A (condensed) primer on PAC-Bayesian Learning
followed by
News from the PAC-Bayes frontline

Benjamin Guedj
https://bguedj.github.io

Research scientist, Inria
Principal research fellow, University College London
Scientific director, The Inria London Programme
Visiting researcher, The Alan Turing Institute

UCL Statistical Science Seminar
January 21, 2021

Greetings!

Greetings!

PhD in mathematics and statistics, 2013 Sorbonne Univ. (France)

Since 2018, principal research fellow at CS and the AI Centre

Interests: statistical learning theory, PAC-Bayes, computational statistics, generalisation bounds for deep learning, and many others.

The Inria London Programme: please get in touch if interested!

Greetings!

PhD in mathematics and statistics, 2013 Sorbonne Univ. (France)

Since 2018, principal research fellow at CS and the AI Centre

Interests: statistical learning theory, PAC-Bayes, computational statistics, generalisation bounds for deep learning, and many others.

The Inria London Programme: please get in touch if interested!

Becoming quite an expert in coupling statistical learning and sleep deprivation.

What to expect

What to expect

I will...

- Provide an overview of what PAC-Bayes is

What to expect

I will...

- Provide an overview of what PAC-Bayes is
- Illustrate its flexibility and relevance to tackle modern machine learning tasks, and rethink generalisation

What to expect

I will...

- Provide an overview of what PAC-Bayes is
- Illustrate its flexibility and relevance to tackle modern machine learning tasks, and rethink generalisation
- Cover key ideas and a few results

What to expect

I will...

- Provide an overview of what PAC-Bayes is
- Illustrate its flexibility and relevance to tackle modern machine learning tasks, and rethink generalisation
- Cover key ideas and a few results
- Briefly present a sample of recent contributions from my group

What to expect

I will...

- Provide an overview of what PAC-Bayes is
- Illustrate its flexibility and relevance to tackle modern machine learning tasks, and rethink generalisation
- Cover key ideas and a few results
- Briefly present a sample of recent contributions from my group

I won't...

What to expect

I will...

- Provide an overview of what PAC-Bayes is
- Illustrate its flexibility and relevance to tackle modern machine learning tasks, and rethink generalisation
- Cover key ideas and a few results
- Briefly present a sample of recent contributions from my group

I won't...

- Cover all of our ICML 2019 tutorial!
  See `https://bguedj.github.io/icml2019/index.html`
- Cover our NIPS 2017 workshop "(Almost) 50 Shades of Bayesian Learning: PAC-Bayesian trends and insights"
  See `https://bguedj.github.io/nips2017/`

Take-home message

Take-home message

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

Take-home message

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

PhD students, postdocs, tenured researchers, visiting positions
Through the Centre for AI at UCL,
and through the newly founded Inria London Programme

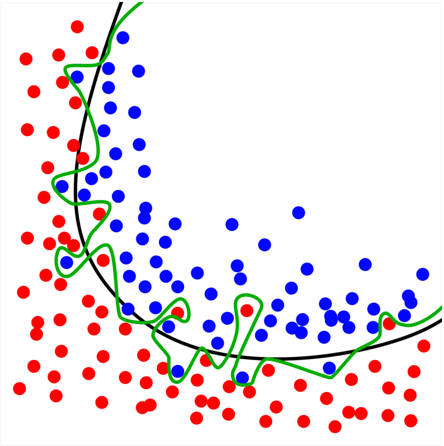
Part I

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

https://bguedj.github.io/icml2019/index.html
Survey in the Journal of the French Mathematical Society: *Guedj (2019)*

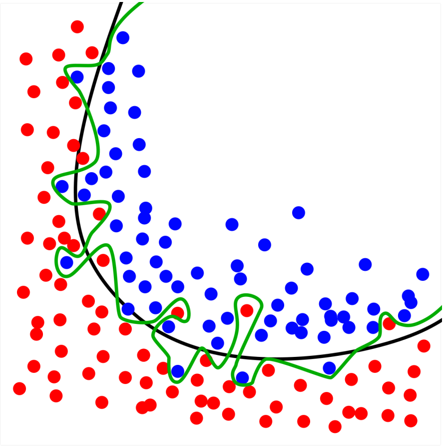
Learning is to be able to generalise

Learning is to be able to generalise



[Figure from Wikipedia]

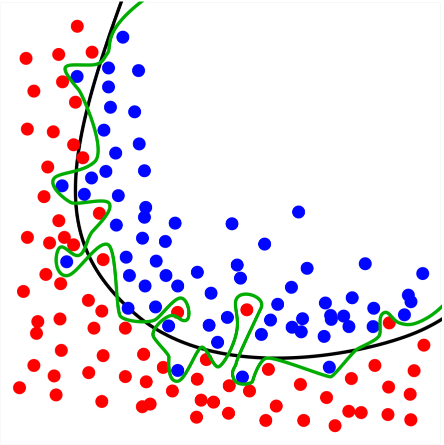
Learning is to be able to generalise



[Figure from Wikipedia]

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

Learning is to be able to generalise

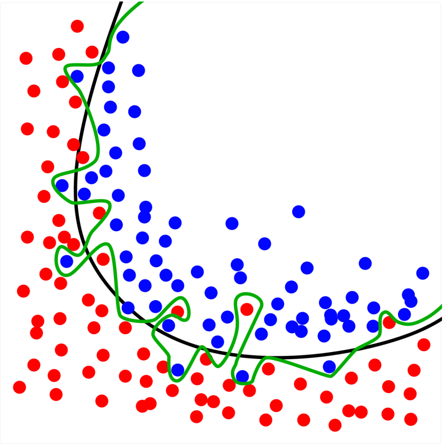


[Figure from Wikipedia]

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

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

Learning is to be able to generalise



[Figure from 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.

Statistical Learning Theory is about high confidence

Statistical Learning Theory is about high confidence

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

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 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

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%

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

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: $\mathbb{P}^m[\text{approximately correct}] \geqslant 1 - \delta$

Mathematical formalisation

Mathematical formalisation

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

- $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
  $\mathcal{X}$ = set of inputs
  $\mathcal{Y}$ = set of outputs (e.g. labels)

- $\mathcal{H}$ = hypothesis class
  = set of predictors
  (e.g. classifiers)
  functions $\mathcal{X} \to \mathcal{Y}$

Mathematical formalisation

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

- $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
  $\mathcal{X}$ = set of inputs
  $\mathcal{Y}$ = set of outputs (e.g. labels)
- $\mathcal{H}$ = hypothesis class
  = set of predictors
  (e.g. classifiers)
  functions $\mathcal{X} \to \mathcal{Y}$

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

Mathematical formalisation

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

- $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
  $\mathcal{X}$ = set of inputs
  $\mathcal{Y}$ = set of outputs (e.g. labels)

- $\mathcal{H}$ = hypothesis class
  = set of predictors
  (e.g. classifiers)
  functions $\mathcal{X} \to \mathcal{Y}$

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

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

What to achieve from the sample?

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!

Risk (aka error) measures

Risk (aka error) measures

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

Risk (aka error) measures

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

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

Risk (aka error) measures

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

Empirical risk: (in-sample)

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

Theoretical risk: (out-of-sample)

$$R_{\text{out}}(h) = \mathbb{E}\left[\ell(h(X), Y)\right]$$

Risk (aka error) measures

A loss function $\ell(h(X), Y)$ is used 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)

Examples:

- $\ell(h(X), Y) = \mathbf{1}[h(X) \neq Y]$: 0-1 loss (classification)
- $\ell(h(X), Y) = (Y - h(X))^2$: square loss (regression)
- $\ell(h(X), Y) = (1 - Yh(X))_+$: hinge loss
- $\ell(h(X), 1) = -\log(h(X))$: log loss (density estimation)
- ...

Generalisation

Generalisation

If predictor h does well on the in-sample (X, Y) pairs...

...will it still do well on out-of-sample pairs?

Generalisation

If predictor h does well on the in-sample (X, Y) pairs...

...will it still do well on out-of-sample pairs?

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

Generalisation

If predictor h does well on the in-sample (X, Y) pairs...

...will it still do well on out-of-sample pairs?

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

Upper bounds: w.h.p. $\Delta(h) \leqslant \epsilon(m, \delta)$

Generalisation

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: w.h.p. $\Delta(h) \leqslant \epsilon(m, \delta)$

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

Generalisation

If predictor h does well on the in-sample (X, Y) pairs...

...will it still do well on out-of-sample pairs?

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

Upper bounds: w.h.p. $\Delta(h) \leqslant \epsilon(m, \delta)$

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

Lower bounds: w.h.p. $\Delta(h) \geqslant \tilde{\epsilon}(m, \delta)$

Generalisation

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

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

Upper bounds: w.h.p. $\Delta(h) \leqslant \epsilon(m, \delta)$

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

Lower bounds: w.h.p. $\Delta(h) \geqslant \tilde{\epsilon}(m, \delta)$

Flavours:

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

Why you should care about generalisation bounds

Why you should care about generalisation bounds

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

Why you should care about generalisation bounds

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

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

Generalisation bounds:

Why you should care about generalisation bounds

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

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

Generalisation bounds:

- may be computed with the training sample only, do not depend on any test sample

Why you should care about generalisation bounds

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

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

Generalisation bounds:

- may be computed with the training sample only, do not depend on any test sample
- provide a computable control on the error on any unseen data with prespecified confidence

Why you should care about generalisation bounds

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

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

Generalisation bounds:

- may be computed with the training sample only, do not depend on any test sample
- provide a computable control on the error on any unseen data with prespecified confidence
- explain why specific learning algorithms actually work

Why you should care about generalisation bounds

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

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

Generalisation bounds:

- may be computed with the training sample only, do not depend on any test sample
- provide a computable control on the error on any unseen data with prespecified confidence
- explain why specific learning algorithms actually work
- and even lead to designing new algorithm which scale to more complex settings

Before PAC-Bayes

Before PAC-Bayes

- Single hypothesis h (building block):

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

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

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

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

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

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

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

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

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

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

- 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.

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

The PAC-Bayes framework

The PAC-Bayes framework

- Before data, fix a distribution $P \in M_1(\mathcal{H})$ ▷ ‘prior’

The PAC-Bayes framework

- Before data, fix a distribution $P \in M_1(\mathcal{H})$ $\triangleright$ ‘prior’
- Based on data, learn a distribution $Q \in M_1(\mathcal{H})$ $\triangleright$ ‘posterior’

The PAC-Bayes framework

- Before data, fix a distribution $P \in M_1(\mathcal{H})$ $\triangleright$ 'prior'
- Based on data, learn a distribution $Q \in M_1(\mathcal{H})$ $\triangleright$ 'posterior'
- Predictions:
  - draw $h \sim Q$ and predict with the chosen h.
  - each prediction with a fresh random draw.

The PAC-Bayes framework

- Before data, fix a distribution $P \in M_1(\mathcal{H})$ ▷ 'prior'
- Based on data, learn a distribution $Q \in M_1(\mathcal{H})$ ▷ 'posterior'
- Predictions:
  - draw $h \sim Q$ and predict with the chosen h.
  - each prediction with a fresh random draw.

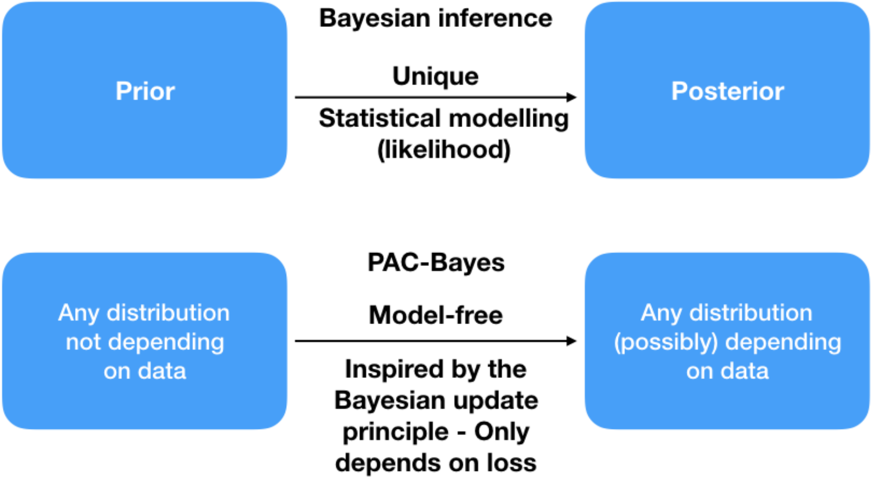
The risk measures $R_{\rm in}(h)$ and $R_{\rm out}(h)$ are extended by averaging:

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

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

PAC-Bayes aka Generalised Bayes

PAC-Bayes aka Generalised Bayes



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

"Posterior" is the twisted prior after confronting with data

PAC-Bayes bounds vs. Bayesian learning

PAC-Bayes bounds vs. Bayesian learning

- Prior

PAC-Bayes bounds vs. Bayesian learning

- Prior
  - PAC-Bayes: bounds hold for any distribution
  - Bayes: prior choice impacts inference

PAC-Bayes bounds vs. Bayesian learning

- Prior
  - PAC-Bayes: bounds hold for any distribution
  - Bayes: prior choice impacts inference
- Posterior

PAC-Bayes bounds vs. Bayesian learning

- **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

PAC-Bayes bounds vs. Bayesian learning

- 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 vs. Bayesian learning

- **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

A classical PAC-Bayesian bound

Pre-history: PAC analysis of Bayesian estimators

Shawe-Taylor and Williamson (1997); Shawe-Taylor et al. (1998)

A classical PAC-Bayesian bound

Pre-history: PAC analysis of Bayesian estimators
Shawe-Taylor and Williamson (1997); Shawe-Taylor et al. (1998)

Birth: PAC-Bayesian bound
McAllester (1998, 1999)

McAllester Bound

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

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

A flexible framework

A flexible framework

Since 1997, PAC-Bayes has been successfully used in many machine learning settings (this list is by no means exhaustive).

Statistical learning theory *Shawe-Taylor and Williamson (1997); McAllester (1998, 1999, 2003a,b); Seeger (2002, 2003); Maurer (2004); Catoni (2004, 2007); Audibert and Bousquet (2007); Thiemann et al. (2017); Guedj (2019); Mhammedi et al. (2019, 2020); Guedj and Pujol (2019); Haddouche et al. (2020)*

SVMs & linear classifiers *Langford and Shawe-Taylor (2002); McAllester (2003a); Germain et al. (2009a)*

Supervised learning algorithms reinterpreted as bound minimizers *Ambroladze et al. (2007); Shawe-Taylor and Hardoon (2009); Germain et al. (2009b)*

High-dimensional regression *Alquier and Lounici (2011); Alquier and Biau (2013); Guedj and Alquier (2013); Li et al. (2013); Guedj and Robbiano (2018)*

Classification *Langford and Shawe-Taylor (2002); Catoni (2004, 2007); Lacasse et al. (2007); Parrado-Hernández et al. (2012)*

A flexible framework

- Transductive learning, domain adaptation *Derbeko et al. (2004); Bégin et al. (2014); Germain et al. (2016); Nozawa et al. (2020)*
- Non-iid or heavy-tailed data *Lever et al. (2010); Seldin et al. (2011, 2012); Alquier and Guedj (2018); Holland (2019)*
- Density estimation *Seldin and Tishby (2010); Higgs and Shawe-Taylor (2010)*
- Reinforcement learning *Fard and Pineau (2010); Fard et al. (2011); Seldin et al. (2011, 2012); Ghavamzadeh et al. (2015)*
- Sequential learning *Gerchinovitz (2011); Li et al. (2018)*
- Algorithmic stability, differential privacy *London et al. (2014); London (2017); Dziugaite and Roy (2018a,b); Rivasplata et al. (2018)*
- Deep neural networks *Dziugaite and Roy (2017); Neyshabur et al. (2017); Zhou et al. (2019); Letarte et al. (2019); Biggs and Guedj (2020)*

…

PAC-Bayes-inspired learning algorithms

PAC-Bayes-inspired learning algorithms

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

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

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

PAC-Bayes-inspired 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:

$$h \sim Q^\star$$

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

PAC-Bayes-inspired 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:

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

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

PAC-Bayes-inspired 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:

$$h \sim Q^\star$$

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

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

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 $\int (\exp \circ \phi) \mathrm{d}P < \infty$,

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

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

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

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

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

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

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

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

$$-\mathrm{KL}(Q, G) = -\int \log \left(\frac{\mathrm{d}Q}{\mathrm{d}P} \frac{\mathrm{d}P}{\mathrm{d}G} \right) \mathrm{d}Q$$

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

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

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

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

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

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

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

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 $Q = G$:

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

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 $Q = G$:

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

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

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 $\phi = -\lambda R_{\mathrm{in}}$,

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

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

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

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.
- ☑ 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.
- ■ Mhammedi, Guedj and Williamson (2020). PAC-Bayesian Bound for the Conditional Value at Risk, NeurIPS 2020 (spotlight).









Alquier and Guedj (2018). Simpler PAC-Bayesian bounds for hostile data, Machine Learning

Learning with non-iid or heavy-tailed data

We drop the iid and bounded loss assumptions.

Learning with non-iid or heavy-tailed data

We drop the iid and bounded loss assumptions. For any integer q,

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

Learning with non-iid or heavy-tailed data

We drop the iid and bounded loss assumptions. For any integer q,

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

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.

Learning with non-iid or heavy-tailed data

We drop the iid and bounded loss assumptions. For any integer q,

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

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 $\mathrm{KL}(Q \| P) = D_{x \log(x)}(Q, P)$.

Power function: $\phi_p \colon x \mapsto x^p$.

PAC-Bayes with f-divergences

PAC-Bayes with f-divergences

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

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

PAC-Bayes with f-divergences

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

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

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).

PAC-Bayes with f-divergences

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

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

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,

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

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

PAC-Bayes with f-divergences

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

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

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,

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

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

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

Proof

Proof

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

Proof

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

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

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

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

Proof

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

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

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

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

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

Proof

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

$$
\begin{aligned}
& \left|\int R_{\mathrm{out}} \mathrm{d}Q - \int R_{\mathrm{in}} \mathrm{d}Q\right| \\
\text{Jensen} \quad & \leqslant \int \Delta \mathrm{d}Q \\
\text{Change of measure} \quad & = \int \Delta \frac{\mathrm{d}Q}{\mathrm{d}P} \mathrm{d}P \\
\text{Hölder} \quad & \leqslant \left(\int \Delta^q \mathrm{d}P\right)^{\frac{1}{q}} \left(\int \left(\frac{\mathrm{d}Q}{\mathrm{d}P}\right)^p \mathrm{d}P\right)^{\frac{1}{p}} \\
\text{Markov} \quad & \underset{1-\delta}{\leqslant} \left(\frac{\mathbb{E}\int \Delta^q \mathrm{d}P}{\delta}\right)^{\frac{1}{q}} \left(\int \left(\frac{\mathrm{d}Q}{\mathrm{d}P}\right)^p \mathrm{d}P\right)^{\frac{1}{p}}
\end{aligned}
$$

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

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

Nozawa, Germain and Guedj (2020). PAC-Bayesian Contrastive Unsupervised Representation Learning, UAI

Letarte, Germain, Guedj and Laviolette (2019).
Dichotomize and generalize: PAC-Bayesian binary activated deep neural networks, NeurIPS 2019

Standard Neural Networks

Classification setting:

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

Standard Neural Networks

Classification setting:

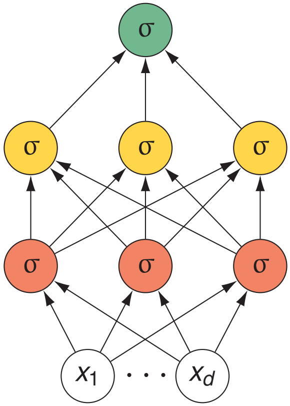
- $\mathbf{x} \in \mathbb{R}^{d_0}$
- $y \in \{-1, 1\}$

Architecture:

- *L fully connected* layers
- d_k denotes the number of neurons of the k^{th} layer
- $\sigma : \mathbb{R} \to \mathbb{R}$ is the *activation function*

Parameters:

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



Standard Neural Networks

Classification setting:

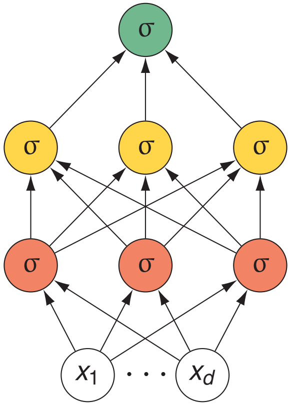
- $\mathbf{x} \in \mathbb{R}^{d_0}$
- $y \in \{-1, 1\}$

Architecture:

- *L fully connected* layers
- d_k denotes the number of neurons of the k^{th} layer
- $\sigma : \mathbb{R} \to \mathbb{R}$ is the *activation function*

Parameters:

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



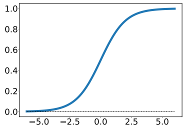
Prediction

$$f_\theta(\mathbf{x}) = \sigma\Big(\mathbf{w}_L \sigma\Big(\mathbf{W}_{L-1} \sigma\Big(\dots \sigma\Big(\mathbf{W}_1 \mathbf{x}\Big)\Big)\Big)\Big)\,.$$

PAC-Bayesian bounds for Stochastic NN

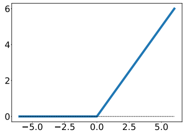
Langford and Caruana (2001)

- Shallow networks ($L = 2$)
- Sigmoid activation functions



Dziugaite and Roy (2017)

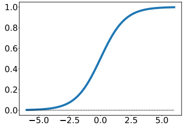
- Deep networks ($L > 2$)
- ReLU activation functions



PAC-Bayesian bounds for Stochastic NN

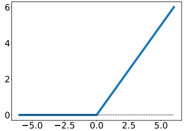
Langford and Caruana (2001)

- Shallow networks ($L = 2$)
- Sigmoid activation functions



Dziugaite and Roy (2017)

- Deep networks ($L > 2$)
- ReLU activation functions



Idea: Bound the expected loss of the network under a Gaussian perturbation of the weights

Empirical loss: $\underset{\theta' \sim \mathcal{N}(\theta, \Sigma)}{\mathbf{E}} R_{\mathrm{in}}(f_{\theta'}) \quad \longrightarrow \quad$ estimated by sampling

Complexity term: $\mathrm{KL}(\mathcal{N}(\theta, \Sigma) \| \mathcal{N}(\theta_0, \Sigma_0)) \quad \longrightarrow \quad$ closed form

Binary Activated Neural Networks

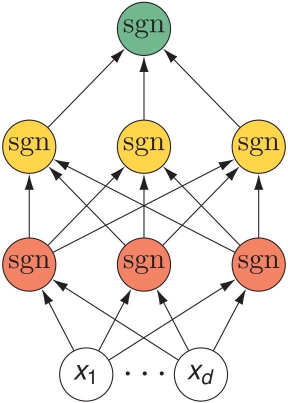
- $\mathbf{x} \in \mathbb{R}^{d_0}$
- $y \in \{-1, 1\}$

Architecture:

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

Parameters:

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



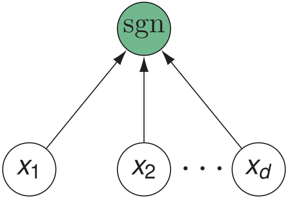
Prediction

$$f_\theta(\mathbf{x}) = \text{sgn}\Big(\mathbf{w}_L \text{sgn}\big(\mathbf{W}_{L-1} \text{sgn}\big(\ldots \text{sgn}\big(\mathbf{W}_1 \mathbf{x}\big)\big)\big)\Big),$$

One Layer (linear predictor)

Germain et al. (2009a)

$$f_{\mathbf{w}}(\mathbf{x}) \stackrel{\text{def}}{=} \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x}), \text{ with } \mathbf{w} \in \mathbb{R}^{d_0}.$$



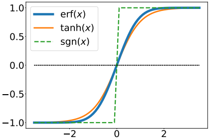
One Layer (linear predictor)

Germain et al. (2009a)

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

PAC-Bayes analysis:

- Space of all linear classifiers $\mathcal{F}_d \stackrel{\text{def}}{=} \{f_{\mathbf{v}} | \mathbf{v} \in \mathbb{R}^d\}$



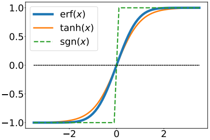
One Layer (linear predictor)

Germain et al. (2009a)

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

PAC-Bayes analysis:

- Space of all linear classifiers $\mathcal{F}_d \stackrel{\text{def}}{=} \{f_{\mathbf{v}} | \mathbf{v} \in \mathbb{R}^d\}$
- Gaussian posterior $Q_{\mathbf{w}} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}, I_d)$ over $\mathcal{F}_d$



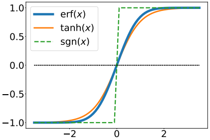
One Layer (linear predictor)

Germain et al. (2009a)

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

PAC-Bayes analysis:

- Space of all linear classifiers $\mathcal{F}_d \stackrel{\text{def}}{=} \{f_{\mathbf{v}} | \mathbf{v}\in\mathbb{R}^d\}$
- Gaussian posterior $Q_{\mathbf{w}} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}, I_d)$ over $\mathcal{F}_d$
- Gaussian prior $P_{\mathbf{w}_0} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}_0, I_d)$ over $\mathcal{F}_d$



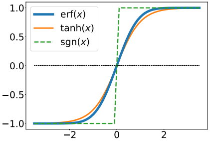
One Layer (linear predictor)

Germain et al. (2009a)

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

PAC-Bayes analysis:

- Space of all linear classifiers $\mathcal{F}_d \stackrel{\text{def}}{=} \{f_{\mathbf{v}} | \mathbf{v} \in \mathbb{R}^d\}$
- Gaussian posterior $Q_{\mathbf{w}} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}, I_d)$ over $\mathcal{F}_d$
- Gaussian prior $P_{\mathbf{w}_0} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}_0, I_d)$ over $\mathcal{F}_d$
- Predictor $F_{\mathbf{w}}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{E}_{\mathbf{v}\sim Q_{\mathbf{w}}} f_{\mathbf{v}}(\mathbf{x}) = \operatorname{erf}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\sqrt{d}\|\mathbf{x}\|}\right)$



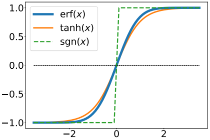
One Layer (linear predictor)

Germain et al. (2009a)

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

PAC-Bayes analysis:

- Space of all linear classifiers $\mathcal{F}_d \stackrel{\text{def}}{=} \{f_{\mathbf{v}} | \mathbf{v}\in\mathbb{R}^d\}$
- Gaussian posterior $Q_{\mathbf{w}} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}, I_d)$ over $\mathcal{F}_d$
- Gaussian prior $P_{\mathbf{w}_0} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}_0, I_d)$ over $\mathcal{F}_d$
- Predictor $F_{\mathbf{w}}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{E}_{\mathbf{v}\sim Q_{\mathbf{w}}} f_{\mathbf{v}}(\mathbf{x}) = \operatorname{erf}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\sqrt{d}\|\mathbf{x}\|}\right)$



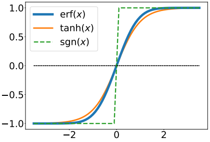
One Layer (linear predictor)

Germain et al. (2009a)

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

PAC-Bayes analysis:

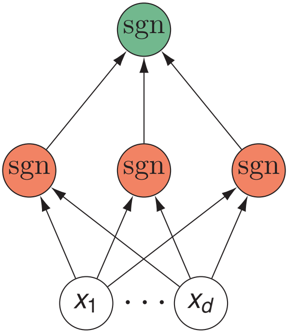
- Space of all linear classifiers $\mathcal{F}_d \stackrel{\text{def}}{=} \{f_{\mathbf{v}} | \mathbf{v} \in \mathbb{R}^d\}$
- Gaussian posterior $Q_{\mathbf{w}} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}, I_d)$ over $\mathcal{F}_d$
- Gaussian prior $P_{\mathbf{w}_0} \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{w}_0, I_d)$ over $\mathcal{F}_d$
- Predictor $F_{\mathbf{w}}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{E}_{\mathbf{v}\sim Q_{\mathbf{w}}} f_{\mathbf{v}}(\mathbf{x}) = \operatorname{erf}\left(\frac{\mathbf{w}\cdot\mathbf{x}}{\sqrt{d}\|\mathbf{x}\|}\right)$



Bound minimisation — under the linear loss $\ell(y, y') := \frac{1}{2}(1 - yy')$

$$C m R_{\text{in}}(F_{\mathbf{w}}) + \mathrm{KL}(Q_{\mathbf{w}} \| P_{\mathbf{w}_0}) = C\, \frac{1}{2} \sum_{i=1}^{m} \operatorname{erf}\left(-y_i \frac{\mathbf{w}\cdot\mathbf{x}_i}{\sqrt{d}\|\mathbf{x}_i\|}\right) + \frac{1}{2}\|\mathbf{w} - \mathbf{w}_0\|^2 .$$

Two Layers (shallow network)



Two Layers (shallow network)

Posterior $Q_\theta = \mathcal{N}(\theta, I_D)$, over the family of all networks $\mathcal{F}_D = \{f_{\tilde{\theta}} \mid \tilde{\theta} \in \mathbb{R}^D\}$, where

$$f_\theta(\mathbf{x}) = \mathrm{sgn}\big(\mathbf{w}_2 \cdot \mathrm{sgn}(\mathbf{W}_1 \mathbf{x})\big) \,.$$

Two Layers (shallow network)

Posterior $Q_\theta = \mathcal{N}(\theta, I_D)$, over the family of all networks $\mathcal{F}_D = \{f_{\tilde{\theta}} \mid \tilde{\theta} \in \mathbb{R}^D\}$, where

$$f_\theta(\mathbf{x}) = \mathrm{sgn}\big(\mathbf{w}_2 \cdot \mathrm{sgn}(\mathbf{W}_1 \mathbf{x})\big) .$$

$$F_\theta(\mathbf{x}) = \mathop{\mathbf{E}}_{\tilde{\theta} \sim Q_\theta} f_{\tilde{\theta}}(\mathbf{x})$$

Two Layers (shallow network)

Posterior $Q_\theta = \mathcal{N}(\theta, I_D)$, over the family of all networks $\mathcal{F}_D = \{f_{\tilde\theta} \mid \tilde\theta \in \mathbb{R}^D\}$, where

$$f_\theta(\mathbf{x}) = \mathrm{sgn}\big(\mathbf{w}_2 \cdot \mathrm{sgn}(\mathbf{W}_1\mathbf{x})\big)\,.$$

$$\begin{aligned}
F_\theta(\mathbf{x}) &= \mathop{\mathbf{E}}_{\tilde\theta\sim Q_\theta} f_{\tilde\theta}(\mathbf{x}) \\
&= \int_{\mathbb{R}^{d_1\times d_0}} Q_1(\mathbf{V}_1) \int_{\mathbb{R}^{d_1}} Q_2(\mathbf{v}_2)\mathrm{sgn}(\mathbf{v}_2\cdot\mathrm{sgn}(\mathbf{V}_1\mathbf{x}))\,d\mathbf{v}_2\,d\mathbf{V}_1 \\
&= \int_{\mathbb{R}^{d_1\times d_0}} Q_1(\mathbf{V}_1)\,\mathrm{erf}\left(\frac{\mathbf{w}_2\cdot\mathrm{sgn}(\mathbf{V}_1\mathbf{x})}{\sqrt{2}\|\mathrm{sgn}(\mathbf{V}_1\mathbf{x})\|}\right) d\mathbf{V}_1 \\
&= \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} \mathrm{erf}\left(\frac{\mathbf{w}_2\cdot\mathbf{s}}{\sqrt{2d_1}}\right) \int_{\mathbb{R}^{d_1\times d_0}} \mathbb{1}[\mathbf{s} = \mathrm{sgn}(\mathbf{V}_1\mathbf{x})]\,Q_1(\mathbf{V}_1)\,d\mathbf{V}_1 \\
&= \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} \underbrace{\mathrm{erf}\left(\frac{\mathbf{w}_2\cdot\mathbf{s}}{\sqrt{2d_1}}\right)}_{F_{\mathbf{w}_2}(\mathbf{s})} \underbrace{\prod_{i=1}^{d_1}\left[\frac{1}{2} + \frac{s_i}{2}\,\mathrm{erf}\left(\frac{\mathbf{w}_1^i\cdot\mathbf{x}}{\sqrt{2}\,\|\mathbf{x}\|}\right)\right]}_{\Pr(\mathbf{s}|\mathbf{x},\mathbf{W}_1)}\,.
\end{aligned}$$

Stochastic Approximation

$$F_\theta(\mathbf{x}) = \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} F_{\mathbf{w}_2}(\mathbf{s}) \Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}_1)$$

Monte Carlo sampling

We generate T random binary vectors $\{\mathbf{s}^t\}_{t=1}^T$ according to $\Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}_1)$

Stochastic Approximation

$$F_\theta(\mathbf{x}) = \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} F_{\mathbf{w}_2}(\mathbf{s}) \Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}_1)$$

Monte Carlo sampling

We generate T random binary vectors $\{\mathbf{s}^t\}_{t=1}^T$ according to $\Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}_1)$

Prediction.

$$F_\theta(\mathbf{x}) \approx \frac{1}{T}\sum_{t=1}^{T} F_{\mathbf{w}_2}(\mathbf{s}^t) .$$

Stochastic Approximation

$$F_\theta(\mathbf{x}) = \sum_{\mathbf{s}\in\{-1,1\}^{d_1}} F_{\mathbf{w}_2}(\mathbf{s}) \Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}_1)$$

Monte Carlo sampling

We generate T random binary vectors $\{\mathbf{s}^t\}_{t=1}^T$ according to $\Pr(\mathbf{s}|\mathbf{x}, \mathbf{W}_1)$

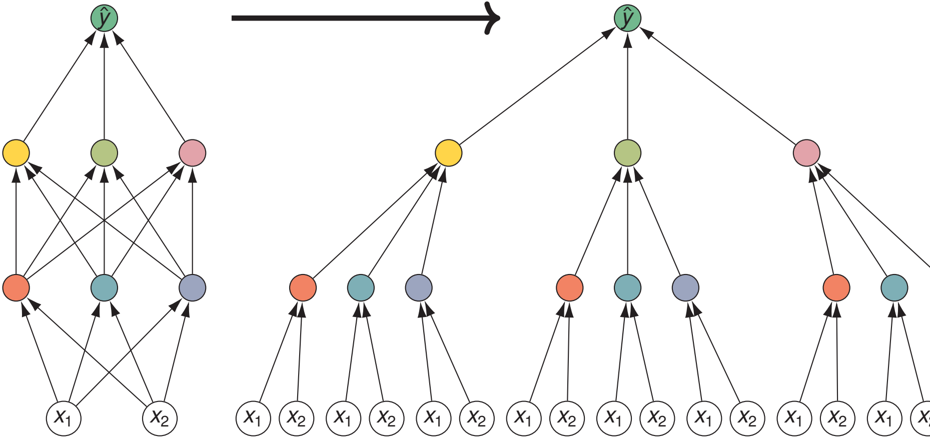
Prediction.

$$F_\theta(\mathbf{x}) \approx \frac{1}{T}\sum_{t=1}^{T} F_{\mathbf{w}_2}(\mathbf{s}^t)\,.$$

Derivatives.

$$\frac{\partial}{\partial \mathbf{w}_1^k} F_\theta(\mathbf{x}) \approx \frac{\mathbf{x}}{2^{\frac{3}{2}}\,\|\mathbf{x}\|}\mathrm{erf}'\left(\frac{\mathbf{w}_1^k\cdot\mathbf{x}}{\sqrt{2}\,\|\mathbf{x}\|}\right)\frac{1}{T}\sum_{t=1}^{T}\frac{s_k^t}{\Pr(s_k^t|\mathbf{x}, \mathbf{w}_1^k)} F_{\mathbf{w}_2}(\mathbf{s}^t)\,.$$

More Layers (deep)



$$F_1^{(j)}(\mathbf{x}) = \operatorname{erf}\left(\frac{\mathbf{w}_1^j \cdot \mathbf{x}}{\sqrt{2}\|\mathbf{x}\|}\right), \qquad F_{k+1}^{(j)}(\mathbf{x}) = \sum_{\mathbf{s}\in\{-1,1\}^{d_k}} \operatorname{erf}\left(\frac{\mathbf{w}_{k+1}^j \cdot \mathbf{s}}{\sqrt{2d_k}}\right) \prod_{i=1}^{d_k} \left(\frac{1}{2} + \frac{1}{2} s_i \times F_k^{(i)}(\mathbf{x})\right)$$

Generalisation bound

Let G_θ denote the predictor with posterior mean as parameters.
With probability at least $1-\delta$, for any $\theta \in \mathbb{R}^D$

$$R_{\mathrm{out}}(G_\theta) \leqslant$$

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

Numerical results

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

Numerical results

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$_{\text{pre}}$		
	E_S	E_T	E_S	E_T	E_S	E_T	Bound	E_S	E_T	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

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-Bayesian online clustering (Li et al., 2018)
- PAC-Bayesian bipartite ranking (Guedj and Robbiano, 2018)
- Online k-means clustering (Cohen-Addad et al., 2019)
- Sequential learning of principal curves (Guedj and Li, 2018)
- Stability and generalisation (Celisse and Guedj, 2016)
- Contrastive unsupervised learning (Nozawa 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 (nonlinear aggregation) in Python (Guedj and Srinivasa Desikan, 2018, 2020)
- Identifying subcommunities in social networks (Vendeville et al., 2020b,a)
- Prediction with multi-task Gaussian processes (Leroy et al., 2020)
- + a few others in the pipe, hopefully soon on arXiv!

This talk: `https://bguedj.github.io/talks/2021-01-21-seminar-ucl-stat`

References I

P. Alquier and G. Biau. Sparse single-index model. *Journal of Machine Learning Research*, 14:243–280, 2013.

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.

P. Alquier and K. Lounici. PAC-Bayesian theorems for sparse regression estimation with exponential weights. *Electronic Journal of Statistics*, 5:127–145, 2011.

A. Ambroladze, E. Parrado-Hernández, and J. Shawe-taylor. Tighter PAC-Bayes bounds. In *Advances in Neural Information Processing Systems, NIPS*, pages 9–16, 2007.

J.-Y. Audibert and O. Bousquet. Combining PAC-Bayesian and generic chaining bounds. *Journal of Machine Learning Research*, 2007.

L. Bégin, P. Germain, F. Laviolette, and J.-F. Roy. PAC-Bayesian theory for transductive learning. In *AISTATS*, 2014.

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

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

O. Catoni. *PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *Lecture notes – Monograph Series*. Institute of Mathematical Statistics, 2007.

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. *arXiv preprint arXiv:1909.06861*, 2019.

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

P. Derbeko, R. El-Yaniv, and R. Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *J. Artif. Intell. Res. (JAIR)*, 22, 2004.

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.

References II

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2017.

G. K. Dziugaite and D. M. Roy. Data-dependent PAC-Bayes priors via differential privacy. In *NeurIPS*, 2018a.

G. K. Dziugaite and D. M. Roy. Entropy-SGD optimizes the prior of a PAC-Bayes bound: Generalization properties of Entropy-SGD and data-dependent priors. In *International Conference on Machine Learning*, pages 1376–1385, 2018b.

M. M. Fard and J. Pineau. PAC-Bayesian model selection for reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

M. M. Fard, J. Pineau, and C. Szepesvári. PAC-Bayesian Policy Evaluation for Reinforcement Learning. In *UAI, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 195–202, 2011.

S. Gerchinovitz. *Prédiction de suites individuelles et cadre statistique classique : étude de quelques liens autour de la régression parcimonieuse et des techniques d'agrégation*. PhD thesis, Université Paris-Sud, 2011.

P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian Learning of Linear Classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*. Association for Computing Machinery, 2009a. doi: 10.1145/1553374.1553419. URL `https://doi.org/10.1145/1553374.1553419`.

P. Germain, A. Lacasse, M. Marchand, S. Shanian, and F. Laviolette. From PAC-Bayes bounds to KL regularization. In *Advances in Neural Information Processing Systems*, pages 603–610, 2009b.

P. Germain, A. Habrard, F. Laviolette, and E. Morvant. A new PAC-Bayesian perspective on domain adaptation. In *Proceedings of International Conference on Machine Learning*, volume 48, 2016.

M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends in Machine Learning*, 8(5-6):359–483, 2015.

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.

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

References III

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/info11020063. 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., 2020. URL https://arxiv.org/abs/2006.07279.

M. Higgs and J. Shawe-Taylor. A PAC-Bayes bound for tailored density estimation. In *Proceedings of the International Conference on Algorithmic Learning Theory (ALT)*, 2010.

M. Holland. PAC-Bayes under potentially heavy tails. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2715–2724. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8539-pac-bayes-under-potentially-heavy-tails.pdf.

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. Lacasse, F. Laviolette, M. Marchand, P. Germain, and N. Usunier. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In *Advances in Neural information processing systems*, pages 769–776, 2007.

J. Langford and R. Caruana. (Not) Bounding the True Error. In *NIPS*, pages 809–816. MIT Press, 2001.

J. Langford and J. Shawe-Taylor. PAC-Bayes & margins. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

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

References IV

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.

G. Lever, F. Laviolette, and J. Shawe-Taylor. Distribution-dependent PAC-Bayes priors. In *International Conference on Algorithmic Learning Theory*, pages 119–133. Springer, 2010.

C. Li, W. Jiang, and M. Tanner. General oracle inequalities for Gibbs posterior with application to ranking. In *Conference on Learning Theory*, pages 512–521, 2013.

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

B. London. A PAC-Bayesian analysis of randomized learning with application to stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2931–2940, 2017.

B. London, B. Huang, B. Taskar, and L. Getoor. PAC-Bayesian collective stability. In *Artificial Intelligence and Statistics*, pages 585–594, 2014.

A. Maurer. A note on the PAC-Bayesian Theorem. *arXiv preprint cs/0411099*, 2004.

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.

D. McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1), 2003a.

D. McAllester. Simplified PAC-Bayesian margin bounds. In *COLT*, 2003b.

Z. Mhammedi, P. D. Grunwald, and B. Guedj. PAC-Bayes Un-Expected Bernstein Inequality. *arXiv preprint arXiv:1905.13367*, 2019. Accepted at NeurIPS 2019.

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

B. Neyshabur, S. Bhojanapalli, D. A. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

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

References V

E. Parrado-Hernández, A. Ambroladze, J. Shawe-Taylor, and S. Sun. PAC-Bayes bounds with data dependent priors. *Journal of Machine Learning Research*, 13:3507–3531, 2012.

O. Rivasplata, E. Parrado-Hernandez, J. Shawe-Taylor, S. Sun, and C. Szepesvari. PAC-Bayes bounds for stable algorithms with instance-dependent priors. In *Advances in Neural Information Processing Systems*, pages 9214–9224, 2018.

M. Seeger. PAC-Bayesian generalization bounds for gaussian processes. *Journal of Machine Learning Research*, 3:233–269, 2002.

M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, 2003.

Y. Seldin and N. Tishby. PAC-Bayesian analysis of co-clustering and beyond. *Journal of Machine Learning Research*, 11:3595–3646, 2010.

Y. Seldin, P. Auer, F. Laviolette, J. Shawe-Taylor, and R. Ortner. PAC-Bayesian analysis of contextual bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer. PAC-Bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 58(12):7086–7093, 2012.

J. Shawe-Taylor and D. Hardoon. Pac-bayes analysis of maximum entropy classification. In *Proceedings on the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

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.

J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5), 1998.

N. Thiemann, C. Igel, O. Wintenberger, and Y. Seldin. A Strongly Quasiconvex PAC-Bayesian Bound. In *International Conference on Algorithmic Learning Theory, ALT*, pages 466–492, 2017.

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

A. Vendeville, B. Guedj, and S. Zhou. Forecasting elections results via the voter model with stubborn nodes. Submitted., 2020a. URL `https://arxiv.org/abs/2009.10627`.

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

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.

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 *ICLR*, 2019.