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

2nde Journée MAS-MODE, SMAI
March 7, 2022
In a nutshell

Research at the crossroads of statistics, probability, machine learning, optimisation. "Mathematical foundations of machine learning" says it all!

Statistical learning theory, PAC-Bayes, computational statistics, theoretical analysis of deep learning and representation learning...
In a nutshell

Research at the crossroads of statistics, probability, machine learning, optimisation. "Mathematical foundations of machine learning" says it all!

Statistical learning theory, PAC-Bayes, computational statistics, theoretical analysis of deep learning and representation learning...

Personal obsession: **generalisation**.
Learning is to be able to generalise

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

Memorising the already seen data is usually bad → 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)
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 and significant environmental footprint.
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 and significant environmental footprint.
A tale of two learners

Second contender: the next iteration
A tale of two learners

Second contender: the next iteration

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.) (if you’re interested, get in touch!)
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.)

(if you’re interested, get in touch!)
Part I

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

The simplest setting

**Learning algorithm** \( A : \mathcal{Z}^m \rightarrow \mathcal{H} \)

- \( \mathcal{Z} = \mathcal{X} \times \mathcal{Y} \)
- \( \mathcal{H} = \) 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 \( \mathcal{P} \) over \( \mathcal{Z} \).
- Learner doesn’t know \( \mathcal{P} \), only sees the training set.
- The training set examples are \textit{i.i.d.} from \( \mathcal{P} : S_m \sim \mathcal{P}^m \)
Statistical Learning Theory is about high confidence
For a fixed algorithm, function class and sample size, generating random samples $\rightarrow$ distribution of test errors
Statistical Learning Theory is about high confidence
For a fixed algorithm, function class and sample size, generating random samples \( \rightarrow \) distribution of test errors

■ Focusing on the mean of the error distribution?
  ▶ can be misleading: learner only has one sample
Statistical Learning Theory is about high confidence

For a fixed algorithm, function class and sample size, generating random samples $\rightarrow$ 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 $m$
Statistical Learning Theory is about high confidence

For a fixed algorithm, function class and sample size, generating random samples → 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%
Statistical Learning Theory is about high confidence
For a fixed algorithm, function class and sample size, generating random samples → 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’ $\delta$: $\mathbb{P}^m[\text{large error}] \leq \delta$
  $\delta$ is the probability of being misled by the training set
Statistical Learning Theory is about high confidence
For a fixed algorithm, function class and sample size, generating random samples $\rightarrow$ 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’ $\delta$: $\mathbb{P}^m[\text{large error}] \leq \delta$
    - $\delta$ is the probability of being misled by the training set

- Hence **high confidence**: $\mathbb{P}^m[\text{approximately correct}] \geq 1 - \delta$
What to achieve from the sample?

Using the available sample to:

1. Learn a predictor
   - Algorithm driven by some learning principle
   - Informed by prior knowledge resulting in inductive bias
2. Certify the predictor's performance
   - What happens beyond the training set
   - Generalisation bounds

Actually these two goals interact with each other!
What to achieve from the sample?

Use the available sample to:

1. learn a predictor
2. certify the predictor’s performance
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
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
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:** $R_{\text{out}}(h) = \mathbb{E}[\ell(h(X), Y)]$
(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 $\Delta(h) \leq \epsilon(m, \delta)$

$\Rightarrow \quad R_{\text{out}}(h) \leq R_{\text{in}}(h) + \epsilon(m, \delta)$

**Flavours:**
- distribution-free
- algorithm-free
- distribution-dependent
- 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 $\delta > 0$,

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

Think of $\varepsilon(m, \delta)$ as 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.
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/


NeurIPS 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 $\geq 1 - \delta$, $R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{\delta} \right)}$. 

Finite function class $H$ (worst-case approach):

with probability $\geq 1 - \delta$, $\forall h \in H$, $R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log (|H| \delta)}$.

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

with probability $\geq 1 - \delta$, $\forall h_i \in H$, $R_{\text{out}}(h_i) \leq R_{\text{in}}(h_i) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{p_i \delta} \right)}$.

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.

$\rightarrow$ Extension: PAC-Bayes allows to consider distributions over hypotheses.
Before PAC-Bayes

- Single hypothesis $h$ (building block):
  
  with probability $\geq 1 - \delta$, \( R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{\delta} \right)} \).

- Finite function class $\mathcal{H}$ (worst-case approach):
  
  w.p. $\geq 1 - \delta$, \( \forall h \in \mathcal{H}, \ R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{\delta} \right)} \).

- Structural risk minimisation: data-dependent hypotheses $h_i$ associated with prior weight $p_i$
  
  w.p. $\geq 1 - \delta$, \( \forall h_i \in \mathcal{H}, \ R_{\text{out}}(h_i) \leq R_{\text{in}}(h_i) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{p_i \delta} \right)} \).

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

- Single hypothesis $h$ (building block):
  
  with probability $\geq 1 - \delta$, \( R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{\delta} \right)} \).

- Finite function class $\mathcal{H}$ (worst-case approach):
  
  w.p. $\geq 1 - \delta$, $\forall h \in \mathcal{H}$, \( R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log \left( \frac{\left| \mathcal{H} \right|}{\delta} \right)} \).

- Structural risk minimisation: data-dependent hypotheses $h_i$ associated with prior weight $p_i$
  
  w.p. $\geq 1 - \delta$, $\forall h_i \in \mathcal{H}$, \( R_{\text{out}}(h_i) \leq R_{\text{in}}(h_i) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{p_i \delta} \right)} \).

- 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.
Before PAC-Bayes

- **Single hypothesis** $h$ (building block):
  
  with probability $\geq 1 - \delta$, $R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{\delta} \right)}$.

- **Finite function class** $\mathcal{H}$ (worst-case approach):
  
  w.p. $\geq 1 - \delta$, $\forall h \in \mathcal{H}$, $R_{\text{out}}(h) \leq R_{\text{in}}(h) + \sqrt{\frac{1}{2m} \log \left( \frac{|\mathcal{H}|}{\delta} \right)}$

- **Structural risk minimisation**: data-dependent hypotheses $h_i$ associated with prior weight $p_i$
  
  w.p. $\geq 1 - \delta$, $\forall h_i \in \mathcal{H}$, $R_{\text{out}}(h_i) \leq R_{\text{in}}(h_i) + \sqrt{\frac{1}{2m} \log \left( \frac{1}{p_i \delta} \right)}$

- **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.

→ **Extension**: PAC-Bayes allows to consider *distributions* over hypotheses.
"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:

\[
R_{\text{in}}(Q) \equiv \int_{\mathcal{H}} R_{\text{in}}(h) \, dQ(h) \quad \text{and} \quad R_{\text{out}}(Q) \equiv \int_{\mathcal{H}} R_{\text{out}}(h) \, dQ(h)
\]

Kullback-Leibler divergence $\text{KL}(Q||P) = E_{h \sim Q} \ln \frac{Q(h)}{P(h)}$.

- **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

$$
\mathbb{P}_m \left( \forall Q \text{ on } \mathcal{H}: \ R_{\text{out}}(Q) \leq R_{\text{in}}(Q) + \sqrt{\frac{\text{KL}(Q\|P) + \ln \frac{2\sqrt{m}}{\delta}}{2m}} \right) \geq 1 - \delta,
$$
PAC-Bayes-driven learning algorithms

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

\[
\text{Error on unseen data} \leq \text{Error on sample} + \text{complexity term} \quad R_{\text{out}}(Q) \leq R_{\text{in}}(Q) + F(Q, \cdot)
\]

This defines a principled strategy to obtain new learning algorithms:

\[
h \sim Q^* \quad Q^* \in \arg \inf_{Q \ll P} \left\{ R_{\text{in}}(Q) + F(Q, \cdot) \right\}
\]

(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 $\text{KL}$-divergence (Csiszár, 1975; Donsker and Varadhan, 1975; Catoni, 2004).
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 \(\phi : A \rightarrow \mathbb{R}\) such that \(\int (\exp \circ \phi) \, dP < \infty\),

\[
\log \int (\exp \circ \phi) \, dP = \sup_{Q \ll P} \left\{ \int \phi \, dQ - \text{KL}(Q, P) \right\}.
\]

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

\[
\frac{dG}{dP}(a) = \frac{\exp \circ \phi(a)}{\int (\exp \circ \phi) \, dP}, \quad a \in A.
\]
\[
\log \int (\exp \circ \phi) dP = \sup_{Q \ll P} \left\{ \int \phi dQ - \text{KL}(Q, P) \right\}, \quad \frac{dG}{dP} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) dP}.
\]

Proof: let \(Q \ll P\).
\[
\log \int (\exp \circ \phi) dP = \sup_{Q \ll P} \left\{ \int \phi dQ - \text{KL}(Q, P) \right\}, \quad \frac{dG}{dP} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) dP}.
\]

Proof: let \( Q \ll P \).

\[-\text{KL}(Q, G) = - \int \log \left( \frac{dQ}{dP} \frac{dP}{dG} \right) dQ \]
\[ \log \int (\exp \circ \phi) dP = \sup_{Q \ll P} \{ \int \phi dQ - \text{KL}(Q, P) \}, \quad \frac{dG}{dP} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) dP}. \]

Proof: let \( Q \ll P \).

\[-\text{KL}(Q, G) = -\int \log \left( \frac{dQ}{dP} \frac{dP}{dG} \right) dQ = -\int \log \left( \frac{dQ}{dP} \right) dQ + \int \log \left( \frac{dG}{dP} \right) dQ \]
\[
\log \int (\exp \circ \phi) dP = \sup_{Q \ll P} \{ \int \phi dQ - \KL(Q, P) \}, \quad \frac{dG}{dP} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) dP}.
\]

Proof: let \( Q \ll P \).

\[
- \KL(Q, G) = - \int \log \left( \frac{dQ}{dP} \frac{dP}{dG} \right) dQ
\]

\[
= - \int \log \left( \frac{dQ}{dP} \right) dQ + \int \log \left( \frac{dG}{dP} \right) dQ
\]

\[
= - \KL(Q, P) + \int \phi dQ - \log \int (\exp \circ \phi) dP.
\]
\[
\log \int (\exp \circ \phi) dP = \sup_{Q \ll P} \left\{ \int \phi dQ - \text{KL}(Q, P) \right\}, \quad \frac{dG}{dP} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) dP}.
\]

Proof: let \( Q \ll P \).

\[
- \text{KL}(Q, G) = - \int \log \left( \frac{dQ}{dP} \frac{dP}{dG} \right) dQ
= - \int \log \left( \frac{dQ}{dP} \right) dQ + \int \log \left( \frac{dG}{dP} \right) dQ
= -\text{KL}(Q, P) + \int \phi dQ - \log \int (\exp \circ \phi) dP.
\]

\( \text{KL}(\cdot, \cdot) \) is non-negative, \( Q \mapsto -\text{KL}(Q, G) \) reaches its max. in \( Q = G \):
\[
\log \int (\exp \circ \phi) dP = \sup_{Q \ll P} \left\{ \int \phi dQ - KL(Q, P) \right\}, \quad \frac{dG}{dP} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) dP}.
\]

Proof: let \( Q \ll P \).

\[
- KL(Q, G) = - \int \log \left( \frac{dQ}{dP} \frac{dP}{dG} \right) dQ
\]
\[
= - \int \log \left( \frac{dQ}{dP} \right) dQ + \int \log \left( \frac{dG}{dP} \right) dQ
\]
\[
= -KL(Q, P) + \int \phi dQ - \log \int (\exp \circ \phi) dP.
\]

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

\[
0 = \sup_{Q \ll P} \left\{ \int \phi dQ - KL(Q, P) \right\} - \log \int (\exp \circ \phi) dP.
\]
\[
\log \int \left( \exp \circ \phi \right) dP = \sup_{Q \ll P} \left\{ \int \phi dQ - KL(Q, P) \right\}, \quad \frac{dG}{dP} = \frac{\exp \circ \phi}{\int (\exp \circ \phi) dP}.
\]

Proof: let \( Q \ll P \).

\[
-\text{KL}(Q, G) = -\int \log \left( \frac{dQ}{dP} \frac{dP}{dG} \right) dQ
= -\int \log \left( \frac{dQ}{dP} \right) dQ + \int \log \left( \frac{dG}{dP} \right) dQ
= -\text{KL}(Q, P) + \int \phi dQ - \log \int \left( \exp \circ \phi \right) dP.
\]

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

\[
0 = \sup_{Q \ll P} \left\{ \int \phi dQ - \text{KL}(Q, P) \right\} - \log \int \left( \exp \circ \phi \right) dP.
\]

Let \( \lambda > 0 \) and take \( \phi = -\lambda R_{\text{in}} \),

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

What we’ve seen so far
Recap

What we’ve seen so far

- Statistical learning theory is about high confidence control of generalisation
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...
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
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 (and in particular deep learning)
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** (and in particular **deep learning**)

What is coming next
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 (and in particular deep learning)

What is coming next

- What we’ve been up to with PAC-Bayes recently!
Part II

News from the PAC-Bayes frontline

Some of my partners in crime
On Margins and Derandomisation in PAC-Bayes

We provide a unified framework for derandomising PAC-Bayes bounds with margins, leading to new bounds or greatly simplified proofs for

- $L_2$ and $L_1$ normed linear predictors,
- Linear predictors with a learned randomised feature space,
- One-hidden-layer neural networks with erf activations,
- Deep ReLU networks.

Key idea: PAC-Bayes bounds are (mostly) SOTA, but apply for non-deterministic randomised predictions. Large margin deterministic predictors give similar predictive performance to their randomised counterparts.
SHEL: An unusual neural architecture

Binary $y = \{\pm 1\}$ or multiclass classification $y = \{1, \ldots, c\}$. Predictors $f$ are score-valued: $f(x) \in \mathbb{R}^c$ (multiclass) or $f(x) \in \mathbb{R}$ (binary). We define the binary margin $M_{\text{bin}}(f, (x, y)) = yf(x)$ and multiclass margin $M_{\text{multi}}(f, (x, y)) = f(x)[y] - \max_{k \neq y} f(x)[k]$.

$R_{\text{out}}(f) = \Pr\{(x, y) : M(f, (x, y)) \leq 0\}$,

$R_{\text{in}, \gamma}(f) = m^{-1} \left| \{(x, y) \in S : M(f, (x, y)) \leq \gamma\} \right|$.

**SHEL network**: elementwise error function activations $F_{U,V}(x) = \text{Verf}(Ux)$.

**Theorem.** For SHEL network with $K$ hidden units,

$$R_{\text{out}}(F_{U,V}) \leq R_{\text{in}, \gamma}(F_{U,V}) + \tilde{O} \left( \frac{\sqrt{K}}{\gamma \sqrt{m}} (\|V\|_{\text{max}} \|U - U^0\|_F + \|V\|_F) \right).$$
On PAC-Bayesian reconstruction guarantees for VAEs

In 3 weeks at AISTATS 2022

x → encoder → Latent representation → decoder → \( \hat{x} = d(e(x)) \)

[Credits: Danijar Hafner]
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 self-certifying specific algorithms (deep neural networks, NMF, ...)
- Towards new measures of performance (CVaR, ranking, contrastive losses, ...)
- Coupling theory and implemented algorithms: bound-driven algorithms
- Impact beyond learning theory (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, 2021a,b; Perez-Ortiz et al., 2021a,b)
- PAC-Bayes and robust learning (Guedj and Pujol, 2021)
- PAC-Bayes for unbounded losses (Haddouche et al., 2021)
- 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, 2021)
- PAC-Bayes for heavy-tailed, dependent data (Alquier and Guedj, 2018)
- Stability and generalisation (Celisse and Guedj, 2016)
- Additive regression (Guedj and Alquier, 2013)
- Stochastic majority votes (Zantedeschi et al., 2021)

- Contrastive unsupervised learning (Nozawa et al., 2020)
- Generalisation bounds for structured prediction (Cantelobre et al., 2020)
- MMD aggregated two sample tests (Schrab et al., 2021)
- 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., 2020)
- 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., 2021a,b)
- Upper and lower bounds for kernel PCA (Haddouche et al., 2020)
- Prediction with multi-task Gaussian processes (Leroy et al., 2020b,a)

+ a few more in the pipe, soon on arXiv

[https://bguedj.github.io](https://bguedj.github.io)
References


References III


