Q \in \mathcal{C}$ such that for any sample S, $Q(S,.) = Q_1(S,.) \otimes ... \otimes Q_m(S,.)$.

From now, we fix a dataset S and, for the sake of clarity, we assimilate in what follows the stochastic kernels Q_i, P_i to the data-dependent distributions $Q_i(S,.), P_i(S,.)$ (i.e. we drop the dependency in S).

Under those choices, one has:

$$
\begin{aligned}
Q_S[A_S] &= \int_{h \in \mathcal{H}_m} A(S,h) Q_S(dh_1, ..., dh_m) \\
&= \left(\int_{h \in \mathcal{H}_m} \sum_{i=1}^m \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] Q_S(dh_1, ..., dh_m), \int_{h \in \mathcal{H}_m} \sum_{i=1}^m \ell(h_i, z_i) Q_S(dh_1, ..., dh_m) \right)
\end{aligned}
$$

Furthermore, $Q \in \mathcal{C}$, thus $Q_S(dh_1, ..., dh_m) = \Pi_{i=1}^m Q_i(dh_i)$ so:‘

$$
Q_S[A_S] = \left(\sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}[\mathbb{E}\left[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}\right]], \sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}[\ell(h_i, z_i)] \right)
$$

Finally:

$$
F(Q_S[A_S]) = \lambda \left(\sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}[\mathbb{E}\left[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}\right]] - \sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}[\ell(h_i, z_i)] \right)
$$

Applying Thm. D.1 and re-organising the terms gives us with probability $1 - \delta$:

$$
\sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}\left[\mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}]\right] \leq \sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}[\ell(h_i, z_i)] + \frac{KL(Q_S \| Q^0_S)}{\lambda} + \frac{\log(\xi_m/\delta)}{\lambda}
$$

Thus:

$$
\sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}\left[\mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}]\right] \leq \sum_{i=1}^m \mathbb{E}_{h_i \sim Q_i}[\ell(h_i, z_i)] + \sum_{i=1}^m \frac{KL(Q_i \| P_i)}{\lambda} + \frac{\log(\xi_m/\delta)}{\lambda} \quad (11)
$$

The last line holding because for a fixed S, $Q_S = Q_1 \otimes ... \otimes Q_m$ and $Q^0_S = P_1 \otimes ... \otimes P_m$.

The last term to control is

$$
\xi_m = \mathbb{E}_S \left[\mathbb{E}_{h_1, ..., h_m \sim Q^0_S} \left[\exp\left(\lambda \sum_{i=1}^m \tilde{\ell}_i(h_i, z_i) \right) \right] \right]
$$

with $\tilde{\ell}_i(h_i, z_i) = \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] - \ell(h_i, z_i)$. Hence the following lemma.

Lemma D.2. *One has for any m, $\xi_m \leq \exp\left(\frac{\lambda^2 m K^2}{2}\right)$ with K bounding ℓ.*

The proof of this lemma is deferred to Appendix D.1.1

To conclude the proof, we just bound ξ_m by the result of Lemma D.2 within Eq. (11). □

D.1.1 Proof of Lemma D.2

Proof of Lemma D.2. We prove our result by recursion: for $m = 1$, $S = z_1$ and one knows that P_1 is $\mathcal{F}_0$ measurable yet it does not depend on S. Thus for any $h_1 \in \mathcal{H}$, $\mathbb{E}[\ell(h_1, z_1) \mid \mathcal{F}_0] = \mathbb{E}[\ell(h_1, z_1)]$.

We then has:

$$\begin{aligned}\xi_1 &= \mathbb{E}_S \mathbb{E}_{h_1 \sim P_1}[\tilde{\ell}_1(h_1, z_1)] \\ &= \mathbb{E}_{h_1 \sim P_1} \mathbb{E}_S[\tilde{\ell}_1(h_1, z_1)] && \text{by Fubini} \\ &\leq \exp \frac{\lambda^2 K^2}{2}\end{aligned}$$

The last line holding because for any $h_1 \in \mathcal{H}$, $\tilde{\ell}_1(h_1, z_1)$ is a centered variable belonging in $[-K, K]$ a.s. and so one can apply Hoeffding's lemma to conclude.

Assume the result is true at rank $m - 1 \geq 0$. We then has to prove the result at rank m. Our strategy consists in conditioning by $\mathcal{F}_{m-1}$ within the expectation over S:

$$\xi_m = \mathbb{E}_S \left[\mathbb{E}_{h_1, \ldots, h_m \sim Q_S^0} \left[\exp \left(\lambda \sum_{i=1}^m \tilde{\ell}_i(h_i, z_i) \right) \right] \right]$$

First, we use that $Q^0 \in \mathcal{C}$, thus $Q_S^0 = P_1 \otimes ... \otimes P_m$ (i.e. our data are drawn independently for a given S):

$$= \mathbb{E}_S \left[\Pi_{i=1}^m \mathbb{E}_{h_i \sim P_i} \left[\exp \left(\lambda \tilde{\ell}_i(h_i, z_i) \right) \right] \right]$$

We now condition by $\mathcal{F}_{m-1}$ and use that $\Pi_{i=1}^{m-1} \mathbb{E}_{h_i \sim P_i} \left[\exp \left(\lambda \tilde{\ell}_i(h_i, z_i) \right) \right]$ is a $\mathcal{F}_{m-1}$-measurable r.v.

$$\xi_m = \mathbb{E}_S \left[\Pi_{i=1}^{m-1} \mathbb{E}_{h_i \sim P_i} \left[\exp \left(\lambda \tilde{\ell}_i(h_i, z_i) \right) \right] \mathbb{E} \left[\mathbb{E}_{h_m \sim P_m} [\exp(\lambda \tilde{\ell}_m(h_m, z_m))] \mid \mathcal{F}_{m-1} \right] \right]$$

Now our next step is to use a variant of Fubini valid for $\mathcal{F}_{m-1}$- measurable measures.

Lemma D.3 (Conditional Fubini)**.** *Let $f : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$. For a sigma-algebra $\mathcal{F}$ over $\mathcal{Z}$ and a measure P over $\mathcal{H}$ such that*

- *P is a $\mathcal{F}$-measurable r.v.*
- *There exists a constant measure (a.s.) P_0 such that $P >> P_0$.*

Then one has almost surely, for any r.v. z over $\mathcal{Z}$:

$$\mathbb{E} \left[\mathbb{E}_{h \sim P}[f(h, z)] \mid \mathcal{F} \right] = \mathbb{E}_{h \sim P} \left[\mathbb{E}[f(h, z) \mid \mathcal{F}] \right]$$

The proof of this lemma lies at the end of this section.

We then fix $\mathcal{F} = \mathcal{F}_{m-1}$ and $f(h, z) = \exp(\lambda \tilde{\ell}_i(h, z))$. Furthermore, because we assumed the sequence (P_i) to be an online predictive sequence, P_m is $\mathcal{F}_{m-1}$-measurable and $P_m >> P_1$ with P_1 a data-free prior. One then applies Lemma D.3:

$$\mathbb{E} \left[\mathbb{E}_{h_m \sim P_m} [\exp(\lambda \tilde{\ell}_m(h_m, z_m))] \mid \mathcal{F}_{m-1} \right] = \mathbb{E}_{h_m \sim P_m} \left[\mathbb{E}[\exp(\lambda \tilde{\ell}_m(h_m, z_m)) \mid \mathcal{F}_{m-1}] \right].$$

Yet, injecting this result onto ξ_m provides:

$$\xi_m = \mathbb{E}_S \left[\Pi_{i=1}^{m-1} \mathbb{E}_{h_i \sim P_i} \left[\exp \left(\lambda \tilde{\ell}_i(h_i, z_i) \right) \right] \mathbb{E}_{h_m \sim P_m} \left[\mathbb{E}[\exp(\lambda \tilde{\ell}_i(h_m, z_m)) \mid \mathcal{F}_{m-1}] \right] \right]$$

The final remark is to notice that for any $h_m \in \mathcal{H}$, $\mathbb{E}[\tilde{\ell}_m(h_m, z_m) \mid \mathcal{F}_{m-1}] = 0$ and $\tilde{\ell}_m(h_m, z_m) \in [-K, K]$ a.s. then one can apply the conditional Hoeffding's lemma which ensure us that for any $\lambda > 0$:

$$\mathbb{E}[\exp(\lambda\tilde{\ell}_m(h_m, z_m)) \mid \mathcal{F}_{m-1}] \leq \exp\left(\frac{\lambda^2 K^2}{2}\right).$$

One then has $\xi_m \leq \exp\left(\frac{\lambda^2 K^2}{2}\right)\xi_{m-1}$. The recursion assumption concludes the proof.

□

Proof of Lemma D.3. Let A be a $\mathcal{F}$-measurable event. One wants to show that

$$\mathbb{E}\left[\mathbb{E}_{h\sim P}[f(h,z)]\mathbb{1}_A\right] = \mathbb{E}\left[\mathbb{E}_{h\sim P}\left[\mathbb{E}[f(h,z) \mid \mathcal{F}]\right]\mathbb{1}_A\right]$$

Where the first expectation in each term is taken over z. This will be enough to conclude that

$$\mathbb{E}\left[\mathbb{E}_{h\sim P}[f(h,z)] \mid \mathcal{F}\right] = \mathbb{E}_{h\sim P}\left[\mathbb{E}[f(h,z) \mid \mathcal{F}]\right]$$

thanks to the definition of conditional expectation. We first start by using the fact that P is $\mathcal{F}$-measurable and that $P >> P_0$ with P_0 a constant measure. This is enough to obtain that the Radon-Nykodym derivative $\frac{dP}{dP_0}$ is a $\mathcal{F}$-measurable function, thus:

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}_{h\sim P}[f(h,z)]\mathbb{1}_A\right] &= \mathbb{E}\left[\mathbb{E}_{h\sim P_0}\left[f(h,z)\frac{dP}{dP_0}(h)\right]\mathbb{1}_A(z)\right] \\ &= \mathbb{E}\left[\mathbb{E}_{h\sim P_0}\left[f(h,z)\frac{dP}{dP_0}(h)\mathbb{1}_A(z)\right]\right]\end{aligned}$$

Because $f(h,z)\frac{dP}{dP_0}(h)\mathbb{1}_A(z)$ is a positive function, and that P_0 is fixed, one can apply the classical Fubini-Tonelli theorem:

$$= \mathbb{E}_{h\sim P_0}\left[\mathbb{E}\left[f(h,z)\frac{dP}{dP_0}(h)\mathbb{1}_A(z)\right]\right]$$

One now conditions by $\mathcal{F}$ and use the fact that $\frac{dP}{dP_0}, \mathbb{1}_A$ are $\mathcal{F}$-measurable:

$$= \mathbb{E}_{h\sim P_0}\left[\mathbb{E}\left[\mathbb{E}\left[f(h,z) \mid \mathcal{F}\right]\frac{dP}{dP_0}(h)\mathbb{1}_A(z)\right]\right]$$

We finally re-apply Fubini-Tonelli to re-intervert the expectations:

$$\begin{aligned}&= \mathbb{E}\left[\mathbb{E}_{h\sim P_0}\left[\mathbb{E}\left[f(h,z) \mid \mathcal{F}\right]\frac{dP}{dP_0}(h)\mathbb{1}_A(z)\right]\right] \\ &= \mathbb{E}\left[\mathbb{E}_{h\sim P}\left[\mathbb{E}\left[f(h,z) \mid \mathcal{F}\right]\mathbb{1}_A(z)\right]\right]\end{aligned}$$

This finally proves the announced results, yet concludes the proof.

□

D.2 Proofs of Sec. 4

We prove here Corollary 4.1 and Corollary 4.2.

D.2.1 Proof of Corollary 4.1

We fix $\hat{Q}, P$ to be online predictive sequences (with $\hat{Q}_1, P_1$ being data-free priors). Recall that we assimilated the stochastic kernels $\hat{Q}_i, P_i$ to the their associated data-dependent sitribution given a sample S $\hat{Q}_i(S,.), P_i(S,.)$.

As in Thm. 2.3, our predictor set of interest is $\mathcal{H}_m := \mathcal{H}^{\otimes m}$ and then, our predictor h is a tuple $(h_1, .., h_m) \in \mathcal{H}$. We consider the stochastic kernel Q belonging to the class $\mathcal{C}$ defined in Eq. (10) such that for any $S \in \mathcal{S}, Q(S,.) = \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}$. Similarly one defines $Q^0 \in \mathcal{C}$ such that for any $S \in \mathcal{S}, Q^0(S,.) = P_1 \otimes ... \otimes P_m$

Proof for (Ψ_1, Φ_1): For $\lambda > 0$, we set our function f to be for any dataset S and predictor tuple $h = (h_1, ..., h_m)$,

$$f(S, h) = \lambda \left(\sum_{i=1}^{m} \mathbb{E}\left[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}\right] - \sum_{i=1}^{m} \ell(h_i, z_i) \right)$$

We then apply proposition C.1 with the function f, Q, Q^0 defined above. One then has by dividing by λ with probability $1-\delta$ over $S \sim \mu$ and $h = (h_1, ..., h_m) \sim \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}$:

$$\sum_{i=1}^{m} \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] \leq \sum_{i=1}^{m} \ell(h_i, z_i) + \frac{1}{\lambda} \log \left(\frac{dQ_S}{dQ_S^0}(h_i) \right) + \frac{1}{\lambda} \log(\xi_m) + \frac{\log(1/\delta)}{\lambda}$$

And then using the fact that $S \in \mathcal{S}, Q_S = \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}, Q_S^0 = P_1 \otimes ... \otimes P_m$ gives us:

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

with $\xi_m = \mathbb{E}_S \left[\mathbb{E}_{h_1, ..., h_m \sim Q_S} \left[\exp\left(\lambda \sum_{i=1}^{m} \tilde{\ell}_i(h_i, z_i) \right) \right] \right]$ and for any i, $\tilde{\ell}_i(h_i, z_i) = \mathbb{E}\left[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}\right] - \ell(h_i, z_i)$

Notice that, because P is an online predictive sequence, then one can apply directly Lemma D.2 to conclude that $\xi_m \leq \exp\left(\frac{\lambda^2 K^2 m}{2}\right)$.

We also use [Viallard et al., 2021, Lemma 11] which derives the calculation of the disintegrated KL divergence between two Gaussians. One then has for any i, with $h_i = \hat{w}_{i+1} + \varepsilon_i$:

$$\log \left(\frac{d\hat{Q}_{i+1}}{dP_i}(h_i) \right) = \frac{||\hat{w}_{i+1} + \varepsilon_i - w_i^0||^2 - ||\varepsilon||^2}{2\sigma^2}$$

Combining those facts altogether allows us to conclude.

Proof for (Ψ_2, Φ_2): For $\lambda > 0$, we set our function f to be for any dataset S and predictor tuple $(h = h_1, ..., h_m)$,

$$f(S, h) = \exp\left(\lambda \left(\sum_{i=1}^{m} \mathbb{E}\left[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}\right] - \sum_{i=1}^{m} \ell(h_i, z_i) \right) \right)$$

We take $\alpha = 2$ and apply this time proposition C.2. One then has by dividing by 2λ with probability $1-\delta$ over $S \sim \mu$ and $h = (h_1, ..., h_m) \sim \hat{Q}_2 \otimes ... \otimes \hat{Q}_{m+1}$:

$$\sum_{i=1}^{m} \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] \leq \sum_{i=1}^{m} \ell(h_i, z_i) + \frac{3}{2\lambda} \log \frac{2}{\delta} + \frac{D_2\left(Q_S \| Q_S^0\right)}{2\lambda} + \frac{1}{2\lambda} \log \left(\underbrace{\mathop{\mathbb{E}}_{S' \sim \mu} \mathop{\mathbb{E}}_{h' \sim Q_{S'}^0} f\left(S', h'\right)^2}_{:= \xi'_m} \right)$$

We first notice that $D_2\left(Q_S \| Q_S^0\right) = \sum_{i=1}^{m} D_2(\hat{Q}_{i+1} \| P_i)$ as our predictors are drawn independently once S is given.

We also use that for any i, the Rényi divergence with $\alpha = 2$ between $\hat{Q}_{i+1}$ and P_i (two multivariate Gaussians with same covariance matrix) is $\frac{\|\hat{w}_{i+1} - w_i^0\|^2}{\sigma^2}$ (as recalled in Gil et al. [2013]).

We then remark that:

$$\xi'_m = \mathop{\mathbb{E}}_{S'\sim\mu h'\sim Q^0_{S'}} \exp\left(2\lambda\left(\sum_{i=1}^m \mathbb{E}\left[\ell(h'_i, z'_i) \mid \mathcal{F}_{i-1}\right] - \sum_{i=1}^m \ell(h'_i, z'_i)\right)\right).$$

Thus we recover the exponential moment ξ_m from the Rivasplata's case up to a factor 2 within the exponential. We then apply Lemma D.2 with $\lambda' = 2\lambda$ to obtain that $\xi'_m \leq \exp\left(2\lambda^2 K^2 m\right)$.

Combining all those facts allows us to conclude.

D.2.2 Proof of Corollary 4.2

We apply the exact same proof than Corollary 4.1. The only difference is the way to define our stochastic kernels. We now take, for a single online predictive sequence $\hat{Q}$ the following stochastic kernels:

We consider the stochastic kernel Q belonging to the class $\mathcal{C}$ defined in Eq. (10) such that for any $S \in \mathcal{S}, Q(S, .) = \hat{Q}_1 \otimes ... \otimes \hat{Q}_m$ and we take $Q_0 = Q$.

This fact allows the divergence terms (Rényi or KL depending on which bound we consider) to vanish. The rest of the proof remains unchanged.

E Additional experiment

In this section we perform error bars for our OPBD methods in order to evaluate their volatility. We ran $n = 50$ times our algorithms and then show in the table below for each data set the means and the standard deviation of our averaged cumulative losses at regular time steps. We denote for $i \in \{1, 2\}$ 'OPBD Ψ_i' to indicate that this algorithm is our OPBD method used with thev optimisation objective Ψ_i.

Analysis Those tables shows the robustness of our OPBD methods to their intrinsic randomness: we always have a decreasing mean through time as well as an overall variance reduction. Note that for the most complicated problem (California Housing dataset), the variance is the highest. More precisely, we notice that the standard deviation of OPBD with Ψ_1 is always greater than the one of OPBD with Ψ_2 which is not a surprise as Ψ_1 involves a disintegrated KL divergence while Ψ_2 is a proper Rényi divergence. Hence the additional volatility for $OPBD with \Psi_1$.

This fact is particurlaly noticeable on the California Housing dataset where both the means and variance of OPBD with Ψ_1 increase drastically between t=16000 and t=20000 while the increase is more attenuated for OPBD with Ψ_2. This fact is also visible on Fig. 1.

	means OPBD Ψ_1	std OPBD Ψ_1	means OPBD Ψ_2	std OPBD Ψ_2
t=200	0.2014	0.0034	0.1993	0.0007
t=400	0.1888	0.0030	0.1861	0.0004
t=600	0.1867	0.0023	0.1839	0.0003
t=800	0.1714	0.0020	0.1686	0.0003
t=1000	0.1760	0.0016	0.1731	0.0003

Table 1: Error bars for the Boston Housing dataset

	means OPBD Ψ_1	std OPBD Ψ_1	means OPBD Ψ_2	std OPBD Ψ_2
t=100	0.1619	0.0063	0.1601	0.0030
t=200	0.1350	0.0057	0.1361	0.0008
t=300	0.1214	0.0044	0.1241	0.0009
t=400	0.1210	0.0043	0.1238	0.0021
t=500	0.1131	0.0037	0.1159	0.0015

Table 2: Error bars for the Breast Cancer dataset

	means OPBD Ψ_1	std OPBD Ψ_1	means OPBD Ψ_2	std OPBD Ψ_2
t=150	0.7102	0.0061	0.7069	0.0007
t=300	0.6455	0.0056	0.6422	0.0007
t=450	0.6134	0.0042	0.6103	0.0007
t=600	0.5860	0.0035	0.5837	0.0008
t=750	0.5685	0.0031	0.5664	0.0008

Table 3: Error bars for the PIMA Indians dataset

	means OPBD Ψ_1	std OPBD Ψ_1	means OPBD Ψ_2	std OPBD Ψ_2
t=4000	0.9320	0.0572	0.8905	0.0003
t=8000	0.6325	0.0335	0.5947	0.0003
t=12000	0.5314	0.0254	0.4954	0.0002
t=16000	0.4967	0.0299	0.4477	0.0004
t=20000	0.5273	0.1056	0.4355	0.0030

Table 4: Error bars for the California Housing dataset