On generalisation and learning: A (condensed) primer on PAC-Bayes *followed by* News from the PAC-Bayes frontline

Benjamin Guedj
https://bguedj.github.io
@bguedj

Chalmers Machine Learning Seminar
April 19, 2021

Greetings!

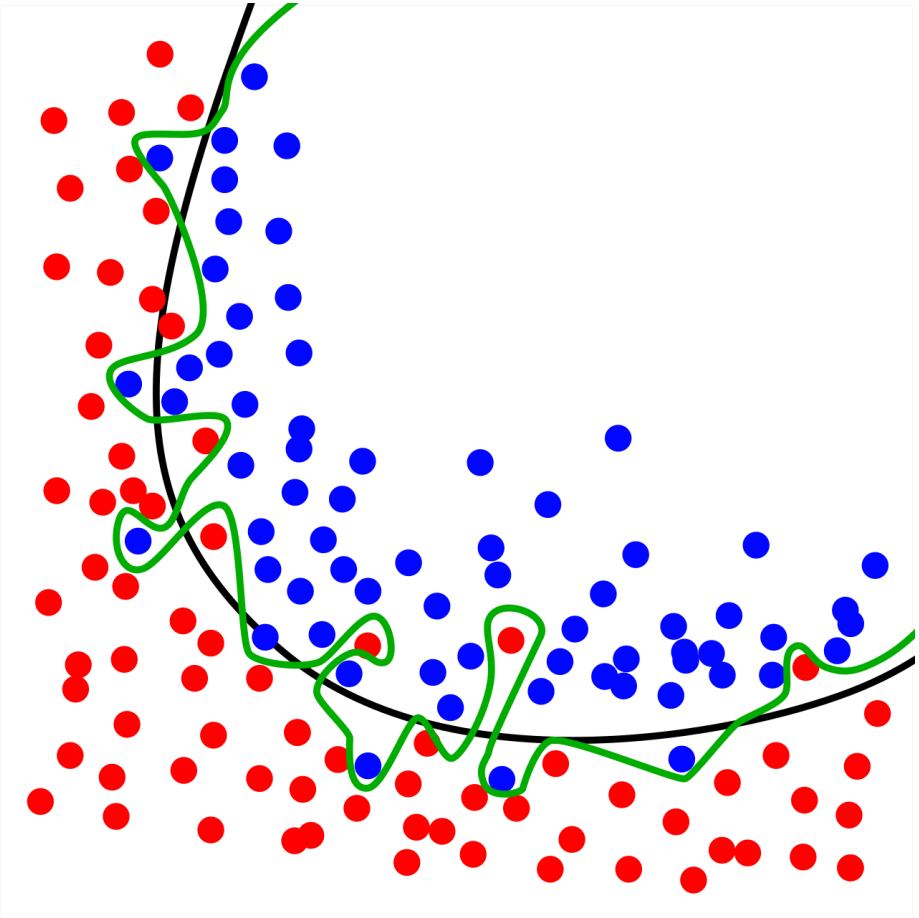
- Undergrad in pure mathematics, PhD in statistics (2013, Sorbonne Univ.) with G. Biau and E. Moulines
- Tenured research scientist at Inria since 2014 – Modal team, Lille - Nord Europe
- Principal research fellow at UCL since 2018, Dept. of Computer Science and Centre for AI, and visiting researcher at the Alan Turing Institute
- Scientific director of the Inria London Programme since 2020

Research at the crossroads of statistics, probability, machine learning, optimisation.

Statistical learning theory, PAC-Bayes, computational statistics, theoretical analysis of deep learning and representation learning to name but a few interests.

Personal obsession: generalisation.

Learning is to be able to generalise



[Credits: Wikipedia]

From examples, what can a system learn about the underlying phenomenon?

Memorising the already seen data is usually bad $\longrightarrow$ overfitting

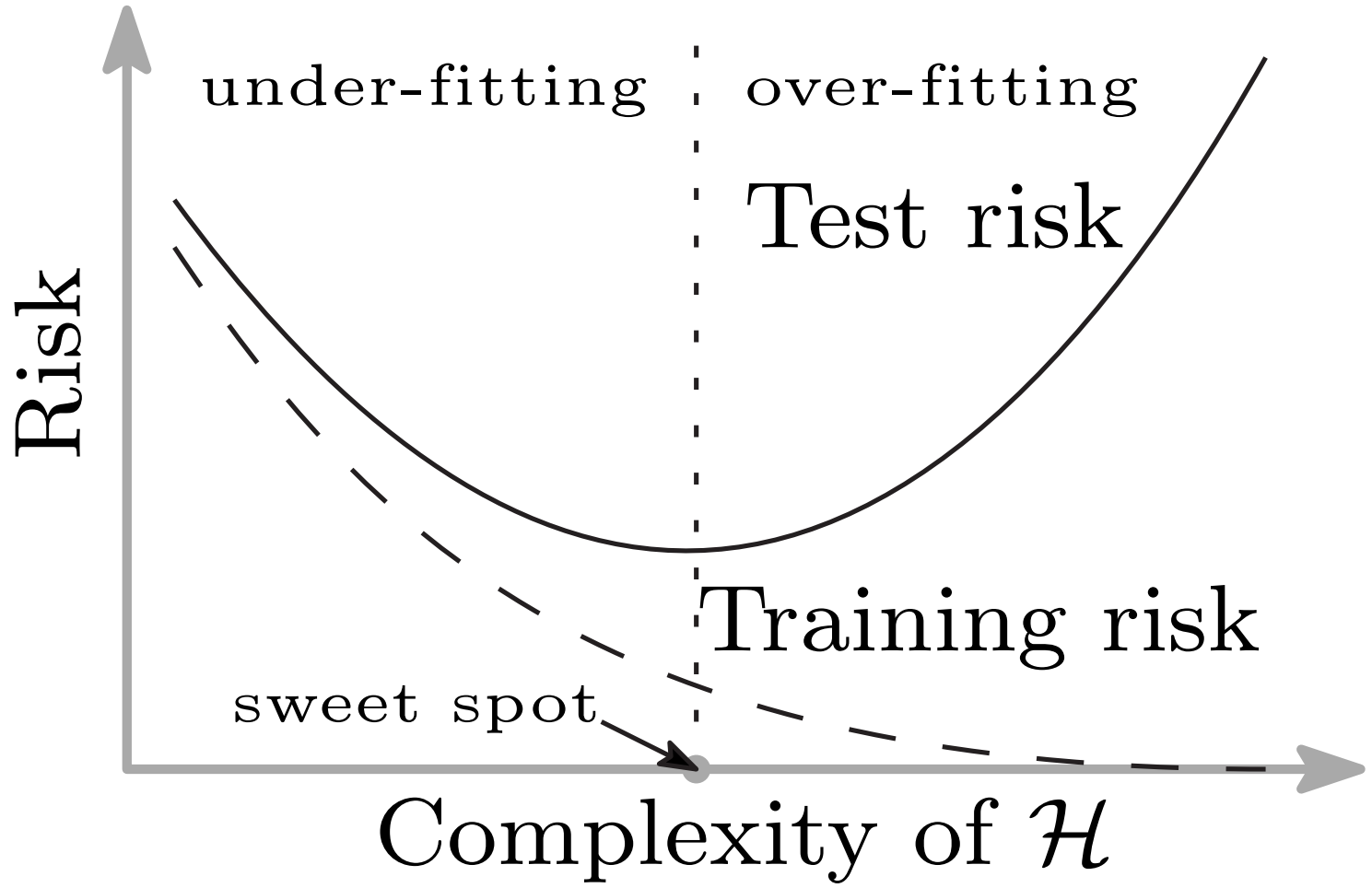
Generalisation is the ability to 'perform' well on unseen data.

Is deep learning breaking statistical learning theory?

Neural networks architectures trained on massive datasets achieve zero training error which does not bode well for their performance: this strongly suggests overfitting...

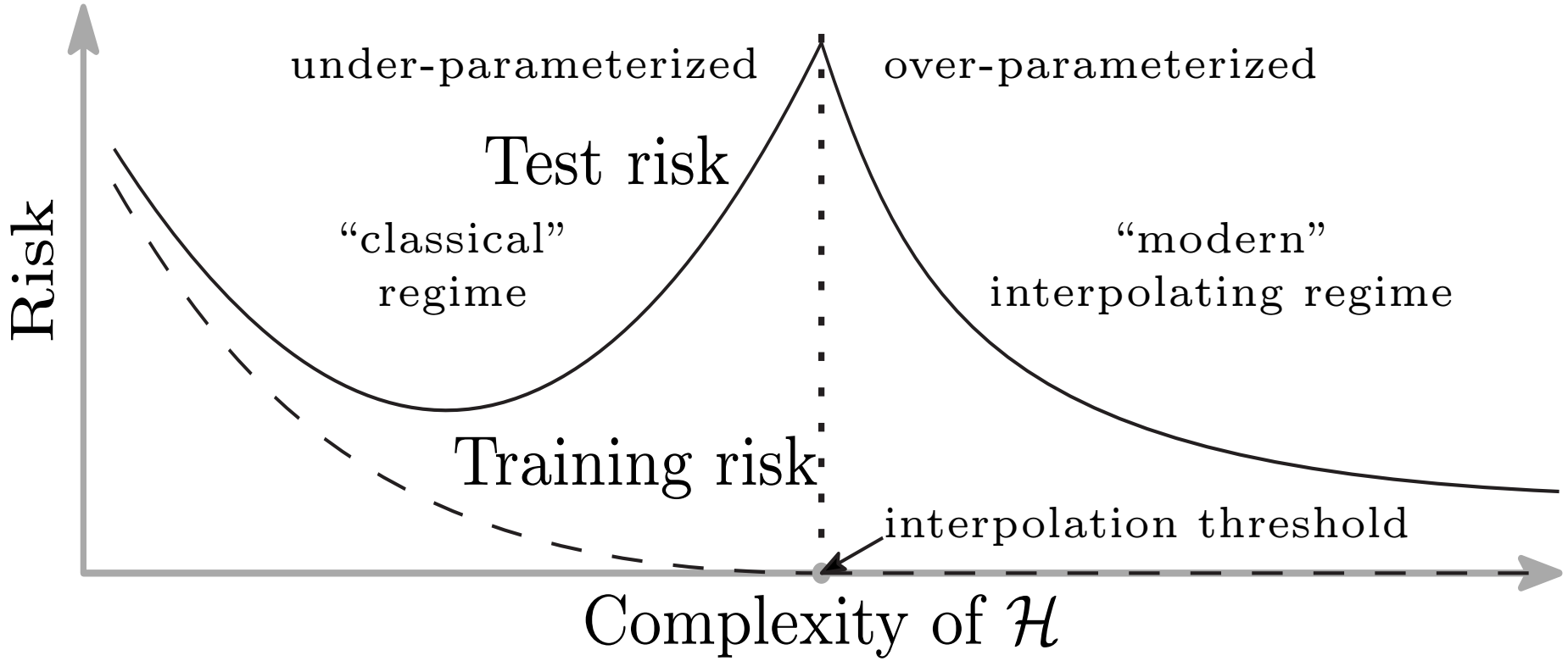
... yet they also achieve remarkably low errors on test sets!

A famous plot...



Belkin et al. (2019)

... which might just be half of the picture



Belkin et al. (2019)

Semantic representation to accelerate learning?

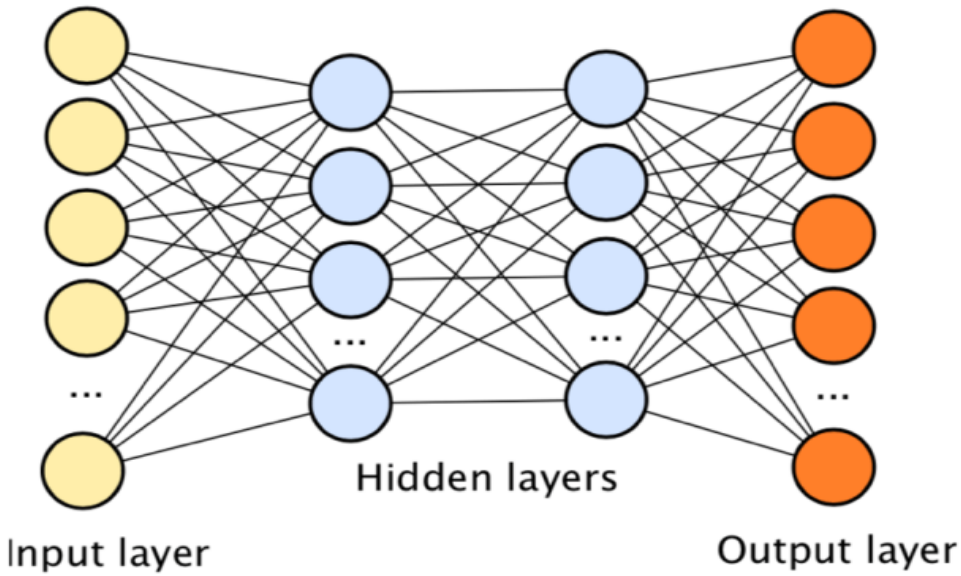
Fig. 1: What image representations do we learn by solving puzzles? Left: The image from which the tiles (marked with green lines) are extracted. Middle: A puzzle obtained by shuffling the tiles. Some tiles might be directly identifiable as object parts, but their identification is much more reliable once the correct ordering is found and the global figure emerges (Right).

Noroozi and Favaro (2016)

Semantic content of data is key! $\longrightarrow$ MURI project

A tale of two learners

First contender: a deep neural network



Typically identifies a specific item (say, a horse) in an image with accuracy $>$ 99%.
Training samples: millions of annotated images of horses – GPU-expensive training.

A tale of two learners

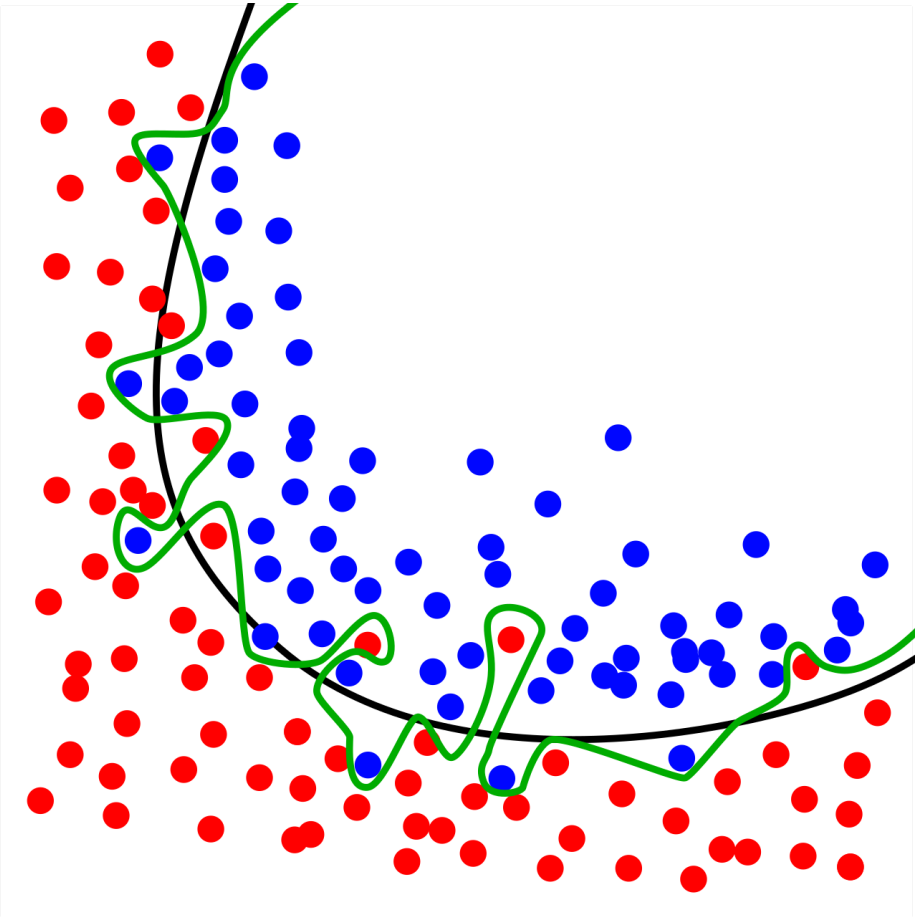
Second contender: my kids

Identify horses with 100% accuracy. Also very good at transferring to *e.g.* zebras

Training samples: a handful of children books, bedtime stories and (poorly executed) drawings.

Also expensive training.

Learning is to be able to generalise...



... but not from scratch! Tackling each learning task as a fresh draw unlikely to be efficient – must not be blind to context.

Need to incorporate structure / semantic information / implicit representations of the ”sensible” world.

Should lead to better algorithms design (more ”intelligent”, frugal / resources-efficient, etc.)

Part I

A Primer on PAC-Bayesian Learning
(short version of our ICML 2019 tutorial)

`https://bguedj.github.io/icml2019/index.html`

The simplest setting

Learning algorithm $\mathsf{A} : \mathcal{Z}^m \to \mathcal{H}$

- $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
- $\mathcal{H}$ = hypothesis class

Training set (aka sample): $S_m = ((X_1, Y_1), \ldots, (X_m, Y_m))$
a finite sequence of input-output examples.

- Data-generating distribution $\mathbb{P}$ over $\mathcal{Z}$.
- Learner doesn't know $\mathbb{P}$, only sees the training set.
- The training set examples are *i.i.d.* from $\mathbb{P}$: $S_m \sim \mathbb{P}^m$

Statistical Learning Theory is about high confidence

For a fixed algorithm, function class and sample size, generating random samples $\longrightarrow$ distribution of test errors

- Focusing on the mean of the error distribution?
  - ▷ can be misleading: learner only has one sample
- Statistical Learning Theory: tail of the distribution
  - ▷ finding bounds which hold with high probability over random samples of size m
- Compare to a statistical test – at 99% confidence level
  - ▷ chances of the conclusion not being true are less than 1%
- PAC: probably approximately correct (Valiant, 1984)
  Use a 'confidence parameter' δ: $\mathbb{P}^m[\text{large error}] \leqslant \delta$
  δ is the probability of being misled by the training set
- Hence high confidence: $\mathbb{P}^m[\text{approximately correct}] \geqslant 1 - \delta$

What to achieve from the sample?

Use the available sample to:

1. learn a predictor
2. certify the predictor's performance

Learning a predictor:

- algorithm driven by some learning principle
- informed by prior knowledge resulting in inductive bias

Certifying performance:

- what happens beyond the training set
- generalisation bounds

Actually these two goals interact with each other!

Generalisation

Loss function $\ell(h(X), Y)$ to measure the discrepancy between a predicted output $h(X)$ and the true output Y.

Empirical risk: $R_{\text{in}}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(X_i), Y_i)$
(in-sample)

Theoretical risk: $R_{\text{out}}(h) = \mathbb{E}\left[\ell(h(X), Y)\right]$
(out-of-sample)

If predictor h does well on the in-sample (X, Y) pairs...
...will it still do well on out-of-sample pairs?

Generalisation gap: $\Delta(h) = R_{\text{out}}(h) - R_{\text{in}}(h)$

Upper bounds: with high probability $\Delta(h) \leqslant \epsilon(m, \delta)$

- $R_{\text{out}}(h) \leqslant R_{\text{in}}(h) + \epsilon(m, \delta)$

Flavours:

- distribution-free
- distribution-dependent
- algorithm-free
- algorithm-dependent

The PAC (Probably Approximately Correct) framework

In a nutshell: with high probability, the generalisation error of an hypothesis h is at most something we can control and even compute. For any $\delta > 0$,

$$\mathbb{P}\left[R_{\text{out}}(h) \leqslant R_{\text{in}}(h) + \epsilon(m, \delta)\right] \geqslant 1 - \delta.$$

Think of $\epsilon(m, \delta)$ as $\text{Complexity} \times \frac{\log \frac{1}{\delta}}{\sqrt{m}}$.

This is about high confidence statements on the tail of the distribution of test errors (compare to a statistical test at level $1 - \delta$).

PAC-Bayes is about PAC generalisation bounds for ***distributions over hypotheses***.

Why should I care about generalisation?

Generalisation bounds are a safety check: they give a theoretical guarantee on the performance of a learning algorithm on any unseen data.

Generalisation bounds:

- provide a computable control on the error on any unseen data with prespecified confidence
- explain why some specific learning algorithms actually work
- and even lead to designing new algorithms which scale to more complex settings

Take-home message

PAC-Bayes is a generic framework to efficiently rethink generalisation for numerous statistical learning algorithms. It leverages the flexibility of Bayesian inference and allows to derive new learning algorithms.

ICML 2019 tutorial "A Primer on PAC-Bayesian Learning"
`https://bguedj.github.io/icml2019/`

Survey in the Journal of the French Mathematical Society: *Guedj (2019)*

NIPS 2017 workshop "(Almost) 50 Shades of Bayesian Learning: PAC-Bayesian trends and insights"
`https://bguedj.github.io/nips2017/`

Before PAC-Bayes

- Single hypothesis h (building block):

  with probability $\geqslant 1-\delta$, $\quad R_{\text{out}}(h) \leqslant R_{\text{in}}(h) + \sqrt{\frac{1}{2m}\log\left(\frac{1}{\delta}\right)}$.

- Finite function class $\mathcal{H}$ (worst-case approach):

  w.p. $\geqslant 1-\delta$, $\quad \forall h \in \mathcal{H},\ R_{\text{out}}(h) \leqslant R_{\text{in}}(h) + \sqrt{\frac{1}{2m}\log\left(\frac{|\mathcal{H}|}{\delta}\right)}$

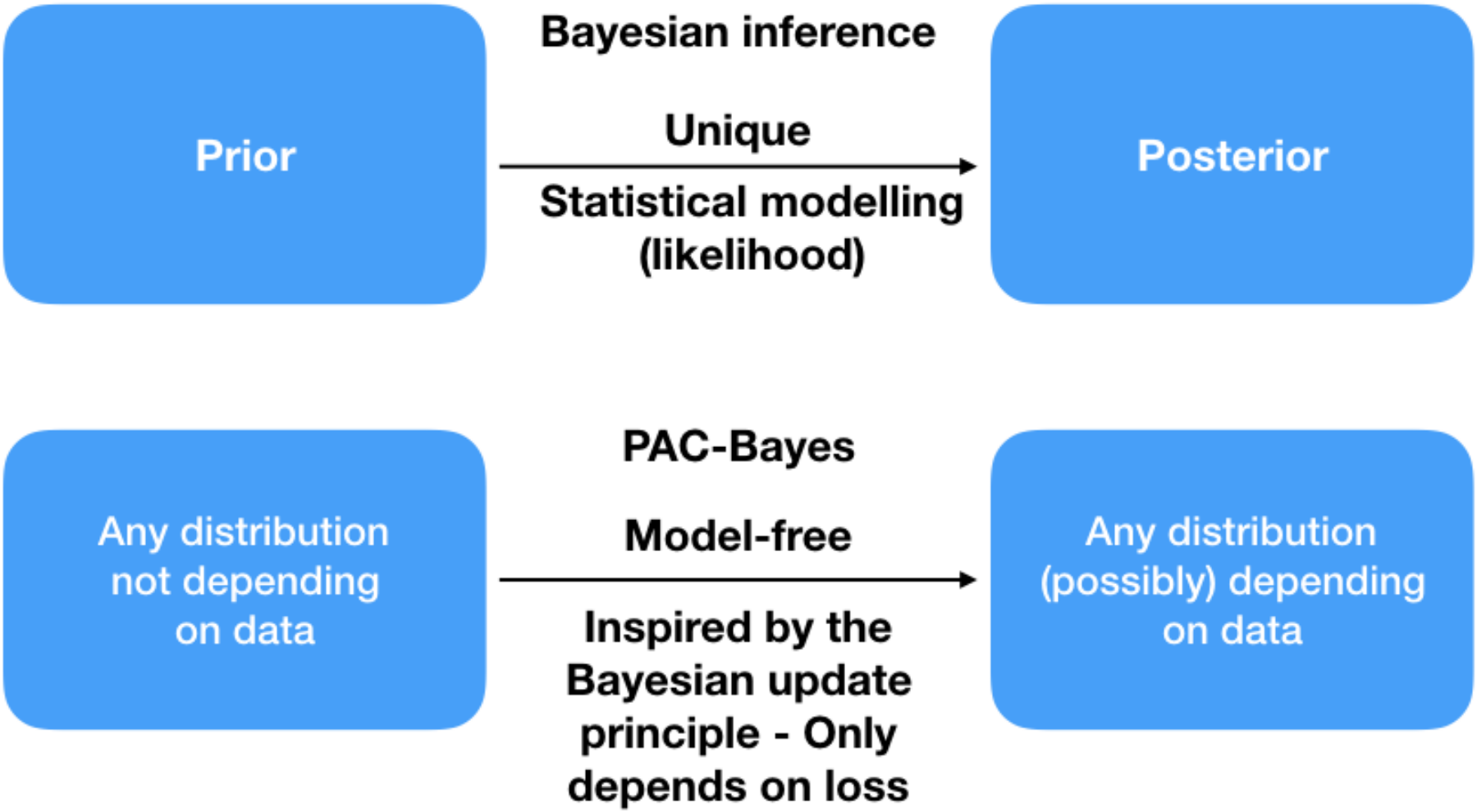
- Structural risk minimisation: data-dependent hypotheses h_i associated with prior weight p_i

  w.p. $\geqslant 1-\delta$, $\quad \forall h_i \in \mathcal{H},\ R_{\text{out}}(h_i) \leqslant R_{\text{in}}(h_i) + \sqrt{\frac{1}{2m}\log\left(\frac{1}{p_i\delta}\right)}$

- Uncountably infinite function class: VC dimension, Rademacher complexity...

These approaches are suited to analyse the performance of individual functions, and take some account of correlations.
$\longrightarrow$ Extension: PAC-Bayes allows to consider *distributions* over hypotheses.

PAC-Bayes



"Prior": exploration mechanism of $\mathcal{H}$

"Posterior" is the twisted prior after confronting with data

PAC-Bayes bounds vs. Bayesian inference

Prior P, posterior $Q \gg P$. Define the risk of a distribution:

$$R_{\text{in}}(Q) \equiv \int_{\mathcal{H}} R_{\text{in}}(h)\, dQ(h) \qquad R_{\text{out}}(Q) \equiv \int_{\mathcal{H}} R_{\text{out}}(h)\, dQ(h)$$

Kullback-Leibler divergence $\mathrm{KL}(Q\|P) = \mathop{\mathbf{E}}_{h\sim Q} \ln \frac{Q(h)}{P(h)}$.

- Prior
  - PAC-Bayes: bounds hold for any distribution
  - Bayes: prior choice impacts inference
- Posterior
  - PAC-Bayes: bounds hold for any distribution
  - Bayes: posterior uniquely defined by prior and statistical model
- Data distribution
  - PAC-Bayes: bounds hold for any distribution
  - Bayes: statistical modelling choices impact inference

A classical PAC-Bayesian bound

Pre-history: PAC analysis of Bayesian estimators

Shawe-Taylor and Williamson (1997)

Birth: PAC-Bayesian bound

McAllester (1998, 1999)

Prototypical bound

For any prior P, any $\delta \in (0, 1]$, we have

$$\mathbb{P}^m\left(\forall\, Q \text{ on } \mathcal{H}\colon \; R_{\text{out}}(Q) \leqslant R_{\text{in}}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \ln \frac{2\sqrt{m}}{\delta}}{2m}}\right) \geqslant 1 - \delta\,,$$

PAC-Bayes-driven learning algorithms

With an arbitrarily high probability and for any posterior distribution Q,

$$\text{Error on unseen data} \leqslant \text{Error on sample} + \text{complexity term}$$

$$R_{\text{out}}(Q) \leqslant R_{\text{in}}(Q) + F(Q, \cdot)$$

This defines a principled strategy to obtain new learning algorithms:

$$h \sim Q^\star$$

$$Q^\star \in \arg\inf_{Q \ll P} \left\{ R_{\text{in}}(Q) + F(Q, \cdot) \right\}$$

(optimisation problem which can be solved or approximated by [stochastic] gradient descent-flavoured methods, Monte Carlo Markov Chain, variational inference...)

SVMs, KL-regularized Adaboost, exponential weights are all minimisers of PAC-Bayes bounds.

Variational definition of KL-divergence (Csiszár, 1975; Donsker and Varadhan, 1975; Catoni, 2004).

Let $(A, \mathcal{A})$ be a measurable space.

(i) For any probability P on $(A, \mathcal{A})$ and any measurable function $\phi : A \to \mathbb{R}$ such that $\int (\exp \circ \phi) \mathrm{d}P < \infty$,

$$\log \int (\exp \circ \phi) \mathrm{d}P = \sup_{Q \ll P} \left\{ \int \phi \mathrm{d}Q - \mathrm{KL}(Q, P) \right\}.$$

(ii) If ϕ is upper-bounded on the support of P, the supremum is reached for the Gibbs distribution G given by

$$\frac{\mathrm{d}G}{\mathrm{d}P}(a) = \frac{\exp \circ \phi(a)}{\int (\exp \circ \phi) \mathrm{d}P}, \quad a \in A.$$

$$\log \int (\exp \circ \phi) \mathrm{d}P = \sup_{Q \ll P} \left\{ \int \phi \mathrm{d}Q - \mathrm{KL}(Q, P) \right\}, \quad \frac{\mathrm{d}G}{\mathrm{d}P} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) \mathrm{d}P}.$$

Proof: let $Q \ll P$ and $P \ll Q$.

$$\begin{aligned} -\mathrm{KL}(Q, G) &= -\int \log \left(\frac{\mathrm{d}Q}{\mathrm{d}P} \frac{\mathrm{d}P}{\mathrm{d}G} \right) \mathrm{d}Q \\ &= -\int \log \left(\frac{\mathrm{d}Q}{\mathrm{d}P} \right) \mathrm{d}Q + \int \log \left(\frac{\mathrm{d}G}{\mathrm{d}P} \right) \mathrm{d}Q \\ &= -\mathrm{KL}(Q, P) + \int \phi \mathrm{d}Q - \log \int (\exp \circ \phi) \, \mathrm{d}P. \end{aligned}$$

$\mathrm{KL}(\cdot, \cdot)$ is non-negative, $Q \mapsto -\mathrm{KL}(Q, G)$ reaches its max. in $Q = G$:

$$0 = \sup_{Q \ll P} \left\{ \int \phi \mathrm{d}Q - \mathrm{KL}(Q, P) \right\} - \log \int (\exp \circ \phi) \, \mathrm{d}P.$$

Let $\lambda > 0$ and take $\phi = -\lambda R_{\mathrm{in}}$,

$$Q_\lambda \propto \exp\left(-\lambda R_{\mathrm{in}}\right) P = \arg\inf_{Q \ll P} \left\{ R_{\mathrm{in}}(Q) + \frac{\mathrm{KL}(Q, P)}{\lambda} \right\}.$$

Recap

What we've seen so far

- Statistical learning theory is about high confidence control of generalisation
- PAC-Bayes is a generic, powerful tool to derive generalisation bounds...
- ... and invent new learning algorithms with a Bayesian flavour
- PAC-Bayes mixes tools from statistics, probability theory, optimisation, and is now quickly re-emerging as a key theory and practical framework in machine learning

What is coming next

- What we've been up to with PAC-Bayes recently!

Part II

News from the PAC-Bayes frontline

- Alquier and Guedj (2018). Simpler PAC-Bayesian bounds for hostile data, Machine Learning.
- Mhammedi, Grünwald and Guedj (2019). PAC-Bayes Un-Expected Bernstein Inequality, NeurIPS 2019.
- Letarte, Germain, Guedj and Laviolette (2019). Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks, NeurIPS 2019.
- Nozawa, Germain and Guedj (2020). PAC-Bayesian contrastive unsupervised representation learning, UAI 2020.
- Haddouche, Guedj, Rivasplata and Shawe-Taylor (2020). PAC-Bayes unleashed: generalisation bounds with unbounded losses, preprint.
- Cantelobre, Guedj, Maria-Ortiz and Shawe-Taylor (2020). A PAC-Bayesian Perspective on Structured Prediction with Implicit Loss Embeddings, preprint.
- Mhammedi, Guedj and Williamson (2020). PAC-Bayesian Bound for the Conditional Value at Risk, NeurIPS 2020 (spotlight).













Learning with non-iid or heavy-tailed data

Alquier and Guedj (2018)

No iid or bounded loss assumptions. For any integer q,

$$\mathcal{M}_q := \int \mathbb{E}\left(|R_{\text{in}}(h) - R_{\text{out}}(h)|^q\right) \mathrm{d}P(h).$$

Csiszár f-divergence: let f be a convex function with $f(1) = 0$,

$$D_f(Q, P) = \int f\left(\frac{\mathrm{d}Q}{\mathrm{d}P}\right) \mathrm{d}P$$

when $Q \ll P$ and $D_f(Q, P) = +\infty$ otherwise.

The KL is given by the special case $\mathrm{KL}(Q\|P) = D_{x\log(x)}(Q, P)$.

PAC-Bayes with f-divergences

Fix $p > 1$, $q = \frac{p}{p-1}$, $\delta \in (0, 1)$ and let $\phi_p \colon x \mapsto x^p$. With probability at least $1 - \delta$ we have for any distribution Q

$$|R_{\text{out}}(Q) - R_{\text{in}}(Q)| \leqslant \left(\frac{\mathcal{M}_q}{\delta}\right)^{\frac{1}{q}} \left(D_{\phi_p - 1}(Q, P) + 1\right)^{\frac{1}{p}}.$$

The bound decouples

- the moment $\mathcal{M}_q$ (which depends on the distribution of the data)
- and the divergence $D_{\phi_p - 1}(Q, P)$ (measure of complexity).

Corolloray: with probability at least $1 - \delta$, for any Q,

$$R_{\text{out}}(Q) \leqslant R_{\text{in}}(Q) + \left(\frac{\mathcal{M}_q}{\delta}\right)^{\frac{1}{q}} \left(D_{\phi_p - 1}(Q, P) + 1\right)^{\frac{1}{p}}.$$

Again, strong incitement to define the ”optimal” posterior as the minimizer of the right-hand side!

For $p = q = 2$, w.p. $\geqslant 1 - \delta$, $R_{\text{out}}(Q) \leqslant R_{\text{in}}(Q) + \sqrt{\frac{\mathcal{V}}{m\delta} \int \left(\frac{\mathrm{d}Q}{\mathrm{d}P}\right)^2 \mathrm{d}P}$.

Proof

Let $\Delta(h) := |R_{\text{in}}(h) - R_{\text{out}}(h)|$.

$$\left| \int R_{\text{out}} \mathrm{d}Q - \int R_{\text{in}} \mathrm{d}Q \right|$$

Jensen

$$\leqslant \int \Delta \mathrm{d}Q$$

Change of measure

$$= \int \Delta \frac{\mathrm{d}Q}{\mathrm{d}P} \mathrm{d}P$$

Hölder

$$\leqslant \left(\int \Delta^q \mathrm{d}P \right)^{\frac{1}{q}} \left(\int \left(\frac{\mathrm{d}Q}{\mathrm{d}P} \right)^p \mathrm{d}P \right)^{\frac{1}{p}}$$

Markov

$$\underset{1-\delta}{\leqslant} \left(\frac{\mathbb{E} \int \Delta^q \mathrm{d}P}{\delta} \right)^{\frac{1}{q}} \left(\int \left(\frac{\mathrm{d}Q}{\mathrm{d}P} \right)^p \mathrm{d}P \right)^{\frac{1}{p}}$$

$$= \left(\frac{\mathcal{M}_q}{\delta} \right)^{\frac{1}{q}} \left(D_{\phi_p - 1}(Q, P) + 1 \right)^{\frac{1}{p}}.$$

Binary Activated Neural Networks

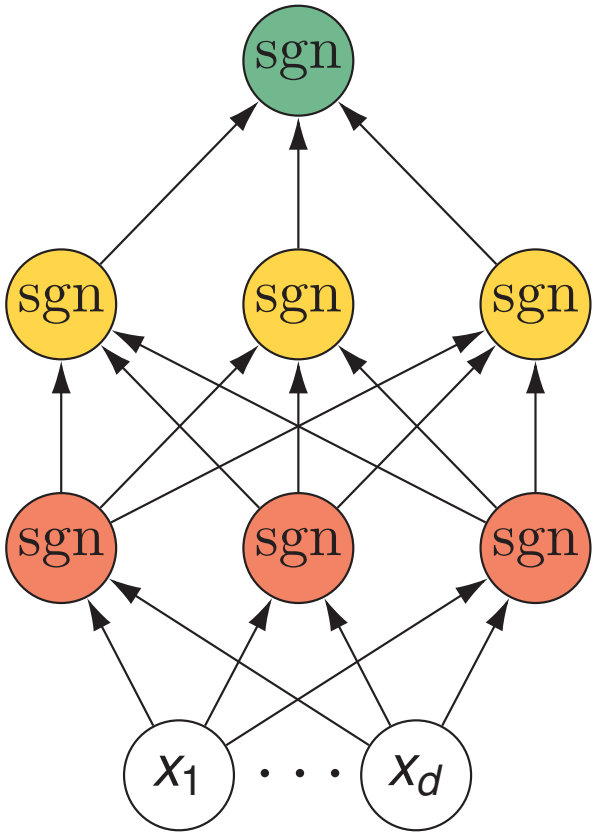
- $\mathbf{x} \in \mathbb{R}^{d_0}$
- $y \in \{-1, 1\}$

Architecture:

- L *fully connected* layers
- d_k denotes the number of neurons of the k^{th} layer
- $\operatorname{sgn}(a) = 1$ if $a > 0$ and $\operatorname{sgn}(a) = -1$ otherwise

Parameters:

- $\mathbf{W}_k \in \mathbb{R}^{d_k \times d_{k-1}}$ denotes the weight matrices.
- $\theta = \operatorname{vec}\left(\{\mathbf{W}_k\}_{k=1}^{L}\right) \in \mathbb{R}^{D}$



Prediction

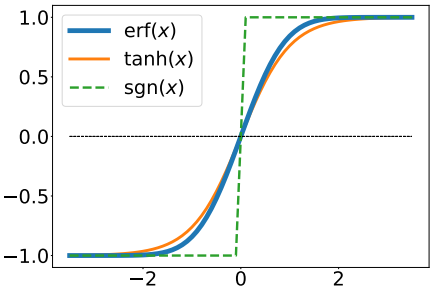
$$f_\theta(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w}_L \operatorname{sgn}\left(\mathbf{W}_{L-1} \operatorname{sgn}\left(\ldots \operatorname{sgn}\left(\mathbf{W}_1 \mathbf{x}\right)\right)\right)\right),$$

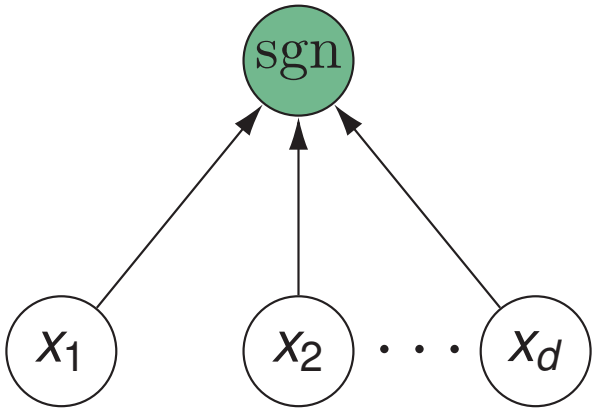
Building block: one layer (aka linear predictor)

Letarte et al. (2019)

Model $f_{\mathbf{w}}(\mathbf{x}) \stackrel{\text{def}}{=} \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x})$, with $\mathbf{w} \in \mathbb{R}^d$.

- Linear classifiers $\mathcal{F}_d \stackrel{\text{def}}{=} \{f_{\mathbf{v}} | \mathbf{v} \in \mathbb{R}^d\}$
- Predictor $F_{\mathbf{w}}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{E}_{\mathbf{v} \sim Q_{\mathbf{w}}} f_{\mathbf{v}}(\mathbf{x}) = \operatorname{erf}\left(\frac{\mathbf{w} \cdot \mathbf{x}}{\sqrt{d}\|\mathbf{x}\|}\right)$
- Sampling + closed form of the KL + a few other tricks + extension to an arbitrary number of layers





Generalisation bound

Let F_θ denote the network with parameter θ. With probability at least $1-\delta$, for any $\theta \in \mathbb{R}^D$

$$R_{\text{out}}(F_\theta) \leqslant \inf_{C>0} \left\{ \frac{1}{1-e^{-C}} \left(1 - \exp\left(-C R_{\text{in}}(F_\theta) - \frac{\mathrm{KL}(\theta, \theta_0) + \log \frac{2\sqrt{m}}{\delta}}{m} \right) \right) \right\}.$$

Numerical experiments

Model name	Cost function	Train split	Valid split	Model selection	Prior
MLP–tanh	linear loss, L2 regularized	80%	20%	valid linear loss	-
PBGNet$_\ell$	linear loss, L2 regularized	80%	20%	valid linear loss	random init
PBGNet	**PAC-Bayes bound**	**100 %**	**-**	**PAC-Bayes bound**	**random init**
PBGNet$_{\text{pre}}$					
– pretrain	linear loss (20 epochs)	50%	-	-	random init
– final	PAC-Bayes bound	50%	-	PAC-Bayes bound	pretrain

Dataset	MLP–tanh		PBGNet$_\ell$		PBGNet			PBGNet$_{\text{pre}}$		
	R_{in}	R_{out}	R_{in}	R_{out}	R_{in}	R_{out}	Bound	R_{in}	R_{out}	Bound
ads	0.021	0.037	0.018	**0.032**	0.024	0.038	0.283	0.034	0.033	0.058
adult	0.128	0.149	0.136	**0.148**	0.158	0.154	0.227	0.153	0.151	0.165
mnist17	0.003	**0.004**	0.008	0.005	0.007	0.009	0.067	0.003	0.005	0.009
mnist49	0.002	**0.013**	0.003	0.018	0.034	0.039	0.153	0.018	0.021	0.030
mnist56	0.002	0.009	0.002	0.009	0.022	0.026	0.103	0.008	**0.008**	0.017
mnistLH	0.004	**0.017**	0.005	0.019	0.071	0.073	0.186	0.026	0.026	0.033

An attempt at summarising my research

Quest for generalisation guarantees (about half *via* PAC-Bayes)

Directions:

- Generic bounds (relaxing assumptions such as iid or boundedness, new concentration inequalities, . . .)
- Tight bounds for specific algorithms (deep neural networks, NMF, . . .)
- Towards new measures of performance (CVaR, ranking, contrastive losses, . . .)
- Coupling theory and implemented algorithms: bound-driven algorithms
- Applications (providing guidelines to machine learning users, sustainable / frugal machine learning)

Thanks!

What this talk could have been about...

- Tighter PAC-Bayes bounds (Mhammedi et al., 2019)
- PAC-Bayes for conditional value at risk (Mhammedi et al., 2020)
- PAC-Bayes-driven deep neural networks (Biggs and Guedj, 2020)
- PAC-Bayes and robust learning (Guedj and Pujol, 2019)
- PAC-Bayes for unbounded losses (Haddouche et al., 2020a)
- PAC-Bayesian online clustering (Li et al., 2018)
- PAC-Bayesian bipartite ranking (Guedj and Robbiano, 2018)
- Online k-means clustering (Cohen-Addad et al., 2021)
- Sequential learning of principal curves (Guedj and Li, 2018)
- PAC-Bayes for heavy-tailed, dependent data (Alquier and Guedj, 2018)
- Stability and generalisation (Celisse and Guedj, 2016)
- Additive regression (Guedj and Alquier, 2013)
- Contrastive unsupervised learning (Nozawa et al., 2020)
- Generalisation bounds for structured prediction (Cantelobre et al., 2020)
- Image denoising (Guedj and Rengot, 2020)
- Matrix factorisation (Alquier and Guedj, 2017; Chrétien and Guedj, 2020)
- Preventing model overfitting (Zhang et al., 2019)
- Decentralised learning with aggregation (Klein et al., 2019)
- Ensemble learning and nonlinear aggregation (Biau et al., 2016) in Python (Guedj and Srinivasa Desikan, 2018, 2020)
- Identifying subcommunities in social networks and application to forecasting elections (Vendeville et al., 2021, 2020)
- Upper and lower bounds for kernel PCA (Haddouche et al., 2020b)
- Prediction with multi-task Gaussian processes (Leroy et al., 2020b,a)

+ a few more in the pipe, soon on arXiv

References I

P. Alquier and B. Guedj. An oracle inequality for quasi-Bayesian nonnegative matrix factorization. *Mathematical Methods of Statistics*, 26(1):55–67, 2017.

P. Alquier and B. Guedj. Simpler PAC-Bayesian bounds for hostile data. *Machine Learning*, 107(5):887–902, 2018.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1903070116. URL https://www.pnas.org/content/116/32/15849.

G. Biau, A. Fischer, B. Guedj, and J. D. Malley. Cobra: A combined regression strategy. *Journal of Multivariate Analysis*, 146:18–28, 2016. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2015.04.007. URL http://www.sciencedirect.com/science/article/pii/S0047259X15000950. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces.

F. Biggs and B. Guedj. Differentiable pac-bayes objectives with partially aggregated neural networks. Submitted., 2020. URL https://arxiv.org/abs/2006.12228.

T. Cantelobre, B. Guedj, M. Pérez-Ortiz, and J. Shawe-Taylor. A pac-bayesian perspective on structured prediction with implicit loss embeddings. Submitted., 2020. URL https://arxiv.org/abs/2012.03780.

O. Catoni. *Statistical Learning Theory and Stochastic Optimization*. École d'Été de Probabilités de Saint-Flour 2001. Springer, 2004.

A. Celisse and B. Guedj. Stability revisited: new generalisation bounds for the leave-one-out. *arXiv preprint arXiv:1608.06412*, 2016.

S. Chrétien and B. Guedj. Revisiting clustering as matrix factorisation on the Stiefel manifold. In *LOD - The Sixth International Conference on Machine Learning, Optimization, and Data Science*, 2020. URL https://arxiv.org/abs/1903.04479.

V. Cohen-Addad, B. Guedj, V. Kanade, and G. Rom. Online k-means clustering. In *AISTATS*, 2021. URL https://arxiv.org/abs/1909.06861.

I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3:146–158, 1975.

M. D. Donsker and S. S. Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. *Communications on Pure and Applied Mathematics*, 28, 1975.

B. Guedj. A Primer on PAC-Bayesian Learning. In *Proceedings of the second congress of the French Mathematical Society*, 2019. URL https://arxiv.org/abs/1901.05353.

B. Guedj and P. Alquier. PAC-Bayesian estimation and prediction in sparse additive models. *Electron. J. Statist.*, 7:264–291, 2013.

References II

B. Guedj and L. Li. Sequential learning of principal curves: Summarizing data streams on the fly. *arXiv preprint arXiv:1805.07418*, 2018.

B. Guedj and L. Pujol. Still no free lunches: the price to pay for tighter PAC-Bayes bounds. *arXiv preprint arXiv:1910.04460*, 2019.

B. Guedj and J. Rengot. Non-linear aggregation of filters to improve image denoising. In *Computing Conference*, 2020. URL `https://arxiv.org/abs/1904.00865`.

B. Guedj and S. Robbiano. PAC-Bayesian high dimensional bipartite ranking. *Journal of Statistical Planning and Inference*, 196:70 – 86, 2018. ISSN 0378-3758.

B. Guedj and B. Srinivasa Desikan. Pycobra: A python toolbox for ensemble learning and visualisation. *Journal of Machine Learning Research*, 18(190):1–5, 2018. URL `http://jmlr.org/papers/v18/17-228.html`.

B. Guedj and B. Srinivasa Desikan. Kernel-based ensemble learning in python. *Information*, 11(2):63, Jan 2020. ISSN 2078-2489. doi: $10.3390/\mathrm{info}11020063$. URL `http://dx.doi.org/10.3390/info11020063`.

M. Haddouche, B. Guedj, O. Rivasplata, and J. Shawe-Taylor. PAC-Bayes unleashed: generalisation bounds with unbounded losses. Submitted., 2020a. URL `https://arxiv.org/abs/2006.07279`.

M. Haddouche, B. Guedj, O. Rivasplata, and J. Shawe-Taylor. Upper and Lower Bounds on the Performance of Kernel PCA. Submitted., 2020b. URL `https://arxiv.org/abs/2012.10369`.

J. Klein, M. Albardan, B. Guedj, and O. Colot. Decentralized learning with budgeted network load using gaussian copulas and classifier ensembles. In *ECML-PKDD, Decentralised Machine Learning at the Edge workshop*, 2019. arXiv:1804.10028.

A. Leroy, P. Latouche, B. Guedj, and S. Gey. Cluster-specific predictions with multi-task gaussian processes. Submitted., 2020a. URL `https://arxiv.org/abs/2011.07866`.

A. Leroy, P. Latouche, B. Guedj, and S. Gey. Magma: Inference and prediction with multi-task gaussian processes. Submitted., 2020b. URL `https://arxiv.org/abs/2007.10731`.

G. Letarte, P. Germain, B. Guedj, and F. Laviolette. Dichotomize and Generalize: PAC-Bayesian Binary Activated Deep Neural Networks. *arXiv:1905.10259*, 2019. To appear at NeurIPS.

L. Li, B. Guedj, and S. Loustau. A quasi-Bayesian perspective to online clustering. *Electron. J. Statist.*, 12(2):3071–3113, 2018.

References III

D. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the International Conference on Computational Learning Theory (COLT)*, 1998.

D. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37, 1999.

Z. Mhammedi, P. D. Grunwald, and B. Guedj. PAC-Bayes Un-Expected Bernstein Inequality. In *NeurIPS 2019*, 2019.

Z. Mhammedi, B. Guedj, and R. C. Williamson. PAC-Bayesian Bound for the Conditional Value at Risk. In *NeurIPS 2020*, 2020. URL `https://arxiv.org/abs/2006.14763`.

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

K. Nozawa, P. Germain, and B. Guedj. PAC-Bayesian contrastive unsupervised representation learning. In *UAI*, 2020. URL `https://arxiv.org/abs/1910.04464`.

J. Shawe-Taylor and R. C. Williamson. A PAC analysis of a Bayes estimator. In *Proceedings of the 10th annual conference on Computational Learning Theory*, pages 2–9. ACM, 1997. doi: $10.1145/267460.267466$.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

A. Vendeville, B. Guedj, and S. Zhou. Voter model with stubborn agents on strongly connected social networks. Submitted., 2020. URL `https://arxiv.org/abs/2006.07265`.

A. Vendeville, B. Guedj, and S. Zhou. Forecasting elections results via the voter model with stubborn nodes. *Applied Network Science*, 6, 2021. doi: $10.1007/s41109\text{-}020\text{-}00342\text{-}7$. URL `https://arxiv.org/abs/2009.10627`.

J. M. Zhang, M. Harman, B. Guedj, E. T. Barr, and J. Shawe-Taylor. Perturbation validation: A new heuristic to validate machine learning models. *arXiv preprint arXiv:1905.10201*, 2019.