Learning via Surrogate PAC-Bayes

Antoine Picard-Weibel, Roman Moscoviz, Benjamin Guedj

Learning via Surrogate PAC-Bayes

Antoine Picard-Weibel
MODAL
Inria
59650, Villeneuve d'Ascq, France
antoine.picard.ext@suez.com

Roman Moscoviz
CIRSEE
SUEZ
78230 Le Pecq, France
roman.moscoviz@suez.com

Benjamin Guedj
MODAL
Inria
59650, Villeneuve d'Ascq, France
benjamin.guedj@inria.fr

October 11, 2024

Abstract

PAC-Bayes learning is a comprehensive setting for (i) studying the generalisation ability of learning algorithms and (ii) deriving new learning algorithms by optimising a generalisation bound. However, optimising generalisation bounds might not always be viable for tractable or computational reasons, or both. For example, iteratively querying the empirical risk might prove computationally expensive. In response, we introduce a novel principled strategy for building an iterative learning algorithm via the optimisation of a sequence of surrogate training objectives, inherited from PAC-Bayes generalisation bounds. The key argument is to replace the empirical risk (seen as a function of hypotheses) in the generalisation bound by its projection onto a constructible low dimensional functional space: these projections can be queried much more efficiently than the initial risk. On top of providing that generic recipe for learning via surrogate PAC-Bayes bounds, we (i) contribute theoretical results establishing that iteratively optimising our surrogates implies the optimisation of the original

generalisation bounds, (ii) instantiate this strategy to the framework of meta-learning, introducing a meta-objective offering a closed form expression for meta-gradient, (iii) illustrate our approach with numerical experiments inspired by an industrial biochemical problem.

1 Introduction

Generalisation is arguably one of the central problems in machine learning. Among the different techniques to study generalisation, PAC-Bayes has gained considerable traction over the past decade, as evidenced by the surge in publications. We refer to the seminal works of Shawe-Taylor and Williamson [1997], McAllester [1999], Catoni [2004, 2007] and to the recent surveys and monographs from Guedj [2019], Hellström et al. [2023], Alquier [2024] for a thorough overview of the field.

One appealing feature is that PAC-Bayes learning is a comprehensive setting for (i) studying the generalisation ability of learning algorithms and (ii) deriving new learning algorithms by optimising a PAC-Bayes generalisation bound. This is the strategy pursued in a number of recent works, among which Germain et al. [2009], Biggs and Guedj [2021], Germain et al. [2015], Viallard et al. [2023], Zantedeschi et al. [2021], Rivasplata et al. [2019], Pérez-Ortiz et al. [2021], Zhou et al. [2019].

We now regard this strategy of substituting a generalisation bound to more classical training objectives as established, and we focus here on the computational aspect of this strategy. Indeed, optimising generalisation bounds might not always be viable for tractable or computational reasons, or both. Most PAC-Bayes bounds do not admit a close form minima formulation; moreover, such bounds involve expectations and divergence terms which in general settings can not be evaluated in closed form and thus require the use of approximation methods such as Monte-Carlo sampling (see amongst others Seldin and Tishby [2010], Dziugaite and Roy [2017], Neyshabur et al. [2017], Mhammedi et al. [2019]). Such approximation methods can prove computationally intensive, notably if the empirical risk, whose expectation is optimised in the bound, is hard to query. Picard-Weibel et al. [2024] reports that such queries proved to be the main computational bottleneck when optimising a PAC-Bayes bound in a bio-chemical model calibration task. More generally, models whose predictions require solving stiff ordinary differential equations (ODE) or partial differential equations (PDE), such as naturally occurs in physics or biology inspired problems, result in empirical risks whose query can be computationally expensive, in practice all but making numerous iterative computations of PAC-Bayes objective's gradients impracticable.

In response to the aforementioned difficulties for optimising PAC-Bayes generalisation bounds in practice, we introduce a novel principled strategy designed to mitigate the computational cost of querying the empirical risk. We build a learning algorithm which iteratively optimises a sequence of surrogate training objectives in which the empirical risk is replaced by a proxy. This proxy is built as the orthogonal projection of the true empirical risk on a functional

vector space of finite dimension, which we conjecture can be queried much more efficiently than the initial risk. A key motivation is that such surrogate objectives can offer adequate approximations of the true objective valid much further away than the linear approximation offered by the gradient, and enable larger optimisation steps. This effectively decouples the complexity of querying the empirical risk and optimising PAC-Bayes objectives.

Our contributions. Our four main contributions span theory, algorithmic, application to meta-learning and numerical experiments.

1. We provide a generic recipe for learning via surrogate PAC-Bayes bounds, which we believe is of practical interest for machine learning tasks involving computationally intensive models with moderate dimension (e.g. physics models with less than few hundred parameters),

2. contribute theoretical results establishing that iteratively optimising our surrogates implies the optimisation of the original generalisation bounds. This is established by theorem 1 and further developed in corollary 1 and theorem 2,

3. instantiate this strategy to the framework of meta-learning, introducing a meta-objective with a closed form expression for meta-gradient,

4. illustrate our approach with numerical experiments inspired by an industrial biochemical setting using an anaerobic digestion model.

Outline. The paper is organised as follows: in section 2 we set the stage and introduce our generic framework. In section 3, we construct functional approximation spaces and establish generic guarantees for our framework. In section 4, we focus on Catoni's bound [Catoni, 2007] and describe a practical implementation of our framework. In section 5, we investigate how our surrogate PAC-Bayes minimisation strategy can be used in meta-learning settings. Numerical experiments are described in section 6. Future prospects are discussed in section 7. The manuscript closes with an appendix in which we gather (i) technical proofs in appendix A, (ii) implementation details in appendix B.

2 A generic surrogate framework

Consider a measurable space $\mathcal{H}$ of predictors, denote $\mathcal{P}$ the set of all probability distributions on $\mathcal{H}$, and $\mathcal{M}(\mathcal{H})$ the set of measurable real valued functions. For a probability distribution $\pi \in \mathcal{P}$, let $\mathrm{L}^1(\pi)$ (resp. $\mathrm{L}^2(\pi)$) denote the set of integrable (resp. square integrable) functions with respect to π. For a $f \in \mathrm{L}^1(\pi)$, $\pi[f]$ denotes the mean of f with respect to π (the notation is extended for functions outputting vectors), while for functions in $\mathrm{L}^2(\pi)$, $\mathbb{V}_\pi[f]$ denotes the variance of f (resp. covariance).

A PAC-Bayes bound, denoted PB, can generically be summarised as a real valued function of four variables: a generic distribution $\pi \in \mathcal{P}$, a prior distribution $\pi_{\mathrm{p}} \in \mathcal{P}$, an empirical risk function $R \in \mathcal{P}$, and other factors which we regroup as γ (e.g. the confidence level, the PAC-Bayes temperature, the size of the dataset). A PAC-Bayes theorem states that, under given assumptions on the data generation mechanisms and risk, the average risk function $\overline{R} = \mathbb{E}[R]$ satisfies for some function q

$$\mathbb{P}\left[\forall \pi \in \mathcal{P},\ \pi\left[\overline{R}\right] \leq \mathrm{PB}\left(\pi, R, \pi_{\mathrm{p}}, \gamma\right)\right] \geq 1 - q(\gamma), \tag{1}$$

where the probability is taken on the data generation mechanism. Due to the bound holding simultaneously for all distributions with high probability, it notably holds with high probability on the minimiser of the bound, hence the PAC-Bayes minimisation task

$$\arg\inf_{\pi \in \mathcal{P}} \mathrm{PB}\left(\pi, R, \pi_{\mathrm{p}}, \gamma\right). \tag{2}$$

We consider a restriction of this minimisation task on a subset $\Pi \subset \mathcal{P}$ of all probability distributions. Such a restriction might be justified by various considerations, including storage of the calibrated distribution, simplification of the minimisation task or even expert knowledge [Alquier et al., 2016, Dziugaite and Roy, 2017, Picard-Weibel et al., 2024]. However, even this simplified minimisation problem might prove computationally difficult for Gradient Descent (GD) based algorithm. This is especially the case when evaluating the empirical risk is costly, e.g. when the prediction model involves solving stiff ODEs or PDEs. As PAC-Bayes bounds depend on the π-mean of the empirical risk, each gradient estimation rely on numerous new evaluations of the empirical risk. For ODEs $\dot{X} = F(X, t, \theta)$ where F is very sensitive with respect to X, numerous evaluations of F are required to obtain adequate numerical solutions. These evaluations must moreover be performed iteratively, and hence can not be parallelized. Moreover, implementing the ODE solver in a way to benefit from GPU speeds up might not be practicable, since most ODE solver use a varying step size which will depend on θ. This will result in typically long model calls which can not be massively parallelised. To overcome this difficulty, we introduce the Surrogate PAC-Bayes bound learning framework (SuPAC), which is based on alternatively building and solving surrogate problems.

Formally, we consider an approximation algorithm $\mathrm{F} : \Pi \times \mathcal{M}(\mathcal{H}) \mapsto \mathcal{M}(\mathcal{H})$ in conjunction with an approximate solving algorithm $\mathrm{Solve} : \mathcal{P} \times \Pi \times \mathcal{M}(\mathcal{H}) \mapsto \Pi$. Informally, F constructs a proxy of the empirical risk valid for the current posterior estimation π; while Solve updates the posterior estimation by solving the resulting surrogate objective.

Algorithm 1 offers a lot of leeway for building surrogates (e.g., iteratively refining an ODE/PDE solver, tailored made surrogates for physical models, polynomial approximations) as well as solving the surrogate problem. For such a framework to be practicable, two conditions should apply: the construction of the surrogate and approximate solving should be faster than solving the initial problem, and the algorithm's result should tend to diminish the PAC-Bayes

bound. Intuitively, the choice of the approximation mechanisms plays a critical role; indeed, the more precise the approximation, the more likely is the minima of the surrogate task to be close to the true minimiser, but the harder the approximation task and the surrogate construction task.

3 Constructing surrogate function spaces

Algorithm 1 Surrogate PAC-Bayes Learning framework (SuPAC)

Require: PB, $\pi_0 \in \Pi$, $\pi_{\mathrm{p}} \in \mathcal{P}$, $R \in \mathcal{M}(\mathcal{H})$
  $\pi \leftarrow \pi_0$
  while not converged **do**
    $f \leftarrow \mathrm{F}(\pi, R)$
    $\pi \leftarrow \mathrm{Solve}(\pi_{\mathrm{p}}, \pi, f)$
  end while

A core contribution of the present work is to show that for generic PAC-Bayes bounds and generic probability families Π of dimension d, $\mathrm{L}^2(\pi)$ orthogonal projection of the true score on a functional vector space of dimension $d+1$ is sufficient to obtain convergence guarantees.

A few assumptions on the PAC-Bayes bounds, the risk R and the probability family Π are required.

Assumptions. *(A_1)* $\Pi = \{\pi_\theta, \theta \in \Theta\}$ *is a parametric set indexed by an open subset* $\Theta \subseteq \mathbb{R}^d$*;*

(A_2) $\forall \theta \in \Theta$, π_θ *is absolutely continuous with respect to* π_{p} *and* $\frac{d\pi_\theta}{d\pi_{\mathrm{p}}}(x) = \exp(\ell(\theta, x))$ *with* $\theta \mapsto \ell(\theta, x)$ *differentiable for all* x*;*

(A_3) $\forall \theta \in \Theta$, $\exists N_\theta$ *a neighbourhood of* θ *such that* $x \mapsto \sup_{\theta \in N_\theta} |\partial_\theta \ell(\theta, x)| \in L^2(\pi_\theta)$*;*

(A_4) $R \in \cap_{\theta \in \Theta} L^2(\pi_\theta)$*;*

(A_5) *There exists* $\widetilde{\mathrm{PB}}$ *such that* $\mathrm{PB}(\pi_\theta, R, \pi_{\mathrm{p}}, \gamma) = \widetilde{\mathrm{PB}}(\theta, \pi_\theta[R], \pi_{\mathrm{p}}, \gamma)$ *(*i.e. *the PAC-Bayes bound dependence on the empirical risk is limited to the posterior average of the empirical risk). Moreover,* $\widetilde{\mathrm{PB}}$ *is differentiable with respect to its two first arguments.*

We emphasise that these assumptions are valid for practically all PAC-Bayes bounds, most risks, and for a wide variety of probability families, and are thus rather more technical than restrictive. Although the second assumption rules out probability distributions whose support is not included in the prior support, we remark that such distributions usually obtain unbounded PAC-Bayes bounds due

to penalisation terms, and as such are already ruled out. Most standard family of distributions, including Gaussian and Gaussian mixtures, satisfy (A_1) to (A_3) for adequate parameterizations. The fourth assumption is automatically satisfied for all bounded risks, which is a typical assumption of PAC-Bayes bounds, but also allows for unbounded risks provided that they are square integrable (e.g. polynomials if Π span Gaussian would satisfy (A_4)). The last assumption is satisfied by most PAC-Bayes bound, e.g. McAllester [1999], Maurer [2004].

Since Π is parameterised by Θ, we will abuse notations for functions of Π and write $G(\theta) := G(\pi_\theta)$. For a given θ, the functional vector space $\mathcal{F}_\theta := \left\{ f_{\eta,C} : x \mapsto \eta \cdot \partial_\theta \ell(\theta, x) + C \mid \eta \in \mathbb{R}^d, C \in \mathbb{R} \right\}$ provides a natural approximation space of dimension $d+1$. We are now in a position to state our main approximation result

Theorem 1. *Under assumptions (A_1) to (A_5), replacing the empirical risk R by the proxy risk*

$$f^{R,\theta} := \arg \inf_{f \in \mathcal{F}_\theta} \pi_\theta[(R-f)^2]$$

leaves the gradient of the objective PB *invariant, i. e.*

$$\partial_1 \mathrm{PB}(\theta, R, \pi_\mathrm{p}, \gamma) = \partial_1 \mathrm{PB}(\theta, f^{R,\theta}, \pi_\mathrm{p}, \gamma).$$

This result also holds if the approximation space $\mathcal{F}_\theta$ is replaced by $\{f + g \mid f \in \mathcal{F}_\theta, \mathcal{G}\}$ for any set $\mathcal{G} \subset L^2(\pi_\theta)$.

Proof. Assumptions *(A_3)* and *(A_4)* allow differentiating $\theta \mapsto \pi_\theta[R] = \pi\left[\frac{\mathrm{d}\pi_\theta}{\mathrm{d}\pi} R\right]$ under the integral sign (see Theorem 6.28 in Klenke [2020]), yielding $\nabla \pi_\theta[R] = \pi_\theta[R \partial_\theta \ell]$. As such, the derivative of $\widetilde{\mathrm{PB}}(\theta, \pi_\theta[R], \pi_\mathrm{p}, \gamma)$ with respect to θ equals $\partial_1 \widetilde{\mathrm{PB}}(\theta, \pi_\theta[R], \pi_\mathrm{p}, \gamma) + \partial_2 \widetilde{\mathrm{PB}}(\theta, \pi_\theta[R], \pi_\mathrm{p}, \gamma) \pi_\theta[R \partial_\theta \ell]$.

As the only dependence on the gradient with respect to R is on the value of $\pi[R]$ at which the derivative is evaluated and on the vector $\pi_\theta[R\partial_\theta \ell]$, it follows that $\partial_\theta \mathrm{PB}$ is not modified by replacing R by a function $f \in L^2(\pi_\theta)$ satisfying the following linear system:

$$\begin{cases} \pi_\theta[R \partial_\theta \ell] &= \pi_\theta[f \partial_\theta \ell], \\ \pi_\theta[R] &= \pi_\theta[f]. \end{cases} \tag{3}$$

By construction of $\mathcal{F}_\theta$, the linear system (3) is satisfied if and only if $(f - R) \in \mathcal{F}_\theta^\perp$, where $A^\perp$ denotes the orthogonal complement of A in $\mathrm{L}^2(\pi_\theta)$. Hence for any set $\mathcal{G} \subset \mathrm{L}^2(\pi_\theta)$, the orthogonal projection of R on $\tilde{\mathcal{F}} = \mathcal{F}_\theta + \mathcal{G}$ satisfies the linear system (3). Noticing that the orthogonal projection $f^{R,\theta}$ of R on space $\tilde{\mathcal{F}}$ satisfies $f^{R,\theta} = \arg\inf_{f \in \tilde{\mathcal{F}}} \pi_\theta[(R-f)^2]$ ends the proof. □

Informally, Theorem 1 guarantees that if searching for a PAC-Bayes posterior in a space of size d, adequately projecting the score on a space of dimension at most $d+1$ preserves the immediate surrounding of the PAC-Bayes objective. If the approximation built at θ maintains near optimal performance for a large neighbourhood of θ, this surrogate task provides a valid approximation of the

true task for a wide range of distributions, and offers approximate solutions $\tilde{\theta}$ much further away than the range of validity of the objective's gradient.

The extension of the result for $\mathcal{F}_\theta + \mathcal{G}$ implies that proxy score functions combining a known, simplified model with a learnt correction term can be used. For $\mathcal{G} = \{h\}$, it implies that the result holds if the approximation space consists of a fixed user defined proxy and a correction term. This can have direct practical implications in settings where efficient, natural proxy are available; the learnt corrective term would presumably be smaller, and hence the approximation's validity larger.

A direct consequence of Theorem 1 is a fixed point characterisation of the minima of the PAC-Bayes objective for instances of Algorithm 1 using GD based surrogate solver (see proof in appendix A.1):

Corollary 1. *Under assumptions (A_1) to (A_5), the minimiser $\hat{\theta}$ of the original PAC-Bayes bound is a fixed point of any instance of Algorithm 1 such that:*

- *the approximation function is $F(\pi_\theta, R) := \arg\inf_{f \in \mathcal{F}_\theta} \pi_\theta[(R - f)^2]$,*
- *the surrogate solving Solve strategy is any (corrected) gradient descent strategy starting at the current θ, using update steps of form $\mathrm{Updt}(\theta) = \theta - M(\pi, \theta, f, \gamma)\partial_\theta \mathrm{PB}(\theta, f, \pi_\mathrm{p}, \gamma)$, where M stands for any function returning an endomorphism, for any number of steps, any convergence criteria.*

It should be stressed that Corollary 1 does not imply that algorithm 1 improves on GD. Corollary 1 only guarantees that replacing the score by a low dimensional approximation is harmless locally. Informally, if the approximation built at θ maintains near optimal performance for a large neighbourhood of θ, this surrogate task provides a valid approximation of the objective for this wide radius, and can construct approximate solutions $\tilde{\theta}$ much further away than the range of validity of the gradient. SuPAC decouples the variations of the bound due to the evolution of θ and $f^{\theta,R}$; such a decoupling is particularly interesting if $f^{\theta,R}$ is stable.

4 Exponential family and Catoni's bound

4.1 Closed form surrogate solution and fixed point property

Theorem 1 involves approximation of the empirical risk through orthogonal projection on a local functional vector space $\mathcal{F}_\theta$ of dimension at most $d + 1$. A setting of particular interest concerns families of probabilities such that the space $\mathcal{F}_\theta$ does not depend on θ. Exponential families, i.e. family of distributions of the form

$$\Pi_T = \left\{ \pi_\theta \mid \frac{d\pi_\theta}{d\pi_\mathrm{ref}} = \exp(\theta \cdot T - g(\theta) + h) \right\},$$

are a well studied class of probability family which satisfy this property (and essentially the only such class if Θ is connected and the likelihood smooth, see theorem 3 in appendix A.3). The approximation space can be written as

$\mathcal{F} = \{f_{C,\theta} := \theta \cdot T + C\}$. Without loss of generality, we assume that functions $(1, T_1, \ldots, T_d)$ are linearly independent.

Exponential families define a tractable, yet flexible class of probability families, spanning from simple, fixed variance distributions to multimodal distributions [Cobb et al., 1983]. They englobe most familiar distribution families such as multivariate Gaussians, Beta and Gamma [Brown, 1986]. The approximation space they generate can equally vary. For Gaussian distributions, we remark that $\mathcal{F}$ covers quadratic forms.

We now focus on Catoni's PAC-Bayes bound [Catoni, 2007, Alquier, 2024],

$$\mathrm{PB}_{\mathrm{Cat}}(\pi, \pi_{\mathrm{p}}, R, (\lambda, \delta, n, C)) = \pi[R] + \lambda \mathrm{KL}(\pi, \pi_{\mathrm{p}}) + \frac{C^2}{8\lambda n} - \lambda \log(\delta),$$

where $\mathrm{KL}(\nu, \mu) = \nu\left[\frac{\mathrm{d}\nu}{\mathrm{d}\mu}\right]$ is the Kullback–Leibler divergence and λ is the PAC-Bayes temperature. Catoni's bound holds with probability $1 - \delta$ if $0 \leq R \leq C$. Due to its particular form, minimising the bound amounts to minimising the simpler objective $\pi[R] + \lambda \mathrm{KL}(\pi, \pi_{\mathrm{p}})$.

For simplicity's sake, we will assume that $\pi_{\mathrm{p}} = \pi_{\theta_p} \in \Theta$. In this setting, *($A_1$)*, *($A_2$)* and *($A_4$)* are automatically verified. A key incentive to use Catoni's bound is that the surrogate objective can be solved in closed form; for risks of form $f_{\eta,C}$, if the prior belongs to the exponential family, the minimiser of Catoni's bound on $\mathcal{P}$ belongs to Π, and it follows that

$$\arg\inf_{\theta} \mathrm{PB}_{\mathrm{Cat}}(\pi_\theta, \pi_{\theta_p}, f_{\eta,C}) = \tilde{\theta}(\eta) := \theta_p - \lambda^{-1}\eta,$$

provided that $\theta_p - \lambda^{-1}\eta \in \Theta$ (if not, Catoni's bound does not admit a minima) (see Lemma 2.2, and Corollary 2.3 in Alquier [2024]). Since the posterior distribution does not depend on the constant term C we will note f_η for any $f_{\eta,C} \in \mathcal{F}$.

We can here use the exact solution of the surrogate PAC-Bayes bound rather than have to minimise the bound through GD. The following lemma (proved in appendix A.2) bridges the gap by showing that the update rule using the closed form solution can be interpreted as a corrected GD step:

Lemma 1. *Consider an exponential family $\Pi := \{\pi_\theta \Theta \in \Theta\}$ with sufficient statistic T. Noting $\mathcal{F} := \{f_\eta : x \mapsto \eta \cdot T(x) + C \mid \eta \in \mathbb{R}^d, C \in \mathbb{R}\}$, let $f_\eta \in \mathcal{F}$. Then for any prior parameter $\theta_p \in \Theta$, for any parameter θ, the mapping $\tilde{\theta}(\eta) := \theta_p - \lambda^{-1}\eta$ satisfies:*

$$\tilde{\theta} = -\lambda^{-1} I(\theta)^{-1} \nabla_\theta \mathrm{PB}_{\mathrm{Cat}}(\theta, \theta_p, f_\eta, \gamma) + \theta,$$

where $I(\theta)$ denotes Fisher's information matrix.

A direct consequence of Lemma 1 is that Corollary 1 applies when using the exact solver for the surrogate Catoni task. Since Fisher's information is positive, it follows that the update direction $\tilde{\theta} - \theta$ always diminishes the bound locally. We summarise these results in the following theorem.

Theorem 2. *The minimiser of Catoni's PAC-Bayes objective on an exponential family is a fixed point of Algorithm 1 with approximation function*

$$F(\pi_\theta, R) := \arg \inf_{f \in \mathcal{F}} \pi_\theta[(R - f)^2],$$

and surrogate solver

$$\mathit{Solve}(\pi_\mathrm{p}, \theta, f_\eta) := \theta_p - \lambda^{-1}\eta = \arg \inf_{\theta \in \Theta} \mathrm{PB}_\mathrm{Cat}(\theta, \pi_\mathrm{p}, f_\eta, \gamma).$$

Moreover, for all θ,

$$\nabla \mathrm{PB}_\mathrm{Cat} \cdot (\mathit{Solve}(\pi_\mathrm{p}, \theta, F(\theta, R)) - \theta) \leq 0.$$

As noted above, the solution of the surrogate task must belong to Θ to define a probability distribution. There is however no guarantee that such is the case for any approximated risk. For instance, if the risk is estimated close to a local maxima by a quadratic function, the resulting surrogate task might not have a minima, and hence the resulting $\theta(\eta)$ might fail to be a probability distribution, causing the algorithm to break. Another difficulty lies in solving the approximation task. Involving an integral of a function of the risk, the objective theoretically requires evaluations of the risk at all predictors. We show in the next section how both these issues can be solved in practice.

4.2 Framework implementation: SuPAC -CE

Following theorem 2, we propose an algorithm, SuPAC -CE (code here), designed to efficiently find the minimiser of Catoni's bound on Exponential families.

4.2.1 Implementing the approximation

As the surrogate PAC-Bayes bound is solved using a closed form expression, the computational bottleneck of algorithm 1 is the approximation task of computing $\eta(\theta) = \arg\inf_{\mathbb{R}^d} \pi_\theta[(f_\eta - R - \pi_\theta[f_\eta - R])^2]$. Due to the form of f_η, this is formally a least square weighted linear approximation problem with infinite number of observations, whose solution can be explicitly written as $\eta = \mathbb{V}_\pi[T]^{-1}\pi[R(T - \pi[T])]$. This solution can be approximated using a finite number of function evaluations $R(x_i)$, replacing the probability π by an empirical counterpart $\pi_\mathrm{emp} = \sum_{i=1}^N \omega_i \delta_{x_i}$.

Different choices of (x_i, ω_i) can be considered. A first approach consists in drawing i.i.d. samples from π_θ and considering uniform weights. This guarantees that the approximated objective is unbiased. A main shortcoming of this approach, however, is that it disregards all previous risk evaluations at each step. Corrections of the form $\frac{\mathrm{d}\pi_\theta}{\mathrm{d}\pi_{\tilde{\theta}}}$ can be used to salvage samples drawn from $\pi_{\tilde{\theta}}$, all the while guaranteeing unbiased approximated objective. This however can drastically increase the variance, and thus might not be practical.

We advocate a "generation agnostic" approach for the weighing process, which treats all available risk evaluations in a like manner. We assume that

$\mathcal{H}$ is a metric space. For all predictors $(x_i)_{i\in[1,N]}$ whose risk $R(x_i)$ is known, target weights $\tilde{\omega}_i$ are defined as the probability given to the Voronoi cell $\overline{x}_i$ by distribution π_θ. This target weight can be approximated using Monte Carlo simulations and solving nearest neighbour in $(x_i)_{i\in[1,N]}$ tasks. The distance used for the Voronoi cell can depend on the distribution π_θ –(e.g. Mahalanobis distance for Gaussian exponential families). This approach requires, if the empirical distribution $\sum \omega_i \delta_{x_i}$ is to form an adequate approximation of the distribution π_θ, some queries from to π_θ. The stack of function evaluation is hence appended at each approximation step by evaluating samples from π_θ. As this weight computation can bring some overhead, it is only appropriate when risk queries are the main computational bottleneck.

4.2.2 Boundary issues

PAC-Bayes bounds typically hold for empirical risk functions satisfying moment bounds (with respect to the data generation mechanism) or boundedness conditions (the latter being usually required for Catoni's bound). Such assumptions might no longer be met for the approximated risks. A consequence is that the minimiser of the surrogate task might not exist. For instance, a local quadratic approximation of the score near a local maxima can induce a surrogate task whose minima is $-\inf$.

To ensure that for any score approximation $f_{\eta,C}$, the surrogate solver always define a probability distribution, two regularisation hyperparameters $\mathrm{kl}_{\max}$ and $\alpha_{\max}$ are introduced. $\mathrm{kl}_{\max} \in \mathbb{R}_+ \cup +\infty$ determines the maximum step size allowed between two successive posterior estimation, measured in Kullback–Leibler divergence. $\alpha_{\max} \in]0, 1]$ acts as a dampening hyperparameter. The corrected update rule is changed to $\tilde{\theta}_c(\theta) = \tilde{\alpha}(\tilde{\theta}(\eta) - \theta) + \theta$ with $\tilde{\alpha}$ the highest $\alpha \leq \alpha_{\max}$ such that $\mathrm{KL}(\tilde{\theta}_c, \theta) \leq \mathrm{kl}_{\max}$. Such $\tilde{\alpha}$ can be easily obtained through a Newton scheme or dichotomy, noticing that it is defined through $f(\tilde{\alpha}) = C$ for a non decreasing function f.

This modification does not impact the fixed point property of Theorem 2. Moreover, if the empirical risk R belongs to $\mathcal{F}$, choosing $\mathrm{kl}_{\max} < \infty$, $\alpha_{\max} = 1$ results in convergence in a finite number of steps (resp. exponential convergence for $\alpha_{\max} < 1$) (see Appendix A.4).

5 Surrogate Catoni in a Meta Learning framework

Both the Bayes and PAC-Bayes framework offer a natural connection with Meta-Learning, as both involve a natural inductive bias in the form of the prior. Previous work which studied Meta-Learning for PAC-Bayes include Pentina and Lampert [2014], Amit and Meir [2018], Rothfuss et al. [2023], Zakerinia et al. [2024]. The aim of PAC-Bayes Meta-Learning is the construction, from a sample of independent train tasks, of a prior yielding optimal generalisation bounds on new unknown test tasks. Such optimisation of the prior brings two benefits: tighter generalisation bounds (smaller penalisation); and simplified PAC-Bayes

learning task (better initial guess). For PAC-Bayes meta learning, a natural training objective can be derived from the minimised PAC-Bayes bounds obtained for each task. This defines the following meta training objective, analogue to an empirical risk at the meta level:

$$M(\pi_{\mathrm{p}}) = \sum_i \inf_{\pi \in \Pi} \mathrm{PB}(\hat{\pi}, R_i, \pi_{\mathrm{p}}, \eta_i), \tag{4}$$

where $\hat{\pi}$ denotes the task posterior. The objective defined in eq. (4) departs from previous formulations which typically involve a further penalisation term at the meta level. We advance two justifications for this simplification. First, the extra penalisation term involves divergence terms between a meta prior and meta posterior (both distributions on probability distributions) which in practice make the bound vacuous and thus of limited practical interest. Second, PAC-Bayes theory already offers guarantees on the generalisation performances of each test task, limiting the need to assess the generalisation performance at the meta level. Arguably, the task specific bound provided by using PAC-Bayes as inner algorithm is more informative than the "mean" task bound offered by a meta PAC-Bayes algorithm (when PAC-Bayes learning is used both as inner algorithm and meta training algorithm).

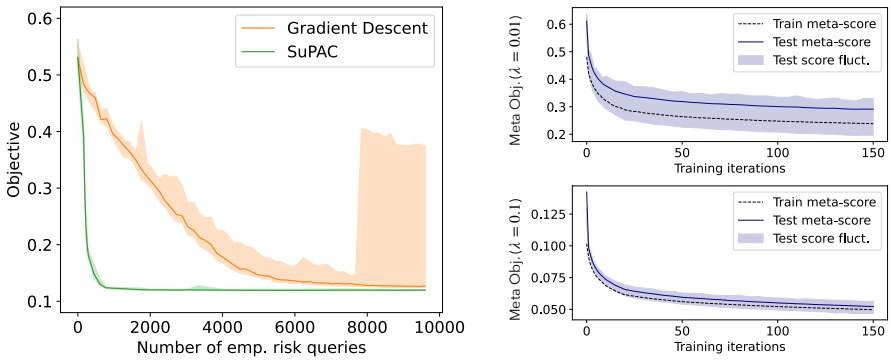
We consider that assumptions *(A_1)* to *(A_5)* hold, and also these further mild assumptions: the prior is looked for in Π, i.e $\pi = \pi_{\theta_p}$; the PAC-Bayes bound PB is differentiable w.r.t. θ_p. Then, noting $\hat{\theta}_i$ the posterior parameter for each task, a simplification of the meta gradient occurs:

$$\nabla M(\theta_p) = \partial_{\theta_p} \mathrm{PB}(\hat{\theta}_i(\theta_p), R_i, \theta_p, \eta_i) = \partial_3 \mathrm{PB}(\hat{\theta}_i, R_i, \theta_p, \eta_i). \tag{5}$$

Remarkably, the knowledge of the derivative of $\hat{\theta}_i$ with respect to θ_p is not required to compute the meta gradient. This is due to ∂_1PB being 0 when evaluated for the prior posterior. We stress that such a simplification is specific to our meta-learning objective. It does not occur in meta-learning strategies such as MAML [Finn et al., 2017], where the performance of each task is assessed on a test set. In the context of PAC-Bayes, such reliance on test sets can be optimistically replaced by the PAC-Bayes bounds, which give test guarantees with high probability. It is unclear whether such a simplification occurs in previous PAC-Bayes Meta Learning objectives from the literature, as these involve distributions on priors rather than a single prior.

A key consequence is that training the meta learning algorithm is as hard as cycling all the Bayesian optimisation tasks. In a nutshell, meta learning is as hard as re optimising the bound for a new prior.

SuPAC -CE brings two main benefits when used in conjunction with meta-learning. First, by improving the optimisation efficiency for a given prior, SuPAC -CE speeds up the meta-learning procedure. Second, the "generation agnostic" weighing approach implies that risk revaluations from previous optimisation procedures can be reused. As a consequence, re optimisation of a PAC-Bayes bound for a new prior can conceivably be performed with few risk queries, bringing an additional speed-up. Moreover, the setting considered for SuPAC -CE enjoys



(a) Optim. perf.

(b) Meta Learning

Figure 1: Experiments results. Figure 1a compares the optimisation performance of our algorithm SuPAC -CE with gradient descent approaches on an biochemical calibration task. Optimisation procedures were repeated 20 times; median performance and quantiles 0.2 and 0.8 are represented. Figure 1b investigates train and test performance of the meta-learning approach of Section 5. Mean test performance, as well as quantiles 0.2 and 0.8 for the sequence of built prior is assessed on 40 tasks and compared to the train performance.

an analytical expression for meta-gradients, $\nabla M(\theta_p) = \sum_i \lambda_i (\nabla g(\hat{\theta}_i) - \nabla g(\theta_p))$ which can be efficiently evaluated.

6 Experiments

SuPAC -CE was assessed on the learning task described by Picard-Weibel et al. [2024]. A PAC-Bayes bound is minimised on Gaussian distributions with block diagonal covariance in order to calibrate 30 parameters of a biological inspired numerical model describing anaerobic digestion processes, ADM1 [Batstone et al., 2002]. This model relies on solving a stiff ODE to predict the evolution of the states, and is therefore quite computationally intensive (about 3 seconds per model query in our experiments).

We compared SuPAC -CE to standard GD on a synthetic dataset from Picard-Weibel et al. [2024], using the same family of distributions and risk function. For SuPAC -CE , 160 risk queries where performed for the initial step, and 32 for all further step. A maximal budget of 9600 empirical risk queries was fixed; hyperparameters for the GD were selected after evaluating a grid on the first 1600 queries. Mean risks were assessed at test time by resampling new predictors from the posterior. The PAC-Bayes temperature was set to 0.002. Training procedures were repeated 20 times.

The performance of the sequence of posteriors were compared by aligning the number of empirical risk queries. Indeed, the main motivation of SuPAC

-CE is the setting when querying the empirical risk is computationally expensive, and can be assumed to be the computational bottleneck. This is indeed the case for the anaerobic digestion example considered here. At equal number of risk queries, SuPAC -CE required an extra 3.5% processing time compared to gradient descent, mainly caused by the weighing process.

SuPAC -CE proved significantly more efficient at minimising the bound than GD (see Figure 1a). The average performance of our algorithm proved better after 1800 queries than the best performance obtained after the full 9600 queries for GD. The experiments also indicate that our procedure offers much higher stability compared to GD, both during training and between the training duplicates. This could be attributed to the "generation agnostic" weighing approach, which relies on all previous risk evaluations at each step and is thus more stable. On the other hand, the noisy gradients estimates have some probability of leading to problematic steps during GD, leading to sharp increase in the objective. In our experiments, 4 out of 20 GD procedures thus led to a worse performance than the one obtained by a single optimisation step of SuPAC -CE . The posterior distributions constructed through SuPAC -CE obtained an average empirical risk of 0.102 ± 0.003, similar to the 0.101 value reported in Picard-Weibel et al. [2024]. The resulting PAC-Bayes bound proved also similar (0.121 ± 0.004 vs. 0.122). Thus SuPAC -CE constructed as good a posterior as Picard-Weibel et al. [2024], but twenty times faster.

Further assessments of SuPAC -CE 's performance for other hyperparameters values and comparison to Nesterov accelerated GD were also conducted. SuPAC -CE proved to have a stable behaviour for a wide range of hyperparameters value ($0.25 \leq \alpha_{\max} \leq 0.75$, $0.5 \leq \mathrm{kl}_{\max} \leq 2$), with instabilities starting to appear for $\mathrm{kl}_{\max} > 5$, and speed decrease for $\mathrm{kl}_{\max} < 0.1$. Nesterov acceleration, requiring some iterations to build up momentum, proved unable to compete with SuPAC -CE 's almost instantaneous optimisation. Results for these experiments can be found in the appendix B.

Preliminary experiments were also performed for the meta-learning objective described in section 5. To facilitate the evaluation of the learnt meta priors, wholly synthetic risk functions were used in this case, and PAC-Bayes objective minimised on Gaussian distributions. The risk functions considered were bounded, smooth functions of $\mathbb{R}^8$, achieving their global minima at $x_0 \sim \mathbb{N}(\tilde{x}_0, \Sigma_0)$. $\tilde{x}_0$ was chosen so that $\|\tilde{x}_0\| = 2$, and Σ_0 such that only two of its eigenvalues are higher than 0.05^2 (drawn at random between $\exp(-1)$ and $\exp(1)$). Such choices ensure that the original prior distribution, $\mathcal{N}(0, I_k)$, can be improved upon both by shifting its mass centre and adjusting its covariance. The performance of the meta-learning algorithm was assessed for two temperatures, $\lambda = 0.1$ and $\lambda = 0.01$. Meta training was performed using stochastic gradient descent. The sequence of prior thus constructed was evaluated on a further 40 test tasks, each time restarting the optimisation procedure from scratch, and evaluating the final score on 10^4 draws from the posterior.

The meta-learning algorithm was able to satisfactorily reduce the objective, from an initial average generalisation bound of 0.61 (resp. 0.14) to 0.24 (resp. 0.050) after 150 gradient steps for $\lambda = 0.1$ (resp. $\lambda = 0.01$). Most of the

meta-objective reduction takes place during the early phase of training, with the first 15 steps amounting to more than 80 % of the objective decrease. For both temperatures tested, the average performance on the test tasks followed the objective decrease throughout training, even though the number of queries per optimisation was minimal after the first meta step (less than 40), supporting both our meta-learning objective and the use of SuPAC -CE .

Full implementation details on the experiments can be found in appendix B and in the source code

7 Discussions

The present work shows that it is possible to locally decouple the complexity related to querying the empirical risk and the minimisation of a PAC-Bayes bound. A main motivation for such decoupling is that the approximated risk function define non linear surrogate objectives which might be valid for a wider range of probabilities than the linear approximations offered by the gradients. As a consequence, the surrogate bound solution can be reasonably allowed to be much further away from the current posterior estimation than is the case for GD. A key implementation difficulty remains picking the range of validity, i.e. how far away from the current posterior the surrogate solver can be allowed to choose a distribution. Such a choice, formalised in the selection of an adequate surrogate solving algorithm, is analogue to the choice of a step size in gradient descent procedures, and balances the stability and speed of the procedure. Automating the selection of the surrogate validity range offers an exciting prospect for the framework.

The Voronoi cell weighing approach used to solve the approximation problem is equivalent to replacing the empirical risk function by a 1-nearest neighbour trained predictor, and approximating this predictor. Variants following this two step approximation approach could be worth investigating. Notably, an interesting perspective would be to approximate the empirical risk through Gaussian processes, taking inspiration from Gaussian Optimisation. This would notably track the uncertainty on the approximate risk on extrapolated values, which could drive the choice of new predictors to evaluate and improve on the current random draws.

A key restriction of the present work is that our surrogate PAC-Bayes framework is only practicable when the dimension of the predictor space and of the probability family are small (i.e. less than a few hundreds). This is due to two factors; first of all, the larger the dimension of the probability family, the larger becomes the approximation space, and hence the more empirical risk evaluations are required. Notably, at least $d + 1$ evaluations of the empirical risk are required for probability families of dimension d. The second factor is that the "generation agnostic" weighing approach described in section 4.2.1 is unlikely to give adequate performances if $\mathcal{H}$ is high dimensional. This effectively rules out deep learning settings, which have been recently the main focus of the PAC-Bayes community. Still, we believe that PAC-Bayes learning offers

meaningful prospects for a wide range of physics, biology or medical inspired problems which involve few parameters and expensive model computations, and therefore can be efficiently trained using our framework. Concrete fields of application of SuPAC -CE include, but are not limited to, fluid dynamics simulations with dimension reduction [Callaham et al., 2021], metabolic models for microbial communities [Cerk et al., 2024] and greenhouse gas emission inverse problems [Nalini et al., 2022].

8 Conclusion

We introduced a generic framework for minimising PAC-Bayes bounds designed to tackle computationally intensive empirical risks for low to moderate dimensional problems such as naturally arise in physical models. We established that our optimisation strategy was theoretically well supported. We instantiated this framework for the optimisation of bounds on exponential family, and considered how this implementation could interact with meta-learning. Preliminary experiments showed that our framework could significantly reduce the number of empirical risks queries when calibrating a biochemical model, thus opening exciting new fields of applications for PAC-Bayes.

Acknowledgements

We warmly thank reviewers and the Area Chair who provided insigthful comments and suggestions which greatly helped us improve our manuscript. B.G. acknowledges partial support by the U.S. Army Research Laboratory and the U.S. Army Research Office, and by the U.K. Ministry of Defence and the U.K. Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/R013616/1; B.G. also acknowledges partial support from the French National Agency for Research, grants ANR-18-CE40-0016-01 and ANR-18-CE23-0015-02

References

Pierre Alquier. User-friendly introduction to pac-bayes bounds. *Foundations and Trends® in Machine Learning*, 17(2):174–303, 2024. ISSN 1935-8237. doi: 10.1561/2200000100. URL http://dx.doi.org/10.1561/2200000100.

Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research (JMLR)*, 17(236):1–41, 12 2016.

Ron Amit and Ron Meir. Meta-learning by adjusting priors based on extended pac-bayes theory. In *International Conference on Machine Learning*, pages 205–214. PMLR, 2018.

D.J. Batstone, J. Keller, I. Angelidaki, S.V. Kalyuzhnyi, S.G. Pavlostathis, A. Rozzi, W.T.M. Sanders, H. Siegrist, and V.A. Vavilin. The iwa anaerobic digestion model no 1 (adm1). *Water Science and Technology*, 45(10):65–73, May 2002. ISSN 1996-9732. doi: 10.2166/wst.2002.0292. URL http://dx.doi.org/10.2166/wst.2002.0292.

Felix Biggs and Benjamin Guedj. Differentiable PAC–Bayes objectives with partially aggregated neural networks. *Entropy*, 23(10), 9 2021. doi: 10.3390/e23101280.

L.D. Brown. Fundamentals of statistical exponential families: with applications in statistical decision theory. In *Statistics*. Ims, 1986. ISBN 0940600102.

J. L. Callaham, J.-C. Loiseau, G. Rigas, and S. L. Brunton. Nonlinear stochastic modelling with langevin regression. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 477(2250), June 2021. ISSN 1471-2946. doi: 10.1098/rspa.2021.0092. URL http://dx.doi.org/10.1098/rspa.2021.0092.

O. Catoni. PAC-Bayesian supervised classification: The thermodynamics of statistical learning. *IMS Lecture Notes Monograph Series*, 2007.

Olivier Catoni. *Statistical Learning Theory and Stochastic Optimization*. Lecture Notes in Mathematics: Saint-Flour Summer School on Probability Theory XXXI 2001. 2004. doi: 10.1007/b99352.

Klara Cerk, Pablo Ugalde-Salas, Chabname Ghassemi Nedjad, Maxime Lecomte, Coralie Muller, David J. Sherman, Falk Hildebrand, Simon Labarthe, and Clémence Frioux. Community-scale models of microbiomes: Articulating metabolic modelling and metagenome sequencing. *Microbial Biotechnology*, 17(1), January 2024. ISSN 1751-7915. doi: 10.1111/1751-7915.14396. URL http://dx.doi.org/10.1111/1751-7915.14396.

Loren Cobb, Peter Koppstein, and Neng Hsin Chen. Estimation and moment recursion relations for multimodal distributions of the exponential family. *Journal of the American Statistical Association*, 78(381):124–130, 1983.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2024. URL https://arxiv.org/abs/2401.08281.

Gintare K. Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proc. Conf. Uncertainty in Artif. Intell. (UAI)*, Sydney, Australia, 8 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine*

Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proc. Int. Conf. Mach. Learning (ICML)*, Montreal, Canada, 06 2009. doi: 10.1145/1553374.1553419.

Pascal Germain, Alexandre Lacasse, Francois Laviolette, Mario March, and Jean-Francis Roy. Risk bounds for the majority vote: From a PAC-Bayesian analysis to a learning algorithm. *Journal of Machine Learning Research (JMLR)*, 16 (26):787–860, 2015.

B. Guedj. A primer on PAC-Bayesian learning. *Proceedings of the Second Congress of the French Mathematical Society*, 2019.

Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. Generalization bounds: Perspectives from information theory and pac-bayes, 2023. URL https://arxiv.org/abs/2309.04381.

Achim Klenke. *Probability theory: a comprehensive course*. Springer, 2020.

Serge Lang. *Fundamentals of Differential Geometry*. Springer New York, 1999. ISBN 9781461205418. doi: 10.1007/978-1-4612-0541-8. URL http://dx.doi.org/10.1007/978-1-4612-0541-8.

Andreas Maurer. A note on the pac bayesian theorem, 2004. URL https://arxiv.org/abs/cs/0411099.

D.A. McAllester. PAC-Bayesian model averaging. *COLT*, 1999.

Zakaria Mhammedi, Peter Grünwald, and Benjamin Guedj. PAC-Bayes unexpected Bernstein inequality. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, volume 32, Vancouver, Canada, 12 2019.

K. Nalini, T. Lauvaux, C. Abdallah, J. Lian, P. Ciais, H. Utard, O. Laurent, and M. Ramonet. High-resolution lagrangian inverse modeling of co2 emissions over the paris region during the first 2020 lockdown period. *Journal of Geophysical Research: Atmospheres*, 127(14), July 2022. ISSN 2169-8996. doi: 10.1029/2021jd036032. URL http://dx.doi.org/10.1029/2021JD036032.

Behnam Neyshabur, Srinadh Bhojanapalli, David A. Mcallester, and Nati Srebro. Exploring generalization in deep learning. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, 12 2017.

Anastasia Pentina and Christoph Lampert. A pac-bayesian bound for lifelong learning. In *International Conference on Machine Learning*, pages 991–999. PMLR, 2014.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *Journal of Machine Learning Research (JMLR)*, 22(227):1–40, 2021.

Antoine Picard-Weibel, Gabriel Capson-Tojo, Benjamin Guedj, and Roman Moscoviz. Bayesian uncertainty quantification for anaerobic digestion models. *Bioresource Technology*, 394:130147, February 2024. ISSN 0960-8524. doi: 10.1016/j.biortech.2023.130147. URL http://dx.doi.org/10.1016/j.biortech.2023.130147.

Omar Rivasplata, Vikram M Tankasali, and Csaba Szepesvari. PAC-Bayes with backprop. *arXiv*, 10 2019. doi: 10.48550/arxiv.1908.07380.

Jonas Rothfuss, Martin Josifoski, Vincent Fortuin, and Andreas Krause. Scalable pac-bayesian meta-learning via the pac-optimal hyper-posterior: From theory to practice. *Journal of Machine Learning Research*, 24:1–62, 2023.

Yevgeny Seldin and Naftali Tishby. PAC-Bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research (JMLR)*, 11(117):3595–3646, 12 2010.

John Shawe-Taylor and Robert C. Williamson. A PAC analysis of a Bayesian estimator. In *Proc. Conf. Learn. Theory (COLT)*, 7 1997. doi: 10.1145/267460.267466.

Paul Viallard, Maxime Haddouche, Umut Şimşekli, and Benjamin Guedj. Learning via Wasserstein-based high probability generalisation bounds. *arXiv*, 6 2023. doi: 10.48550/arXiv.2306.04375.

Hossein Zakerinia, Amin Behjati, and Christoph H Lampert. More flexible pac-bayesian meta-learning by learning learning algorithms. *arXiv preprint arXiv:2402.04054*, 2024.

Valentina Zantedeschi, Paul Viallard, Emilie Morvant, Rémi Emonet, Amaury Habrard, Pascal Germain, and Benjamin Guedj. Learning stochastic majority votes by minimizing a PAC-Bayes generalization bound. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Virtual Conference, 12 2021.

W. Zhou, V. Veitch, M. Austern, R.P. Adams, and P. Orbanz. Non-vacuous generalization bounds at the ImageNet scale: a PAC-Bayesian compression approach. In *Proc. Int. Conf. Learn. Representations (ICLR)*, New Orleans, LA, 5 2019.

A Technical proofs

A.1 Proof of Corollary 1

As assumptions (A_1) to (A_5) hold, Theorem 3 can be used. It implies that replacing R by $f^{R,\theta}$ does not change the gradient of PB. Hence, starting from

$\theta = \theta^*$, since $\partial_1 \mathrm{PB}(\theta^*, R, \pi_\mathrm{p}, \gamma) = \partial_1 \mathrm{PB}(\theta^*, f^{R,\theta^*}, \pi_\mathrm{p}, \gamma) = 0$, the update step in the solving strategy satisfies $\mathrm{Updt}(\theta^*) = \theta^* - M(\pi, \theta, f, \gamma) \times 0 = \theta^*$. Hence, by recursion, it follows that $\mathrm{Solve}(\pi_\mathrm{p}, \pi_{\theta^*}, f^{R,\theta^*}) = \pi_{\theta^*}$. Since we assume that $\mathrm{F}(\pi_\theta, R) = f^{R,\theta}$, this implies that π^{θ^*} is a fixed step of $\pi \to \mathrm{Solve}(\pi_\mathrm{p}, \pi, \mathrm{F}(\pi, R)))$, and hence that the posterior is a fixed point of SuPAC for the specified F and Solve strategies, concluding the proof.

A.2 Proof of Lemma 1

We consider the broader problem where the prior π_p might not belong to the exponential family, but any probability satisfying the following assumptions:

Assumptions. *(A_6) π_p is absolutely continuous with respect to π_{ref};*

(A_7) $\forall \theta \in \Theta$, $h := \log\left(\frac{d\pi_\mathrm{p}}{d\pi_{ref}}\right) \in L^2(\pi_\theta)$.

Note that when $\pi_\mathrm{p} \in \Pi$, one can use $\pi_\mathrm{ref} = \pi_p$ for which assumptions *(A_6)* and *(A_7)* are automatically fulfilled. The generalisation of the approximation space becomes

$$\mathcal{F} = \{f_{\eta,C} := \theta \cdot T + C + \lambda h\},$$

which fits into the framework described in Theorem 1. For any $f_\eta \in \mathcal{F}$, the solver of Catoni's bound on all distributions is given by $\tilde{\theta} = -\lambda^{-1}\eta$, provided this defines a probability distribution (else Catoni's bound does not reach its minima on Π or $\mathcal{P}$). Note that the choice of $\tilde{\theta}$ is coherent with the formula given in Lemma 1 when the prior belongs to Π, since in that case $h = \theta_p \cdot T$, leading to a change of coordinate in the definition of $\mathcal{F}$.

Under the assumptions, Catoni's bound is differentiable and its gradient with respect to θ can be computed under the integral. Thus, for score f_η,

$$\begin{aligned}
\nabla \mathrm{PB}_\mathrm{Cat} &= \pi_\theta[f_\eta(T - \nabla g(\theta))] + \lambda \pi_\theta[(\theta \cdot T - g(\theta) - h)(T - \nabla g(\theta))] \\
&= \pi_\theta[(f_\eta + \lambda\theta \cdot T - g(\theta) - \lambda h)(T - \nabla g)] \\
&= \pi_\theta[(f_\eta + \lambda\theta \cdot T - \lambda h)(T - \pi_\theta[T])] \\
&= \pi_\theta[(\eta \cdot T + C)(T - \pi_\theta[T])] + \lambda \mathbb{V}_{\pi_\theta}[T]\theta \\
&= \mathbb{V}_{\pi_\theta}[T](\eta + \lambda\theta)
\end{aligned}$$

where we use the well known identity $\pi_\theta[T] = \nabla g$ (see Brown [1986]). For exponential families, the variance $\mathbb{V}_{\pi_\theta}[T]$ coincides with Fisher's information, and hence the previous equality reads $\nabla \mathrm{PB}_\mathrm{Cat} = \lambda I(\theta)(\theta - \tilde{\theta}(\eta))$, which implies Lemma 1.

A.3 Probability families with constant approximation space

Theorem 1 considers projections of the risk on a local vector space of functions $\mathcal{F}_\theta$. A special case of interest concerns families of distributions such that the approximation set is constant. Exponential families offer such a characteristic. We show here that exponential families (and its restrictions) are the only smoothly parameterised distributions with this characteristic:

Theorem 3. *For family of distributions satisfying the first three hypotheses of Section 3 such that, moreover:*

- Θ *is a connected,*
- $\theta \to \ell(\theta, x)$ *is twice continuously differentiable for all* x*.*

If there exists a vector space of finite dimension $\mathcal{F}$ *such that* $\mathcal{F}_\theta \subset \mathcal{F}$ *for all* $\theta \in \Theta$*, then there exists an exponential family* Π_T *defined on* $\widetilde{\Theta}$ *and a connected, open set* Θ_Π *such that* $\Pi = \{\pi_\theta \mid \theta \in \Theta_\Pi\}$*.*

Proof. For $\mathcal{F}$ of dimension $\tilde{d}+1$, choose $T_1, \ldots, T_{\tilde{d}}, T_{\tilde{d}+1} = 1$ a basis of $\mathcal{F}$. Then, for all θ, there exists a unique matrix $A(\theta) \in \mathbb{R}^{d,\tilde{d}}$, and a unique vector $c \in \mathbb{R}^{\tilde{d},1}$ such that

$$\partial_\theta \ell = \begin{pmatrix} A(\theta) & c(\theta) \end{pmatrix} \begin{pmatrix} T_1 \\ \cdots \\ T_{\tilde{d}+1} \end{pmatrix}$$

Assume that $A(\theta)$ and $c(\theta)$ are differentiable (this is proved afterwards). Since ℓ is twice continuously differentiable, it follows $\partial_{\theta_i}\partial_{\theta_j}\ell = \partial_{\theta_j}\partial_{\theta_i}\ell$, and therefore that $\partial_{\theta_i} A_{j,k} = \partial_{\theta_j} A_{i,k}$ and that $\partial_{\theta_j} c_i = \partial_{\theta_i} c_j$. This, in conjunction with the hypothesis that Θ is connected, implies that $A(\theta)$ is a gradient of some $\beta : \mathbb{R}^d \mapsto \mathbb{R}^{\tilde{d}}$ while c is the gradient of some $-g : \mathbb{R}^d \mapsto \mathbb{R}$ (see Lang [1999]). Hence, $\ell(\theta) = \beta(\theta) \cdot T(x) - g(\theta) + h$ for h a solution of $\partial_\theta h = 0$. Since Θ is connected, this implies that h can not be a function of θ. Hence Π is the restriction of an exponential family on Θ.

It remains to show that $A(\theta)$ and $c(\theta)$ are differentiable. First of all, we remark that for all finite collection of linearly independent real valued functions $(f_1, \ldots, f_n)$, there exists d points $(x_1, \ldots, x_n$ such that $(f_i(x_j))_{i,j\leq n}$ is inversible. Indeed, this result holds for a single function, since f_1 must be non zero. Then if the result holds for $x_1, \ldots, x_k$, i.e. $D = \det((f_i(x_j))_{i,j\leq k}) \neq 0$ then consider the matrix $m(z) = (f_i(\tilde{x}_j)_{i,j\leq k+1}$ with $\tilde{x}_j = x_j$ if $j \leq k$, $\tilde{x}_{k+1} = z$. Then the determinant of matrix m is $Df_{k+1}(z) + \sum_{i\leq k} C_i f_i(z)$. Since $f_1, \ldots, f_{k+1}$ are linearly independent and since D is not zero, there must exist z such that $\det(m(z)) \neq 0$, which we can pick as x_{k+1}. This proves the result by recursion.

Since $T_1, \ldots T_{\tilde{d}+1}$ are linearly independent, we can therefore pick such $x_1, \ldots, x_{\tilde{d}+1}$. By definition of $A(\theta)$ and $c(\theta)$, it follows that for all θ,

$$\begin{pmatrix} A(\theta) & c(\theta) \end{pmatrix} = \begin{pmatrix} \partial_{\theta_1}\ell(\theta, x_1) & \ldots & \partial_{\theta_1}\ell(\theta, x_{\tilde{d}+1}) \\ \vdots & & \vdots \\ \partial_{\theta_k}\ell(\theta, x_1) & \ldots & \partial_{\theta_k}\ell(\theta, x_{\tilde{d}+1}) \end{pmatrix} \begin{pmatrix} T_1(x_1) & \ldots & T_1(x_{\tilde{d}+1}) \\ \vdots & & \vdots \\ T_{\tilde{d}+1}(x_1) & \ldots & T_{\tilde{d}+1}(x_{\tilde{d}+1}) \end{pmatrix}^{-1}$$

This implies that A and c are smooth functions of the differentiable $(\partial_\ell(\cdot, x_i))_{i\in[1,\tilde{d}+1]}$, and hence that they are differentiable.

□

A.4 Regularisation and convergence for Catoni's bound

If $R = f_\eta \in \mathcal{F}$, the uncorrected step direction results in one step convergence, implying that the update direction at θ is $\hat{\theta} - \theta$. This implies that all successive estimation θ_i belongs to the segment $[\theta_0, \hat{\theta}]$. Note $\Delta\theta = \hat{\theta} - \theta_0$. Since the normalisation function g is strictly convex, it follows that the function $t \to \Delta\theta \cdot \nabla g(\theta_0 + t\Delta\theta)$ is non decreasing, and hence, for all t,

$$\Delta\theta \cdot \nabla g(\theta_0) \leq \Delta\theta \cdot \nabla g(\theta_0 + t\Delta\theta) \leq \Delta\theta \cdot \nabla g(\hat{\theta}).$$

Using the convexity of g, this implies that for $t_1 < t_2$, $g(\theta_0 + t_1\Delta_\theta) - g(\theta_0 + t_2\Delta_\theta) \leq (t_1 - t_2)\Delta_\theta \cdot \nabla g(\theta_0)$ while $(t_2 - t_1)\Delta_\theta \cdot \nabla g(\theta + t_2\Delta\theta) \leq (t_2 - t_1)\Delta_\theta \cdot \nabla g(\hat{\theta})$.

It follows that for all $t_1 < t_2$,

$$\mathrm{KL}(\theta_0 + t_2\Delta\theta, \theta_0 + t_1\Delta\theta) \leq (t_2 - t_1)\Delta\theta \cdot (\nabla g(\hat{\theta}) - \nabla g(\theta_0)).$$

This implies that for $\theta_i = \theta_0 + t_i\Delta\theta$, $\theta_{i+1} = \theta_0 + t_{i+1}\Delta\theta$, if the condition $\mathrm{KL}(\theta_{i+1}, \theta_i) \leq \mathrm{kl}_{\max}$ is active, then $t_{i+1} - t_i \geq \frac{\mathrm{kl}_{\max}}{\Delta\theta \cdot (\nabla g(\hat{\theta}) - \nabla g(\theta_0))}$. Since $t_{i+1} - t_i \geq 0$ and for all i, $t_i \leq 1$, this implies that the condition is active a finite number of time at most. In the case of $\alpha_{\max} = 1$, this implies convergence in a finite number of steps. For $0 \leq \alpha_{\max} < 1$, this implies that after some K, $t_{i+K} = (1 - \alpha_{\max})^i(1 - t_K)$, and hence exponential convergence of (θ_i) to $\hat{\theta}$.

B Implementation details

The code described in this section can be found in the publication repo: https://tinyurl.com/surpbayes.

B.1 Further notes on SuPAC -CE

SuPAC -CE can be summarised in the following pseudo-code:

Algorithm 2 Surrogate Catoni solver for exponential families (SuPAC -CE)

Require: $\lambda > 0$, $\theta_0 \in \Theta$, $\theta_p \in \Theta$, $R \in \mathcal{M}(\mathcal{H})$, $\mathrm{Ev} = (x_i, R(x_i))_{i=1}^n$, $0 < \alpha_{\max} \leq 1$, $0 < \mathrm{kl}_{\max}$

$\theta \leftarrow \theta_0$

while not converged **do**

Draw i.i.d. $x_{n+1}, \ldots, x_{n+k} \sim \pi_\theta$

$\mathrm{Ev}, n \leftarrow \mathrm{Ev} \cup ((x_{n+1}, R(x_{n+1})), \ldots, (x_{n+k}, R(x_{n+k}))), n + k$

$\omega_i \leftarrow \pi[\overline{x}_i]$ ▷ Solving nearest neighbour problems

$\eta^*, C = \arg\inf_{\eta, C} \sum_{i \leq n} \omega_i (T(x_i) - R(x_i) - C)^2$

$\delta\theta = \theta_0 - \lambda^{-1}\eta^* - \theta$

$\tilde{\alpha} \leftarrow \sup\{\alpha \mid \alpha < \alpha_{\max}, \mathrm{KL}(\theta + \alpha\delta\theta, \theta) \leq \mathrm{kl}_{\max}\}$

$\theta \leftarrow \theta + \tilde{\alpha}\delta\theta$

end while

Our implementation is based on the pre-existing code source provided by Picard-Weibel et al. [2024]. Part of the original code was reworked to fit our new setting. New classes for exponential families of distributions were introduced, and implementation of the Gaussian family classes modified accordingly. A modular and generic solver class for the minimisation of Catoni's PAC-Bayes bound on exponential families was introduced, as well as more specific implementations for probability families outputting Gaussian distributions, using the Mahalanobis distance when approximating the weights. These solvers rely on closed form expressions for the Kullback–Leibler divergence and its derivative, inferred from the normalisation function and its derivatives.

The default weighing approach for the score approximation uses exact 1-NN for a user specified number of samples ("n_estim_weights" argument), performed using Faiss library [Douze et al., 2024]. Another weight approximation method, relying on approximate k-NN solving, is also provided.

The corrected update rule parameter $\tilde{\alpha}$ is estimated by dichotomy, using the fact that for all θ, $\delta\theta$, the function $\alpha \to \mathrm{KL}(\theta + \alpha\delta\theta, \theta)$ is not decreasing. The resulting $\tilde{\alpha}$ is guaranteed to result in a Kullback–Leibler step of less than $\mathrm{kl}_{\max}$.

B.2 Experiments

B.2.1 Catoni's bound minimisation

The implementation of ADM1 from Picard-Weibel et al. [2024] was used to perform the experiments, and slightly modified to benefit from just-in-time compilation. The dataset used was the training part of dataset "LF". The probability family (Gaussian with block diagonal covariance with fixed blocks) and prior distribution considered in the original paper was used. For SuPAC -CE , the regularisation hyperparameters were set to $\mathrm{kl}_{\max} = 1$ and $\alpha_{\max} = 0.5$, while the number of samples generated to evaluate the weights was set to 40 000. The optimisation algorithm was trained on 296 steps; for the initial step, 160 risk queries were performed, while for all the remaining steps, 32 risk queries were performed. This larger number of queries for the initial step is due to the necessity of having a least more evaluations than the dimension of the family of probability.

Hyperparameters for GD were selected after assessing the grid $(\mathrm{per\_step}, \mathrm{step\_size}) \in \{80, 160\} \times \{0.025, 0.05, 0.07\}$ on a preliminary 1600 score queries budget, with 20 repeats. The larger step size 0.07 was rejected due to its erratic behaviour between repeats, obtaining both optimal and worse GD performance. This erratic behaviour was also observed for step size 0.05 when estimating gradients from 80 risk queries. On the other hand, for per_step set to 160, the step size of 0.025 clearly under-performed compared to the step size of 0.05, although slightly more stable. This led to the selection of the two sets of hyperparameters, (per_step=80, step_size=0.025) and (per_step=160, step_size=0.05), which had similar performances. Both were assessed, and the set of hyperparameters obtaining the lowest score, (per_step=160, step_size=0.05), was kept for comparison (see appendix B.2.1).

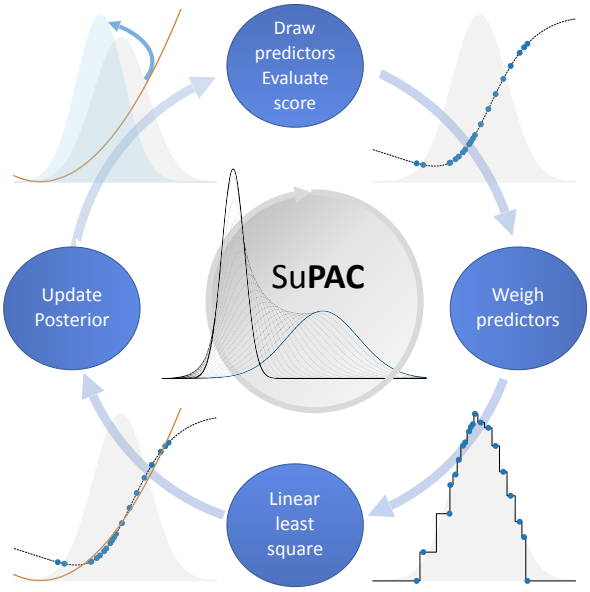
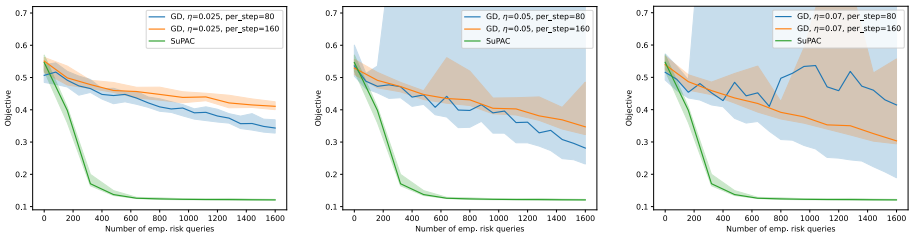


Figure 2: Overview of SuPAC -CE . At each step, some new predictors are drawn from the current posterior approximation and evaluated (top right figure). All evaluated predictors are then weighted according to the weight of their Voronoi cell (bottom right figure). These weighted evaluations are used to construct an optimal approximation of the score through a linear least square task (bottom left figure). The approximated score is used to update the posterior using a closed form expression (top left figure). This procedure is looped until convergence (center).



(a) $\eta = 0.025$ (b) $\eta = 0.05$ (c) $\eta = 0.07$

Figure 3: Preliminary GD optimisation procedures for different choices of hyperparameters. The evaluations of each optimisation procedure was repeated 20 times; the median performance and 0.2 and 0.8 quantiles are represented. The performance of SuPAC -CE is given for comparison.

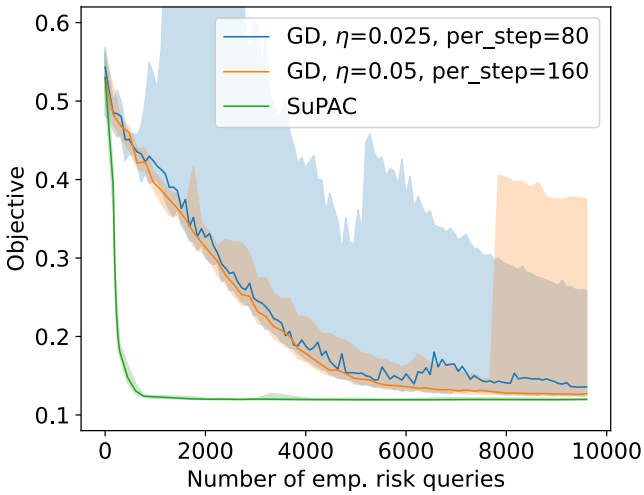


Figure 4: Comparison of the optimisation procedures as performed by SuPAC -CE and gradient descent (GD) for the two selected sets of hyperparameters. Each optimisation procedure was repeated 20 times; the median performance and 0.2 and 0.8 quantiles are represented. SuPAC -CE was performed with hyperparameters $\alpha_{\max} = 0.5$ and $\mathrm{kl}_{\max} = 1$.

SuPAC -CE was further compared to Nesterov accelerated gradient descent (implementation in the publication repo). Starting from the two sets of hyperparameters preselected for GD, optimisation procedures using a momentum of 0.5, 0.9 and 0.95, and either the original step size or twice the step size were assessed. Each of these 12 new optimisation procedures was repeated 8 times, and compared to SuPAC -CE (see appendix B.2.1). For no choice of hyperparameter values did Nesterov accelerated GD proved more efficient than SuPAC -CE (appendix B.2.1). The increase of step size in conjunction with the moderate momentum improved the speed of the optimisation procedure, but at the cost of a higher risk of optimisation failure, leading to 3 out of 8 runs (resp. 2 out of 8 runs) for 160 simulations per step (resp. 80 simulations per step) with a final objective higher than the initial objective. Higher momentum led to major instabilities, with less than 3 runs out of 8 managing to reduce the objective below 0.2 (compared to 0.121 obtained by SuPAC -CE) for all hyperparameter combinations. For the original step size, momentum appeared to improve the stability of the procedures for all setting except moderate momentum for a per step hyperparameter of 80. Higher momentum procedures led to a speed decrease, caused by the larger number of steps necessary for momentum to build up.

The impact of SuPAC -CE 's hyperparameters was investigated by running further optimisation procedures with different choices of hyperparameters. A grid was assessed, with values of $\mathrm{kl}_{\max}$ in $\{0.5, 1, 2\}$ and $\alpha_{\max}$ in $\{0.25, 0.5, 0.75\}$, with each optimisation process repeated ten times (see appendix B.2.1). The resulting optimisation procedures proved to all have similar performances, with only a

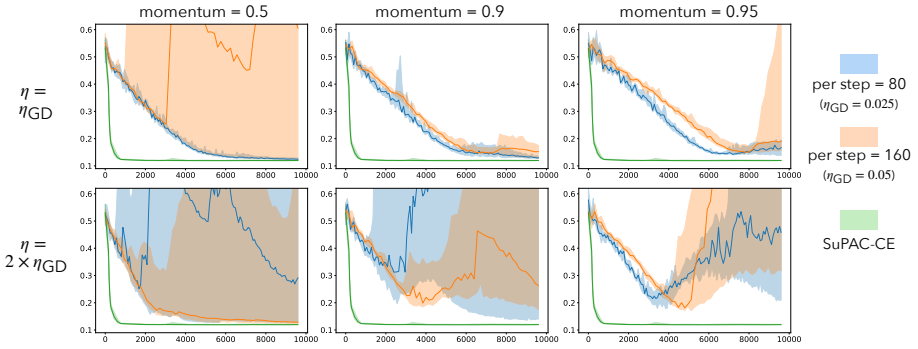


Figure 5: Comparison of the optimisation procedures as performed by SuPAC -CE and Nesterov accelerated gradient descent (x axis: number of empirical risk queries). Each optimisation procedure was repeated 8 times; the median performance and 0.2 and 0.8 quantiles are represented. SuPAC -CE was performed with hyperparameters $\alpha_{\max} = 0.5$ and $\mathrm{kl}_{\max} = 1$. Momentum of 0.5, 0.9 and 0.95 were assessed for Nesterov gradient descent. Both the original step size (η) parameter as well as twice the step size parameter for gradient descent comparisons were investigated. At twice the step size, all momentum accelerated procedures proved unstable. At the original step size, the momentum tended to increase the stability of the procedure at the cost of speed. All Nesterov accelerated gradient descent procedures assessed were slower than SuPAC -CE

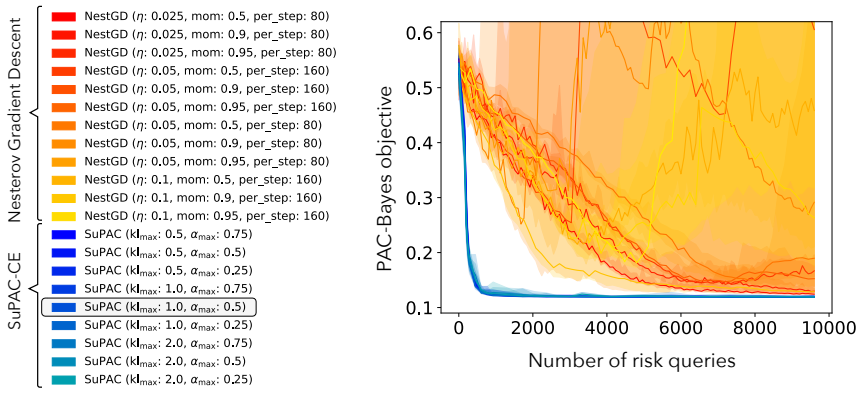


Figure 6: Comparison of SuPAC -CE with Nesterov accelerated gradient descent for a variety of hyperparameters choices. Each optimisation procedure was repeated 8 times; the median performance and 0.2 and 0.8 quantiles are represented. SuPAC -CE proved to be consistently more efficient for all hyperparameters values tested. The hyperparameter for SuPAC -CE assessed in the main part of the publication is highlighted.

slight decrease in speed in the early phase between the most regularized and less regularized hyperparameters which was below the noise level after the fourth optimisation step (see appendix B.2.1). Two further sets of slow hyperparameters values $((\mathrm{kl}_{\max}, \alpha_{\max}) \in \{(0.1, 0.9), (0.01, 0.9)\})$ and fast hyperparameters values $((\mathrm{kl}_{\max} = 5, \alpha_{\max} = 0.1), (\mathrm{kl}_{\max} = 10, \alpha_{\max} = 0))$ were also assessed, with 8 repeats (see appendix B.2.1). The slow hyperparameters led to more stable and reproducible optimisation procedures. For the small maximum step size of $\mathrm{kl}_{\max} = 0.01$, the average performance of the optimisation process was similar (i.e. difference below the noise level) to the performance of the optimisation process with standard hyperparameters after 2000 risk queries. The highest maximal step size assessed of $\mathrm{kl}_{\max} = 10$ resulted in a final average PAC-Bayes bound of 0.147 ± 0.022, with a standard deviation between runs of 0.061, significantly higher than the standard deviation for the standard hyperparameters (0.0032, p-value of $1.95e-09$).

Computations were performed using Azure Machine Learning compute clusters with 32 cores and Intel Xeon Platinum 8272CL processors.

B.3 Meta-Learning experiments

For the meta-learning experiments, the tasks were generated as follow. Empirical risk functions of form

$$R_{\omega,A,x_0} : x \mapsto \tanh(h(\omega \|A(x - x_0)\|^2)/10) \tag{6}$$

with $h(x) = \cos(x) + x$ were considered. These are such that x_0 is the only global minima of R_{ω,A,x_0}, while all xs such that $\omega\|A(x - x_0)\|^2 = \pi/2 + 2k\pi$ are local

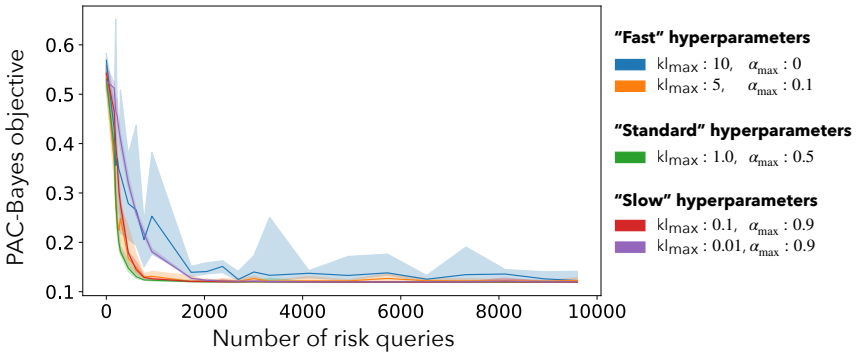


Figure 7: Performance of SuPAC -CE with extreme hyperparameters values. Each optimisation procedure was repeated 8 times; the median performance and 0.2 and 0.8 quantiles are represented. SuPAC -CE exhibited noticeable instabilities and speed loss for hyperparameters leading to insufficient regularization (blue curve). Too much regularisation lead to speed decrease in the early phase of the optimisation procedure (purple curve)

minima. The distributions of the risk parameters are as follow: $x_0 \sim \mathcal{N}(\tilde{x}_0, \Sigma_0)$, $\omega \sim \mathcal{U}(\frac{3}{2}\pi, \frac{5}{2}\pi)$ and $A_{i,j} \sim \mathcal{N}(\delta_{i,j}, \sigma^2 = 0.05^2)$. The mean parameter $\tilde{x}_0$ was initiated at random on the sphere of radius 2, while the covariance Σ_0 was initiated at random as

$$\Sigma_0 = O \times \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2) \times O^t,$$

where $\sigma_1, \ldots, \sigma_{d-2} = 0.05$, $\sigma_{d-1}, \sigma_d \sim \exp(\mathcal{U}(-0.5, 0.5))$ and O is drawn at random amongst orthonormal matrices. The dimension of the predictor space d is fixed to 8.

The meta training process was performed as follow. The initial calibration phase for each task was performed in 15 steps, with 100 score queries for the first five steps and 50 score queries for the remaining steps. The hyperparameters were set to $\mathrm{kl}_{\max} = 0.5$, $\alpha_{\max} = 0.3$ and 10^4 samples are used to estimate weights. This initial meta step used a mini batch size of 10, a maximum meta kl step of 0.2 and step size of λ^{-1}. After all tasks have been trained once, the hyperparameters for SuPAC -CE were modified: the number of steps was reduced to 4, and $\alpha_{\max}$ set to 0.7. 20 risk queries are performed on the first and third step, and none on the second and fourth. This accounts for the fact that the posterior distribution updates are expected to be small at this stage. The mini batch size is increased to 20. After 19 epochs, the step size is reduced to $0.5\lambda^{-1}$ and the maximum meta kl step to 0.1. After 30 more epochs, the step size was reduced to $0.4\lambda^{-1}$, and trained for a further 100 epochs.

The performance of sequence of priors was assessed in the following way. 40 test tasks were drawn. For each prior, a full independent calibration was performed on each task, using 20 steps of SuPAC -CE (100 risk queries for the first 5 steps, 50 for the remaining steps). The resulting posterior performance

is assessed by computing the bound using 10^4 fresh evaluations of the risk. The mean of these performance over the task defines the meta test score. The dispersion of these test performance between different test task is assessed by computing the quantiles 0.2 and 0.8 of the test performances at a given prior. This procedure being quite computationally intensive, only the first ten priors constructed and afterwards one prior out of five were assessed.

Computations were performed using Azure Machine Learning compute clusters with 16 cores and Intel Xeon Platinum 8272CL processors.