Online PAC-Bayes learning: theory and algorithms for non-convex objectives.

Maxime Haddouche

Inria

https://maximehaddouche.github.io/

Presented work

Online PAC-Bayes Learning

Maxime Haddouche
Inria and University College London
France and UK

Benjamin Guedj
Inria and University College London
France and UK

https://arxiv.org/pdf/2206.00024.pdf

Summary

About Online Learning

- Online learning is about to find a way for our algorithm to learn while dealing with an extremely huge amount of data.

About Online Learning

- Online learning is about to find a way for our algorithm to learn while dealing with an extremely huge amount of data.
- When we can not manage the whole dataset at once: treat data sequentially.
  The goal is then to learn simultaneously than this data arrival.

About Online Learning

- Online learning is about to find a way for our algorithm to learn while dealing with an extremely huge amount of data.
- When we can not manage the whole dataset at once: treat data sequentially.
  The goal is then to learn simultaneously than this data arrival.
- That is online learning (OL)!

Online Learning vs Batch Learning

- OL differs from batch learning
- Batch learning widely used in ML: you make your algorithm learn over the full dataset (seen as a batch) over several epochs.

Online Learning vs Batch Learning

- OL differs from batch learning
- Batch learning widely used in ML: you make your algorithm learn over the full dataset (seen as a batch) over several epochs.
- Problem: if too many data available our algorithm cannot learn efficiently on reasonable time!

Online Learning vs Batch Learning

- OL differs from batch learning
- Batch learning widely used in ML: you make your algorithm learn over the full dataset (seen as a batch) over several epochs.
- Problem: if too many data available our algorithm cannot learn efficiently on reasonable time!
- Problem 2: If our learning goal moves through time: all our training is useless!

A classical framework in OL

- A predictor space $\mathcal{H}$.
- An environment $\mathcal{Z}$. At time i, the environment sends a data z_i drawn under un unknown distribution μ_i.

A classical framework in OL

- A predictor space $\mathcal{H}$.
- An environment $\mathcal{Z}$. At time i, the environment sends a data z_i drawn under un unknown distribution μ_i.
- A sequence of loss functions $(\ell_i)_{i \geqslant 1}$. At time i, one wants to produce a good predictor $h_{i+1} \in \mathcal{H}$ s.t. $\ell_{i+1}(h_{i+1})$ is small.

A classical framework in OL

- A predictor space $\mathcal{H}$.
- An environment $\mathcal{Z}$. At time i, the environment sends a data z_i drawn under un unknown distribution μ_i.
- A sequence of loss functions $(\ell_i)_{i \geqslant 1}$. At time i, one wants to produce a good predictor $h_{i+1} \in \mathcal{H}$ s.t. $\ell_{i+1}(h_{i+1})$ is small.

Question: how do we produce such good predictors?

A celebrated algorithm

Online Gradient Descent

Onto a closed convex $\mathcal{K}$, OGD produces predictors from an initial h_1 as follows:

$$\forall i \geqslant 1, h_{i+1} = \Pi_{\mathcal{K}}\left(h_i - \nabla \ell_i(h_i)\right)$$

A celebrated algorithm

Online Gradient Descent

Onto a closed convex $\mathcal{K}$, OGD produces predictors from an initial h_1 as follows:

$$\forall i \geqslant 1, h_{i+1} = \Pi_{\mathcal{K}}\left(h_i - \nabla \ell_i(h_i)\right)$$

Question: how to measure its efficiency?

Regrets

Definition

The *static regret* of a decision sequence $(h_t)_{t \geqslant 0}$ at time T as:

$$Regret_T := \sum_{t=1}^{T} \ell_t(h_t) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell_t(h)$$

The *dynamic regret* is defined as:

$$Dyn - Regret_T := \sum_{t=1}^{T} \ell_t(h_t) - \sum_{t=1}^{T} \inf_{h \in \mathcal{H}} \ell_t(h)$$

(Our) regrets

- The regret compares the quality of our predictions wrt the best strategy.
- Interest of this approach: allows us to exploit tools from convex optimisation
- Can we work beyond convex losses?

(Our) regrets

- The regret compares the quality of our predictions wrt the best strategy.
- Interest of this approach: allows us to exploit tools from convex optimisation
- Can we work beyond convex losses? **Yes, thanks to PAC-Bayesian theory.**

What is PAC-Bayes learning?

- A branch of learning theory
- Emerged in the late 90s with the works of Shawe-Taylor and Williamson, 1997 and McAllester, 1998, 1999.
- Technical tools: measure theory, concentration inequalities, information theory. Also Catoni, 2007 used tools from statistical physics

What is PAC-Bayes learning?

- A branch of learning theory
- Emerged in the late 90s with the works of Shawe-Taylor and Williamson, 1997 and McAllester, 1998, 1999.
- Technical tools: measure theory, concentration inequalities, information theory. Also Catoni, 2007 used tools from statistical physics

For more precision see the recent surveys of:

1. Alquier 2021: `https://arxiv.org/abs/2110.11216`
2. Guedj 2019: `https://arxiv.org/abs/1901.05353`

Terminology

The two terms 'PAC' and 'Bayes' stand for the following.

- PAC is the acronym of 'Probably Approximately Correct'.
- 'Bayes' says that we take inspiration from the Bayesian philopsophy.

Terminology

The two terms 'PAC' and 'Bayes' stand for the following.

- PAC is the acronym of 'Probably Approximately Correct'.
- 'Bayes' says that we take inspiration from the Bayesian philopsophy. Indeed, PAC-Bayesian theory aims to construct distributions over the predictor space instead of a single point. It also exploits the idea of building a posterior distribution from a prior one (without using Bayes formula).

An usual framework

A *learning problem* is specified by tuple $(\mathcal{H}, \mathcal{Z}, \ell)$ where:

- $\mathcal{H}$ is the space of considered predictors
- $\mathcal{Z}$ is the 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.

An usual framework (2)

- $S = (z_1, ...z_m)$ 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 does PAC-Bayes do?

PAC-Bayes theory aims to design a meaningful distribution Q over $\mathcal{H}$.
A classical PAC-Bayes bound controls the *expected generalisation error*:

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

with regards to the *expected empirical error*:

$$\mathbb{E}_{h\sim Q}[R_m(h)] := \mathbb{E}_{h\sim Q}\left[\frac{1}{m}\sum_{i=1}^{m}\ell(h, z_i)\right]$$

McAllester's bound

Assumptions: $\ell \in [0, 1]$, iid data, data-free prior

Theorem

For any prior distribution P, we have with probability $1 - \delta$ over the m-sample S, for any posterior distribution Q such that $Q \ll P$:

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

where KL is the Kullback-Leibler divergence.

Our framework

We stay close from the PAC-Bayes learning framework:

- A data space $\mathcal{Z}$, a predictor space $\mathcal{H}$.

Our framework

We stay close from the PAC-Bayes learning framework:

- A data space $\mathcal{Z}$, a predictor space $\mathcal{H}$.
- A sample $S = (z_1, ..., z_m)$. **No assumptions about the data distribution.**
- $(\mathcal{F}_i)_{i \geqslant 0}$ is an adapted filtration to S.

Our framework

We stay close from the PAC-Bayes learning framework:

- A data space $\mathcal{Z}$, a predictor space $\mathcal{H}$.
- A sample $S = (z_1, ..., z_m)$. **No assumptions about the data distribution.**
- $(\mathcal{F}_i)_{i \geqslant 0}$ is an adapted filtration to S.

Our framework

We stay close from the PAC-Bayes learning framework:

- A data space $\mathcal{Z}$, a predictor space $\mathcal{H}$.
- A sample $S = (z_1, ..., z_m)$. **No assumptions about the data distribution.**
- $(\mathcal{F}_i)_{i \geqslant 0}$ is an adapted filtration to S.
- A loss $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$. **$\ell$ is bounded by $K > 0$.**
  Analogy with OL: $\ell(., z_i) \to \ell_i(.)$.

Our framework (2)

- A sequence $(P_i)_{i \geqslant 1}$ of priors verifying:

Our framework (2)

- A sequence $(P_i)_{i \geqslant 1}$ of priors verifying:

Definition

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

Our priors can depend on the past!

Our main theorem

Theorem

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

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

Analysis

- The controlled term is hybrid between OL and PAC-Bayes.

Analysis

- The controlled term is hybrid between OL and PAC-Bayes.
- The sum: close to the OL philosophy to take into account moving objectives.

Analysis

- The controlled term is hybrid between OL and PAC-Bayes.
- The sum: close to the OL philosophy to take into account moving objectives.
- The conditional expectation: a dynamic generalisation error from PAC-Bayes world: at each time step, how good are we on average?

Analysis

- The controlled term is hybrid between OL and PAC-Bayes.
- The sum: close to the OL philosophy to take into account moving objectives.
- The conditional expectation: a dynamic generalisation error from PAC-Bayes world: at each time step, how good are we on average?
- A sum of clasical PAC-Bayesian quantities appears on the right hand side $\Longrightarrow$ towards an optimisation procedure?

Sketch of the proof: background

Main technical tool: sotchastic kernels of Rivasplata et al. 2020.

Definition (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 $\mathtt{Stoch}(\mathcal{S}, \mathcal{H})$ 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.

Sketch of the proof: background (2)

Theorem

Let $\mu \in \mathcal{M}_1(\mathcal{S})$*,* $Q^0 \in \mathtt{Stoch}(\mathcal{S}, \mathcal{F})$*. 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* $Q \in \mathtt{Stoch}(\mathcal{S}, \mathcal{F})$ *and any* $\delta \in (0, 1)$*, with probability at least* $1 - \delta$ *over the random draw of* $S \sim \mu$ *we have*

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

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

Sketch of the proof: Framework

Main idea: exploit the last theorem by taking for predictor space $\mathcal{H}_m := \mathcal{H}^{\otimes m}$ instead of $\mathcal{H}$.

Sketch of the proof: Framework

Main idea: exploit the last theorem by taking for predictor space $\mathcal{H}_m := \mathcal{H}^{\otimes m}$ instead of $\mathcal{H}$.
Thus, 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:

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

Sketch of the proof: framework

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

and $F(x, y) = \lambda(x - y)$

Sketch of the proof: framework

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

and $F(x,y) = \lambda(x-y)$
for $P = (P_1, ...P_m)$ our online predictive sequence, $Q^0 \in \mathcal{C}$ s.t. $Q_S^0 = P_1(S) \otimes ... \otimes P_m(S)$.
We define Q_S similarly for the posteriors $Q_1, ..., Q_m$ ($Q_S = Q_1(S) \otimes ... \otimes Q_m(S)$) .

Sketch of the proof

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

and applying Rivasplata et al. bound gives:

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

$$\leqslant \sum_{i=1}^{m} \mathbb{E}_{h_i \sim Q_i}[\ell(h_i, z_i)] + \frac{KL(Q_S \| Q_S^0)}{\lambda} + \frac{\log(\xi_m/\delta)}{\lambda}$$

Sketch of the proof

And $KL(Q_S \| Q_S^0) = \sum_{i=1}^m KL(Q_i \| P_i)$ thanks to the definition of our kernels. Then the last term to control is:

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

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

Sketch of the proof

Lemma

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

Hence the final result!

Online PAC-Bayesian (OPB) training bound

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 $S \sim \mu$:

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

Optimisation procedure

For a data stream $S = \{z_1, ..., z_m\}$, a fixed a scale parameter $\lambda > 0$ and an online predictive sequence P_i:

$$\hat{Q}_1 = P, \quad \forall i \geqslant 1 \ \hat{Q}_{i+1} = \mathrm{argmin}_Q \, \mathbb{E}_{h_i \sim Q} \left[\ell(h_i, z_i) \right] + \frac{\mathrm{KL}(Q \| P_i)}{\lambda} \tag{2}$$

which leads to the explicit formulation

$$\frac{d\hat{Q}_{i+1}}{dP_i}(h) = \frac{\exp\left(-\lambda \ell(h, z_i)\right)}{\mathbb{E}_{h \sim P_i}\left[\exp\left(-\lambda \ell(h, z_i)\right)\right]}. \tag{3}$$

Optimisation procedure

For a data stream $S = \{z_1, ..., z_m\}$, a fixed a scale parameter $\lambda > 0$ and an online predictive sequence P_i:

$$\hat{Q}_1 = P, \quad \forall i \geqslant 1 \ \hat{Q}_{i+1} = \operatorname{argmin}_Q \mathbb{E}_{h_i \sim Q} \left[\ell(h_i, z_i)\right] + \frac{\mathrm{KL}(Q \| P_i)}{\lambda} \tag{2}$$

which leads to the explicit formulation

$$\frac{d\hat{Q}_{i+1}}{dP_i}(h) = \frac{\exp\left(-\lambda \ell(h, z_i)\right)}{\mathbb{E}_{h \sim P_i}\left[\exp\left(-\lambda \ell(h, z_i)\right)\right]}. \tag{3}$$

Thus, the formulation of Eq. 3, which has been highlighted by Catoni shows that our online procedure produces Gibbs posteriors.

Analysis

- In the training bound: impacting right hand-side as it provides our OPB algorithm.
- Left hand side: expresses how the posterior $\hat{Q}_{i+1}$ generalises well on average to any new draw of z_i.

Analysis

- In the training bound: impacting right hand-side as it provides our OPB algorithm.
- Left hand side: expresses how the posterior $\hat{Q}_{i+1}$ 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 procedure is robust to the randomness of S, hence the interest of optimising the right hand side of the bound.

An OPB test bound

Our training bound 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. This is the goal of our next theorem.

Corollary (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* $S \sim \mu$*:*

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

Analysis

- This leads to the (empirical) optimal rate of $\sum_{i=1}^{m} \mathbb{E}_{h_i \sim \hat{Q}_i}\left[\ell(h_i, z_i)\right] + O(\sqrt{m \log(1/\delta)})$.
- NB: if we want a guarantee valid for any time T of our procedure $\rightarrow$ union bound $\rightarrow$ $O(\sqrt{m \log(m/\delta)})$.

Analysis

- This leads to the (empirical) optimal rate of $\sum_{i=1}^{m} \mathbb{E}_{h_i \sim \hat{Q}_i} [\ell(h_i, z_i)] + O(\sqrt{m \log(1/\delta)})$.
- NB: if we want a guarantee valid for any time T of our procedure $\rightarrow$ union bound $\rightarrow O(\sqrt{m \log(m/\delta)})$.
- the cost of a more precise control of the behavior of the OPB algorithm at each time step is $\sqrt{\log(m)}$

An issue with the OPB algroithm.

A legitimate criticism to OPB learning: Gibbs posterior can be costful to implement given the need to estimate an expenential moment at each time step.

Can we overcome this difficulty

An issue with the OPB algroithm.

A legitimate criticism to OPB learning: Gibbs posterior can be costful to implement given the need to estimate an expenential moment at each time step.

Can we overcome this difficulty

The answer is Yes! Thans to disintegrated PAC-Bayes bounds.

A general Online PAC-Bayes Disintegrated (OPBD) training bound

A general shape for OPBD training bounds

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

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

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.

A general OPBD algorithm for Gaussian measures

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

A general OPBD algorithm for Gaussian measures

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

The idea of using Gaussian measures comes from Viallard, 2021.
The reason: 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 totally defines h.

Two concrete pairs (Ψ, Φ).

Corollary

For any $\hat{Q}_i = \mathcal{N}(\hat{w}_i, \sigma^2 \mathbf{I}_d)$ 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:

$$\Psi_1(h_i, \hat{w}_{i+1}, w_i^0) = \frac{1}{\lambda}\left(\frac{\|\hat{w}_{i+1} + \varepsilon_i - w_i^0\|^2 - \|\varepsilon\|^2}{2\sigma^2}\right)$$

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

$$\Psi_2(h_i, \hat{w}_{i+1}, w_i^0)) = \frac{1}{\lambda}\frac{\|\hat{w}_{i+1} - w_i^0\|^2}{2\sigma^2} \quad \Phi_2(m) = \lambda m K^2 + \frac{3\log(1/\delta)}{2\lambda}.$$

OPBD test bounds

General shape

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

$$\sum_{i=1}^{m} \mathbb{E}[\ell(h_i, z_i) \mid \mathcal{F}_{i-1}] \leqslant \sum_{i=1}^{m} \ell(h_i, z_i) + \Phi(m), \tag{5}$$

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

Two concrete OPBD test bounds

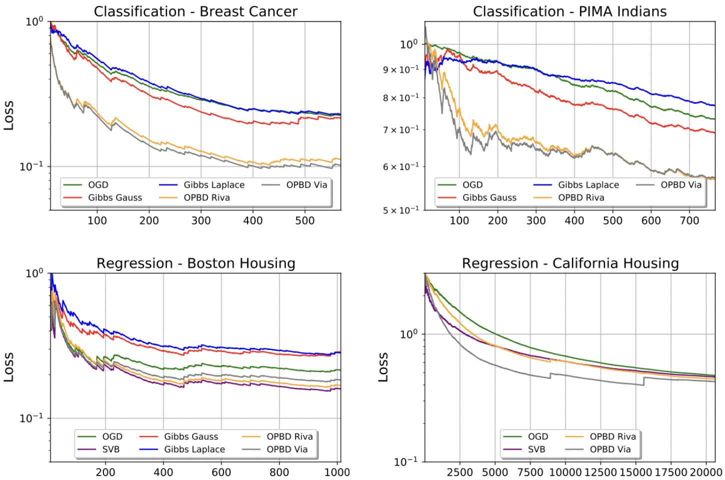
Corollary

For 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. 5 holds with :

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

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

Experiments



What this talk could have also been about

1. PAC-Bayes beyond bounded losses (`https://www.mdpi.com/1099-4300/23/10/1330`)
2. PAC-Bayes for kernel PCA (https://arxiv.org/abs/2012.10369, to be updated)
3. Optimistic adaptation of classical online algorithms (online soon!)

Thank you for listening !