Statistical Learning Theory for Modern Machine Learning

John Shawe-Taylor

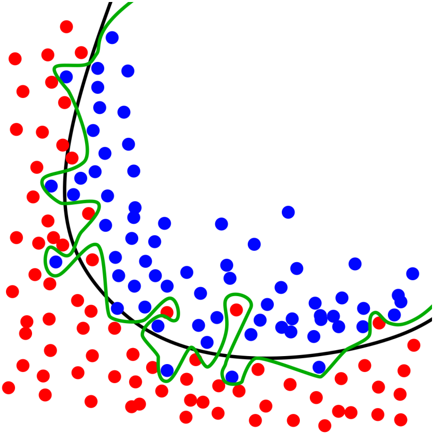
Dept of Computer Science, University College London
IRCAI, Insitute Jozef Stefan, Ljubljana

Collaborators: Benjamin Guedj, Omar Rivasplata, Maria Perez-Ortiz, Emilio Parrado-Hernandez, Amiran Ambroladze, Shiliang Sun

CAIML Symposium: The Secret Ingredients for Improving Artificial Intelligence
May 2023

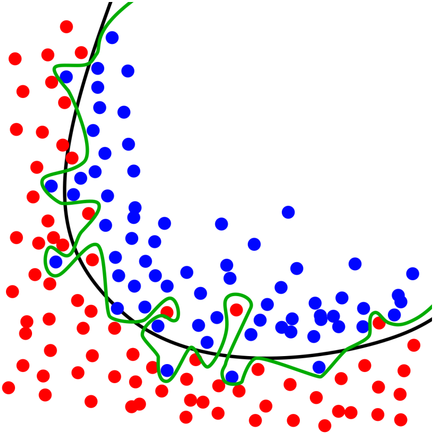
Learning is to be able to generalise

Learning is to be able to generalise



[Figure from Wikipedia]

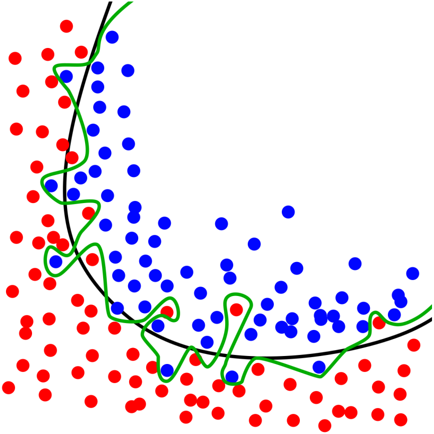
Learning is to be able to generalise



[Figure from Wikipedia]

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

Learning is to be able to generalise

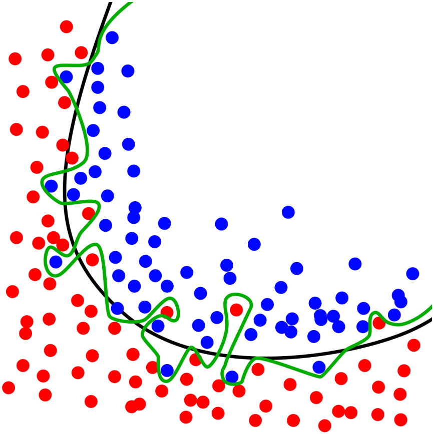


[Figure from Wikipedia]

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?
  $\triangleright$ can be misleading: learner only has one sample

- Statistical Learning Theory: tail of the distribution
  $\triangleright$ finding bounds which hold with high probability
  over random samples of size m

- Compare to a statistical test – at 99% confidence level
  $\triangleright$ chances of the conclusion not being true are less than 1%

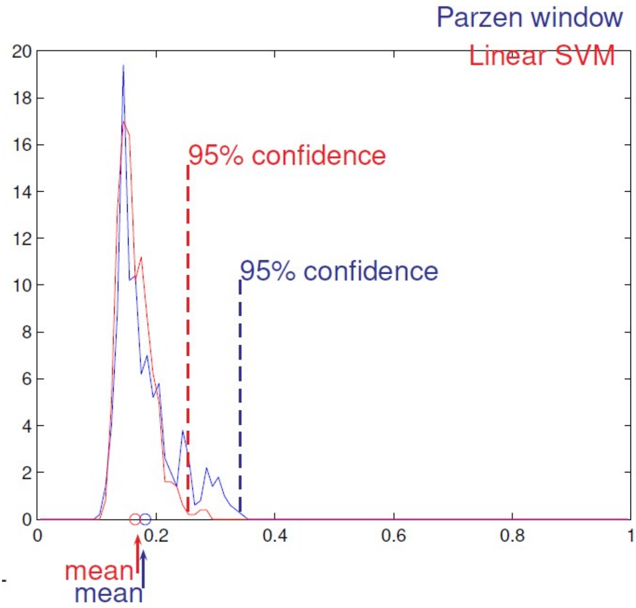
- PAC: probably approximately correct [59]
  Use a 'confidence parameter' δ: $\mathbb{P}^m[\text{large error}] \leqslant \delta$
  δ is the probability of being misled by the training set

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 [59]
  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: $\mathbb{P}^m[\text{approximately correct}] \geqslant 1 - \delta$

Error distribution picture



Mathematical formalization

Mathematical formalization

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)

Mathematical formalization

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)

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

Mathematical formalization

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)

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

Classical assumptions:

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

Mathematical formalization

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)

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

Classical assumptions:

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

▷ these can be relaxed (but not in this talk)

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

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

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

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
- $\ell(h(X), Y) = -\log(h(X))$: log loss (density estimation) TODO

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

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

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

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

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

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

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

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

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

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

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

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

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 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.

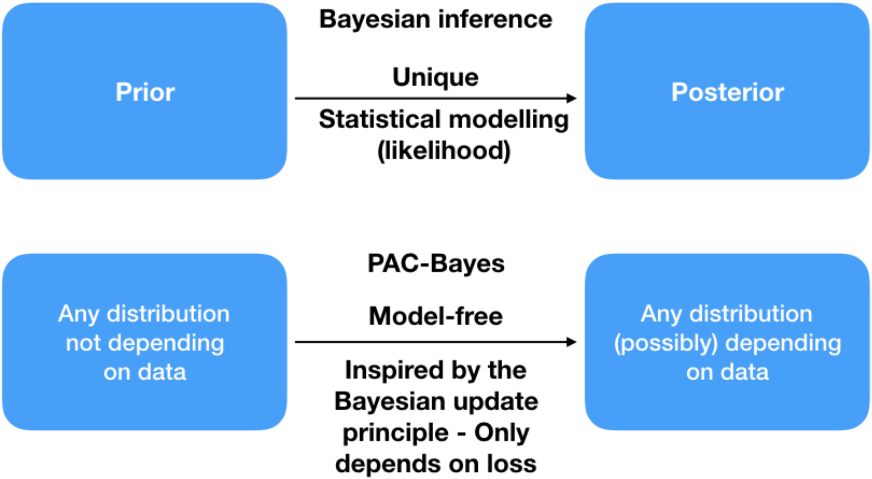
The risk measures $R_{\text{in}}(h)$ and $R_{\text{out}}(h)$ are extended by averaging:

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

$\mathrm{KL}(Q \| P) = \underset{h \sim Q}{\mathbf{E}} \ln \frac{Q(h)}{P(h)}$ 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 the 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: randomness lies in the noise model generating the output

A General PAC-Bayesian Theorem

Δ-function: "distance" between $R_{\text{in}}(Q)$ and $R_{\text{out}}(Q)$

Convex function $\Delta : [0,1] \times [0,1] \to \mathbb{R}$.

General theorem (*Bégin et al. [7, 8], Germain [21]*)

For any distribution D on $\mathcal{X} \times \mathcal{Y}$*, for any set* $\mathcal{H}$ *of voters, for any distribution* P *on* $\mathcal{H}$*, for any* $\delta \in (0,1]$*, and for any* Δ*-function, we have, with probability at least* $1-\delta$ *over the choice of* $S \sim D^m$*,*

$$\forall\, Q \text{ on } \mathcal{H} : \quad \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \leqslant \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{\mathcal{I}_{\Delta}(m)}{\delta}\right],$$

A General PAC-Bayesian Theorem

Δ-function: "distance" between $R_{\mathrm{in}}(Q)$ and $R_{\mathrm{out}}(Q)$

Convex function $\Delta : [0,1] \times [0,1] \to \mathbb{R}$.

General theorem (*Bégin et al. [7, 8], Germain [21]*)

For any distribution D on $\mathcal{X} \times \mathcal{Y}$, for any set $\mathcal{H}$ of voters, for any distribution P on $\mathcal{H}$, for any $\delta \in (0,1]$, and for any Δ-function, we have, with probability at least $1-\delta$ over the choice of $S \sim D^m$,

$$\forall\, Q \text{ on } \mathcal{H} : \quad \Delta\Big(R_{\mathrm{in}}(Q), R_{\mathrm{out}}(Q)\Big) \;\leqslant\; \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{\mathcal{I}_\Delta(m)}{\delta}\right],$$

where

$$\mathcal{I}_\Delta(m) \;=\; \sup_{r \in [0,1]} \left[\, \sum_{k=0}^{m} \underbrace{\binom{m}{k} r^k (1-r)^{m-k}}_{\mathbf{Bin}\left(k;m,r\right)} e^{m\Delta\left(\frac{k}{m},\, r\right)} \right].$$

Proof of the general theorem

General theorem

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof ideas.

Change of Measure Inequality

For any P and Q on $\mathcal{H}$, and for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$, we have

$$\begin{aligned}
-\ln\left(\mathop{\mathbf{E}}_{h\sim P} e^{\phi(h)}\right) &= -\ln \mathop{\mathbf{E}}_{h\sim Q}\left(\frac{P(h)}{Q(h)} e^{\phi(h)}\right) \\
&\leqslant \mathop{\mathbf{E}}_{h\sim Q} \ln\left(\frac{Q(h)}{P(h)}\right) - \mathop{\mathbf{E}}_{h\sim Q} \phi(h) \\
&= \mathrm{KL}(Q\|P) - \mathop{\mathbf{E}}_{h\sim Q} \phi(h).
\end{aligned}$$

Proof of the general theorem

General theorem

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\rm in}(Q), R_{\rm out}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof ideas.

Change of Measure Inequality

For any P and Q on $\mathcal{H}$, and for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$, we have

$$\begin{aligned}
-\ln\left(\mathop{\mathbf{E}}_{h\sim P} e^{\phi(h)}\right) &= -\ln \mathop{\mathbf{E}}_{h\sim Q}\left(\frac{P(h)}{Q(h)}e^{\phi(h)}\right)\\
&\leqslant \mathop{\mathbf{E}}_{h\sim Q} \ln\left(\frac{Q(h)}{P(h)}\right) - \mathop{\mathbf{E}}_{h\sim Q}\phi(h)\\
&= \mathrm{KL}(Q\|P) - \mathop{\mathbf{E}}_{h\sim Q}\phi(h).
\end{aligned}$$

Markov's inequality

for a random variable X satisfying $X \geqslant 0$

$$\Pr\left(X \geqslant a\right) \le \frac{\mathbf{E}\,X}{a} \quad\Longleftrightarrow\quad \Pr\left(X \leqslant \frac{\mathbf{E}\,X}{\delta}\right) \ge 1-\delta\,.$$

Proof of the general theorem

Probability of observing k misclassifications among m examples

Given a voter h, consider a **binomial variable** of m trials with **success** $R_{\mathrm{out}}(h)$:

$$\Pr_{S\sim D^m}\left(R_{\mathrm{in}}(h)=\frac{k}{m}\right) \quad = \quad \binom{m}{k}\Big(R_{\mathrm{out}}(h)\Big)^k\Big(1-R_{\mathrm{out}}(h)\Big)^{m-k} \;=\; \mathbf{Bin}\Big(k; m, R_{\mathrm{out}}(h)\Big)$$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\rm in}(Q), R_{\rm out}(Q)\Big) \ \le\ \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof.

$$m \cdot \Delta\Big(\mathop{\mathbf{E}}_{h\sim Q} R_{\rm in}(h),\ \mathop{\mathbf{E}}_{h\sim Q} R_{\rm out}(h)\Big)$$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof.

$$m\cdot\Delta\Big(\mathop{\mathbf{E}}_{h\sim Q} R_{\text{in}}(h), \mathop{\mathbf{E}}_{h\sim Q} R_{\text{out}}(h)\Big)$$

Jensen's Inequality

$$\leqslant \quad \mathop{\mathbf{E}}_{h\sim Q} m\cdot\Delta\Big(R_{\text{in}}(h), R_{\text{out}}(h)\Big)$$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof.

$$m\cdot\Delta\Big(\mathop{\mathbf{E}}_{h\sim Q} R_{\text{in}}(h),\ \mathop{\mathbf{E}}_{h\sim Q} R_{\text{out}}(h)\Big)$$

Jensen's Inequality $\leqslant \mathop{\mathbf{E}}_{h\sim Q} m\cdot\Delta\Big(R_{\text{in}}(h), R_{\text{out}}(h)\Big)$

Change of measure $\leqslant \mathrm{KL}(Q\|P) + \ln \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta\left(R_{\text{in}}(h), R_{\text{out}}(h)\right)}$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}: \ \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \ \leq \ \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof.

$$m\cdot\Delta\Big(\mathop{\mathbf{E}}_{h\sim Q} R_{\text{in}}(h), \mathop{\mathbf{E}}_{h\sim Q} R_{\text{out}}(h)\Big)$$

Jensen's Inequality	$\leqslant$	$\mathop{\mathbf{E}}_{h\sim Q} m\cdot\Delta\Big(R_{\text{in}}(h), R_{\text{out}}(h)\Big)$
Change of measure	$\leqslant$	$\mathrm{KL}(Q\|P) + \ln \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta\left(R_{\text{in}}(h), R_{\text{out}}(h)\right)}$
Markov's Inequality	$\leq_{1-\delta}$	$\mathrm{KL}(Q\|P) + \ln\frac{1}{\delta}\mathop{\mathbf{E}}_{S'\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\Delta(R_{\text{in}}(h), R_{\text{out}}(h))}$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof.

$$m\cdot\Delta\Big(\mathop{\mathbf{E}}_{h\sim Q} R_{\text{in}}(h), \mathop{\mathbf{E}}_{h\sim Q} R_{\text{out}}(h)\Big)$$

Jensen's Inequality

$$\leqslant \mathop{\mathbf{E}}_{h\sim Q} m\cdot\Delta\Big(R_{\text{in}}(h), R_{\text{out}}(h)\Big)$$

Change of measure

$$\leqslant \mathrm{KL}(Q\|P) + \ln \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta\left(R_{\text{in}}(h), R_{\text{out}}(h)\right)}$$

Markov's Inequality

$$\le_{1-\delta} \mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} \mathop{\mathbf{E}}_{S'\sim D^m} \mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\Delta(R_{\text{in}}(h), R_{\text{out}}(h))}$$

Expectation swap

$$= \mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} \mathop{\mathbf{E}}_{h\sim P} \mathop{\mathbf{E}}_{S'\sim D^m} e^{m\cdot\Delta(R_{\text{in}}(h), R_{\text{out}}(h))}$$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\mathrm{in}}(Q), R_{\mathrm{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof.

$$m\cdot\Delta\Big(\mathop{\mathbf{E}}_{h\sim Q} R_{\mathrm{in}}(h), \mathop{\mathbf{E}}_{h\sim Q} R_{\mathrm{out}}(h)\Big)$$

Jensen's Inequality $\leqslant \quad \mathop{\mathbf{E}}_{h\sim Q} m\cdot\Delta\Big(R_{\mathrm{in}}(h), R_{\mathrm{out}}(h)\Big)$

Change of measure $\leqslant \quad \mathrm{KL}(Q\|P) + \ln \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta\left(R_{\mathrm{in}}(h), R_{\mathrm{out}}(h)\right)}$

Markov's Inequality $\le_{1-\delta} \quad \mathrm{KL}(Q\|P) + \ln\frac{1}{\delta}\mathop{\mathbf{E}}_{S'\sim D^m}\ \mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\Delta(R_{\mathrm{in}}(h), R_{\mathrm{out}}(h))}$

Expectation swap $= \quad \mathrm{KL}(Q\|P) + \ln\frac{1}{\delta}\mathop{\mathbf{E}}_{h\sim P}\ \mathop{\mathbf{E}}_{S'\sim D^m} e^{m\cdot\Delta(R_{\mathrm{in}}(h), R_{\mathrm{out}}(h))}$

Binomial law $= \quad \mathrm{KL}(Q\|P) + \ln\frac{1}{\delta}\mathop{\mathbf{E}}_{h\sim P}\sum_{k=0}^{m}\mathbf{Bin}\Big(k; m, R_{\mathrm{out}}(h)\Big)\, e^{m\cdot\Delta(\frac{k}{m}, R_{\mathrm{out}}(h))}$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{\mathcal{I}_{\Delta}(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Proof.

$$m \cdot \Delta\Big(\mathop{\mathbf{E}}_{h\sim Q} R_{\text{in}}(h), \mathop{\mathbf{E}}_{h\sim Q} R_{\text{out}}(h)\Big)$$

Jensen's Inequality $\leqslant \mathop{\mathbf{E}}_{h\sim Q} m\cdot\Delta\Big(R_{\text{in}}(h), R_{\text{out}}(h)\Big)$

Change of measure $\leqslant \mathrm{KL}(Q\|P) + \ln \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta\left(R_{\text{in}}(h), R_{\text{out}}(h)\right)}$

Markov's Inequality $\leq_{1-\delta} \mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} \mathop{\mathbf{E}}_{S'\sim D^m} \mathop{\mathbf{E}}_{h\sim P} e^{m\cdot\Delta(R_{\text{in}}(h), R_{\text{out}}(h))}$

Expectation swap $= \mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} \mathop{\mathbf{E}}_{h\sim P} \mathop{\mathbf{E}}_{S'\sim D^m} e^{m\cdot\Delta(R_{\text{in}}(h), R_{\text{out}}(h))}$

Binomial law $= \mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} \mathop{\mathbf{E}}_{h\sim P} \sum_{k=0}^{m} \mathbf{Bin}\Big(k; m, R_{\text{out}}(h)\Big) e^{m\cdot\Delta(\frac{k}{m}, R_{\text{out}}(h))}$

Supremum over risk $\leqslant \mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} \sup_{r\in[0,1]} \left[\sum_{k=0}^{m} \mathbf{Bin}\Big(k; m, r\Big) e^{m\Delta(\frac{k}{m}, r)}\right]$

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\rm in}(Q), R_{\rm out}(Q)\Big) \ \le\ \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1{-}\delta\,.$$

Proof.

		$m\cdot\Delta\Big(\mathbf{E}_{h\sim Q} R_{\rm in}(h), \mathbf{E}_{h\sim Q} R_{\rm out}(h)\Big)$
Jensen's Inequality	$\leqslant$	$\mathbf{E}_{h\sim Q}\, m\cdot\Delta\Big(R_{\rm in}(h), R_{\rm out}(h)\Big)$
Change of measure	$\leqslant$	$\mathrm{KL}(Q\|P) + \ln \mathbf{E}_{h\sim P}\, e^{m\Delta\left(R_{\rm in}(h), R_{\rm out}(h)\right)}$
Markov's Inequality	$\le_{1-\delta}$	$\mathrm{KL}(Q\|P) + \ln \frac{1}{\delta}\, \mathbf{E}_{S'\sim D^m}\, \mathbf{E}_{h\sim P}\, e^{m\cdot\Delta(R_{\rm in}(h), R_{\rm out}(h))}$
Expectation swap	$=$	$\mathrm{KL}(Q\|P) + \ln \frac{1}{\delta}\, \mathbf{E}_{h\sim P}\, \mathbf{E}_{S'\sim D^m}\, e^{m\cdot\Delta(R_{\rm in}(h), R_{\rm out}(h))}$
Binomial law	$=$	$\mathrm{KL}(Q\|P) + \ln \frac{1}{\delta}\, \mathbf{E}_{h\sim P} \sum_{k=0}^{m} \mathbf{Bin}\Big(k; m, R_{\rm out}(h)\Big)\, e^{m\cdot\Delta(\frac{k}{m}, R_{\rm out}(h))}$
Supremum over risk	$\leqslant$	$\mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} \sup_{r\in[0,1]} \left[\sum_{k=0}^{m} \mathbf{Bin}\Big(k; m, r\Big)\, e^{m\Delta(\frac{k}{m}, r)}\right]$
	$=$	$\mathrm{KL}(Q\|P) + \ln \frac{1}{\delta}\, \mathcal{I}_\Delta(m)\,.$ $\square$

General theorem

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\mathrm{in}}(Q), R_{\mathrm{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Corollary

[...] with probability at least $1-\delta$ *over the choice of* $S \sim D^m$*, for all* Q *on* $\mathcal{H}$:

(a) $\mathrm{kl}\Big(R_{\mathrm{in}}(Q), R_{\mathrm{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{m}}{\delta}\right]$, *Langford and Seeger [31]*

$$\mathrm{kl}(q,p) \overset{\text{def}}{=} q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}$$

General theorem

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\rm in}(Q), R_{\rm out}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Corollary

[...] with probability at least $1-\delta$ *over the choice of* $S\sim D^m$*, for all* Q *on* $\mathcal{H}$:

(a) $\mathrm{kl}\Big(R_{\rm in}(Q), R_{\rm out}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{m}}{\delta}\right],$ *Langford and Seeger [31]*

(b) $R_{\rm out}(Q) \le R_{\rm in}(Q)) + \sqrt{\frac{1}{2m}\left[\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{m}}{\delta}\right]},$ *McAllester [40, 43]*

$$\mathrm{kl}(q,p) \stackrel{\text{def}}{=} q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p} \geqslant 2(q-p)^2\,,$$

General theorem

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\mathrm{in}}(Q), R_{\mathrm{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{\mathcal{J}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Corollary

[...] with probability at least $1-\delta$ *over the choice of* $S \sim D^m$*, for all* Q *on* $\mathcal{H}$ *:*

(a) $\mathrm{kl}\Big(R_{\mathrm{in}}(Q), R_{\mathrm{out}}(Q)\Big) \le \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{m}}{\delta}\right]$, *Langford and Seeger [31]*

(b) $R_{\mathrm{out}}(Q) \le R_{\mathrm{in}}(Q)) + \sqrt{\frac{1}{2m}\left[\mathrm{KL}(Q\|P) + \ln\frac{2\sqrt{m}}{\delta}\right]}$, *McAllester [40, 43]*

(c) $R_{\mathrm{out}}(Q) \le \frac{1}{1-e^{-c}}\left(c\cdot R_{\mathrm{in}}(Q) + \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln\frac{1}{\delta}\right]\right)$, *Catoni [11]*

$$\mathrm{kl}(q,p) \stackrel{\text{def}}{=} q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p} \geqslant 2(q-p)^2\,,$$

$$\Delta_c(q,p) \stackrel{\text{def}}{=} -\ln[1-(1-e^{-c})\cdot p] - c\cdot q\,,$$

General theorem

$$\Pr_{S\sim D^m}\left(\forall\, Q \text{ on } \mathcal{H}:\ \Delta\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{\mathcal{I}_\Delta(m)}{\delta}\right]\right) \geqslant 1-\delta\,.$$

Corollary

[...] with probability at least $1-\delta$ *over the choice of* $S \sim D^m$*, for all* Q *on* $\mathcal{H}$ *:*

(a) $\mathrm{kl}\Big(R_{\text{in}}(Q), R_{\text{out}}(Q)\Big) \leq \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{2\sqrt{m}}{\delta}\right],$ *Langford and Seeger [31]*

(b) $R_{\text{out}}(Q) \leq R_{\text{in}}(Q)) + \sqrt{\frac{1}{2m}\left[\mathrm{KL}(Q\|P) + \ln \frac{2\sqrt{m}}{\delta}\right]},$ *McAllester [40, 43]*

(c) $R_{\text{out}}(Q) \leq \frac{1}{1-e^{-c}}\left(c \cdot R_{\text{in}}(Q) + \frac{1}{m}\left[\mathrm{KL}(Q\|P) + \ln \frac{1}{\delta}\right]\right),$ *Catoni [11]*

(d) $R_{\text{out}}(Q) \leq R_{\text{in}}(Q) + \frac{1}{\lambda}\left[\mathrm{KL}(Q\|P) + \ln \frac{1}{\delta} + f(\lambda, m)\right].$ *Alquier et al. [4]*

$$\mathrm{kl}(q,p) \overset{\text{def}}{=} q \ln \frac{q}{p} + (1-q)\ln\frac{1-q}{1-p} \geqslant 2(q-p)^2\,,$$

$$\Delta_c(q,p) \overset{\text{def}}{=} -\ln[1-(1-e^{-c})\cdot p] - c\cdot q\,,$$

$$\Delta_\lambda(q,p) \overset{\text{def}}{=} \frac{\lambda}{m}(p-q)\,.$$

Proof of the Langford/Seeger bound

Follows immediately from General Theorem by choosing $\Delta(q, p) = \mathrm{kl}(q, p)$.

Proof of the Langford/Seeger bound

Follows immediately from General Theorem by choosing $\Delta(q,p) = \mathrm{kl}(q,p)$.

- Indeed, in that case we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{S\sim D^m}\mathop{\mathbf{E}}_{h\sim P} e^{m\Delta(R_S(h),R(h))} &= \mathop{\mathbf{E}}_{h\sim P}\mathop{\mathbf{E}}_{S\sim D^m}\left(\frac{R_S(h)}{R(h)}\right)^{mR_S(h)}\left(\frac{1-R_S(h)}{1-R(h)}\right)^{m(1-R_S(h))} \\
&= \mathop{\mathbf{E}}_{h\sim P}\sum_{k=0}^{m}\Pr_{S\sim D^m}\left(R_S(h)=\tfrac{k}{m}\right)\left(\frac{\frac{k}{m}}{R(h)}\right)^k\left(\frac{1-\frac{k}{m}}{1-R(h)}\right)^{m-k} \\
&= \sum_{k=0}^{m}\binom{m}{k}(k/m)^k(1-k/m)^{m-k}, \qquad (1) \\
&\leqslant 2\sqrt{m}.
\end{aligned}$$

□

Proof of the Langford/Seeger bound

Follows immediately from General Theorem by choosing $\Delta(q, p) = \mathrm{kl}(q, p)$.

- Indeed, in that case we have

$$
\begin{aligned}
\mathop{\mathbf{E}}_{S\sim D^m} \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta(R_S(h),R(h))} &= \mathop{\mathbf{E}}_{h\sim P} \mathop{\mathbf{E}}_{S\sim D^m} \left(\tfrac{R_S(h)}{R(h)}\right)^{mR_S(h)} \left(\tfrac{1-R_S(h)}{1-R(h)}\right)^{m(1-R_S(h))} \\
&= \mathop{\mathbf{E}}_{h\sim P} \textstyle\sum_{k=0}^{m} \Pr\limits_{S\sim D^m}\left(R_S(h)=\frac{k}{m}\right)\left(\frac{\frac{k}{m}}{R(h)}\right)^{k}\left(\frac{1-\frac{k}{m}}{1-R(h)}\right)^{m-k} \\
&= \textstyle\sum_{k=0}^{m} \binom{m}{k} (k/m)^k (1-k/m)^{m-k}\,, \qquad (1) \\
&\leqslant 2\sqrt{m}\,.
\end{aligned}
$$

□

- Note that, in Line (1) of the proof, $\Pr\limits_{S\sim D^m}\left(R_S(h) = \frac{k}{m}\right)$ is replaced by the probability mass function of the binomial.

Proof of the Langford/Seeger bound

Follows immediately from General Theorem by choosing $\Delta(q,p) = \mathrm{kl}(q,p)$.

- Indeed, in that case we have

$$
\begin{aligned}
\mathop{\mathbf{E}}_{S\sim D^m} \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta(R_S(h),R(h))} &= \mathop{\mathbf{E}}_{h\sim P} \mathop{\mathbf{E}}_{S\sim D^m} \left(\frac{R_S(h)}{R(h)}\right)^{mR_S(h)} \left(\frac{1-R_S(h)}{1-R(h)}\right)^{m(1-R_S(h))} \\
&= \mathop{\mathbf{E}}_{h\sim P} \sum_{k=0}^{m} \Pr_{S\sim D^m}\left(R_S(h)=\tfrac{k}{m}\right) \left(\frac{\frac{k}{m}}{R(h)}\right)^{k} \left(\frac{1-\frac{k}{m}}{1-R(h)}\right)^{m-k} \\
&= \sum_{k=0}^{m} \binom{m}{k} (k/m)^k (1-k/m)^{m-k}, \qquad (1) \\
&\leqslant 2\sqrt{m}\,.
\end{aligned}
$$

□

- Note that, in Line (1) of the proof, $\Pr_{S\sim D^m}\left(R_S(h) = \frac{k}{m}\right)$ is replaced by the probability mass function of the binomial.
- This is **only true if** the examples of S are drawn iid. (i.e., $S \sim D^m$)

Proof of the Langford/Seeger bound

Follows immediately from General Theorem by choosing $\Delta(q,p) = \mathrm{kl}(q,p)$.

- Indeed, in that case we have

$$
\begin{aligned}
\mathop{\mathbf{E}}_{S\sim D^m} \mathop{\mathbf{E}}_{h\sim P} e^{m\Delta(R_S(h),R(h))} &= \mathop{\mathbf{E}}_{h\sim P} \mathop{\mathbf{E}}_{S\sim D^m} \left(\frac{R_S(h)}{R(h)}\right)^{mR_S(h)} \left(\frac{1-R_S(h)}{1-R(h)}\right)^{m(1-R_S(h))} \\
&= \mathop{\mathbf{E}}_{h\sim P} \sum_{k=0}^{m} \Pr_{S\sim D^m}\left(R_S(h)=\tfrac{k}{m}\right) \left(\frac{\frac{k}{m}}{R(h)}\right)^{k} \left(\frac{1-\frac{k}{m}}{1-R(h)}\right)^{m-k} \\
&= \sum_{k=0}^{m} \binom{m}{k} (k/m)^k (1-k/m)^{m-k}\,, \qquad (1) \\
&\leqslant 2\sqrt{m}\,.
\end{aligned}
$$

□

- Note that, in Line (1) of the proof, $\Pr_{S\sim D^m}\left(R_S(h) = \frac{k}{m}\right)$ is replaced by the probability mass function of the binomial.
- This is **only true if** the examples of S are drawn iid. (i.e., $S \sim D^m$)
- So this result is no longer valid in the non iid case, even if General Theorem is.

Linear classifiers

- We will choose the prior and posterior distributions to be Gaussians with unit variance.

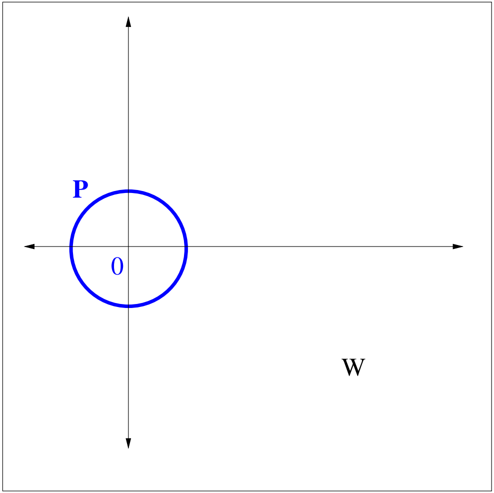
Linear classifiers

- We will choose the prior and posterior distributions to be Gaussians with unit variance.
- The prior P will be centered at the origin with unit variance

Linear classifiers

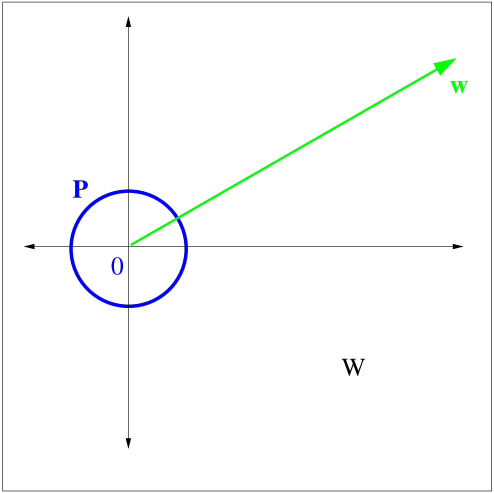
- We will choose the prior and posterior distributions to be Gaussians with unit variance.
- The prior P will be centered at the origin with unit variance
- The specification of the centre for the posterior $Q(\mathbf{w}, \mu)$ will be by a unit vector $\mathbf{w}$ and a scale factor μ.

PAC-Bayes Bound for SVM (1/2)



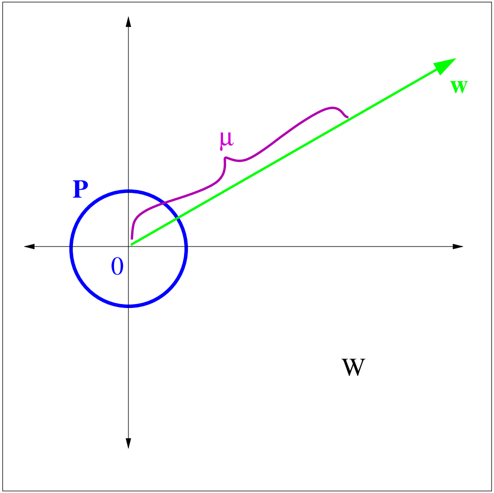
- **Prior** P is Gaussian $\mathcal{N}(0, 1)$
-
-
-

PAC-Bayes Bound for SVM (1/2)



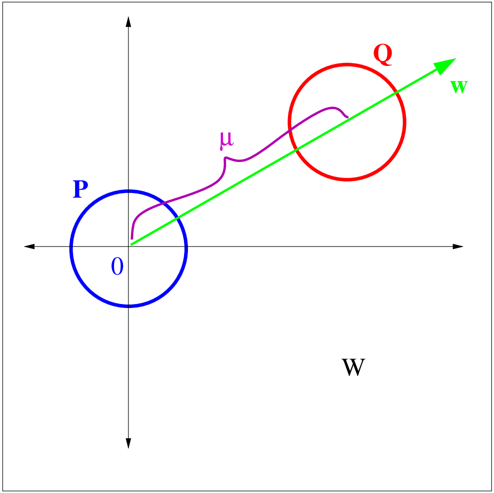
- **Prior** *P* is Gaussian $\mathcal{N}(0, 1)$
- Posterior is in the **direction w**
-
-

PAC-Bayes Bound for SVM (1/2)



- **Prior** P is Gaussian $\mathcal{N}(0, 1)$
- Posterior is in the **direction w**
- at **distance** μ from the origin
-

PAC-Bayes Bound for SVM (1/2)



- **Prior** P is Gaussian $\mathcal{N}(0, 1)$
- Posterior is in the **direction** $\mathbf{w}$
- at **distance** μ from the origin
- **Posterior** Q is Gaussian

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}\left(\hat{Q}_S(\mathbf{w}, \mu) \,\middle\|\, \boxed{Q_{\mathcal{D}}(\mathbf{w}, \mu)}\right) \leqslant \frac{\mathrm{KL}(P \| Q(\mathbf{w}, \mu)) + \ln \frac{m+1}{\delta}}{m}$$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\,\|\,\boxed{Q_{\mathcal{D}}(\mathbf{w},\mu)}) \leqslant \frac{\mathrm{KL}(P\|Q(\mathbf{w},\mu)) + \ln\frac{m+1}{\delta}}{m}$$

- $Q_{\mathcal{D}}(\mathbf{w},\mu)$ true performance of the stochastic classifier

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu) \| \boxed{Q_{\mathcal{D}}(\mathbf{w},\mu)}) \leqslant \frac{\mathrm{KL}(P\|Q(\mathbf{w},\mu)) + \ln\frac{m+1}{\delta}}{m}$$

- $Q_{\mathcal{D}}(\mathbf{w},\mu)$ true performance of the stochastic classifier
- SVM is deterministic classifier that exactly corresponds to $\operatorname{sgn}\left(\mathbb{E}_{c\sim Q(\mathbf{w},\mu)}[c(\mathbf{x})]\right)$ as centre of the Gaussian gives the same classification as halfspace with more weight.

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|\boxed{Q_{\mathcal{D}}(\mathbf{w},\mu)}) \leqslant \frac{\mathrm{KL}(P\|Q(\mathbf{w},\mu)) + \ln\frac{m+1}{\delta}}{m}$$

- $Q_{\mathcal{D}}(\mathbf{w},\mu)$ true performance of the stochastic classifier
- SVM is deterministic classifier that exactly corresponds to $\mathrm{sgn}\left(\mathbb{E}_{c\sim Q(\mathbf{w},\mu)}[c(\mathbf{x})]\right)$ as centre of the Gaussian gives the same classification as halfspace with more weight.
- Hence its error bounded by $2Q_{\mathcal{D}}(\mathbf{w},\mu)$, since as observed above if $\mathbf{x}$ misclassified at least half of $c \sim Q$ err.

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}\left(\boxed{\hat{Q}_S(\mathbf{w}, \mu)} \middle\| Q_{\mathcal{D}}(\mathbf{w}, \mu) \right) \leqslant \frac{\mathrm{KL}(P \| Q(\mathbf{w}, \mu)) + \ln \frac{m+1}{\delta}}{m}$$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}\left(\boxed{\hat{Q}_S(\mathbf{w}, \mu)} \| Q_{\mathcal{D}}(\mathbf{w}, \mu)\right) \leqslant \frac{\mathrm{KL}(P \| Q(\mathbf{w}, \mu)) + \ln \frac{m+1}{\delta}}{m}$$

- $\hat{Q}_S(\mathbf{w}, \mu)$ stochastic measure of the training error

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}\left(\boxed{\hat{Q}_S(\mathbf{w},\mu)} \,\middle\|\, Q_{\mathcal{D}}(\mathbf{w},\mu)\right) \leqslant \frac{\mathrm{KL}(P \| Q(\mathbf{w},\mu)) + \ln\frac{m+1}{\delta}}{m}$$

- $\hat{Q}_S(\mathbf{w},\mu)$ stochastic measure of the training error
- $\hat{Q}_S(\mathbf{w},\mu) = \mathbb{E}_m[\tilde{F}(\mu\gamma(\mathbf{x},y))]$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}\left(\boxed{\hat{Q}_S(\mathbf{w}, \mu)} \| Q_{\mathcal{D}}(\mathbf{w}, \mu)\right) \leqslant \frac{\mathrm{KL}(P \| Q(\mathbf{w}, \mu)) + \ln \frac{m+1}{\delta}}{m}$$

- $\hat{Q}_S(\mathbf{w}, \mu)$ stochastic measure of the training error
- $\hat{Q}_S(\mathbf{w}, \mu) = \mathbb{E}_m[\tilde{F}(\mu\gamma(\mathbf{x}, y))]$
- $\gamma(\mathbf{x}, y) = (y\mathbf{w}^T\phi(\mathbf{x}))/(\|\phi(\mathbf{x})\|\|\mathbf{w}\|)$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}\left(\boxed{\hat{Q}_S(\mathbf{w},\mu)}\,\|\,Q_{\mathcal{D}}(\mathbf{w},\mu)\right) \leqslant \frac{\mathrm{KL}(P\|Q(\mathbf{w},\mu)) + \ln\frac{m+1}{\delta}}{m}$$

- $\hat{Q}_S(\mathbf{w},\mu)$ stochastic measure of the training error
- $\hat{Q}_S(\mathbf{w},\mu) = \mathbb{E}_m[\tilde{F}(\mu\gamma(\mathbf{x},y))]$
- $\gamma(\mathbf{x},y) = (y\mathbf{w}^T\phi(\mathbf{x}))/(\|\phi(\mathbf{x})\|\|\mathbf{w}\|)$
- $\tilde{F}(t) = 1 - \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t} e^{-x^2/2}\mathrm{d}x$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{\boxed{\mathrm{KL}(P\|Q(\mathbf{w},\mu))} + \ln\frac{m+1}{\delta}}{m}$$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w}, \mu) \| Q_{\mathcal{D}}(\mathbf{w}, \mu)) \leqslant \frac{\boxed{\mathrm{KL}(P \| Q(\mathbf{w}, \mu))} + \ln \frac{m+1}{\delta}}{m}$$

- Prior $P \equiv$ Gaussian centered on the origin

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu) \| Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{\boxed{\mathrm{KL}(P \| Q(\mathbf{w},\mu))} + \ln \frac{m+1}{\delta}}{m}$$

- Prior $P \equiv$ Gaussian centered on the origin
- Posterior $Q \equiv$ Gaussian along $\mathbf{w}$ at a distance μ from the origin

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{\boxed{\mathrm{KL}(P\|Q(\mathbf{w},\mu))} + \ln\frac{m+1}{\delta}}{m}$$

- Prior $P \equiv$ Gaussian centered on the origin
- Posterior $Q \equiv$ Gaussian along **w** at a distance μ from the origin
- $\mathrm{KL}(P\|Q) = \mu^2/2$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{\mathrm{KL}(P\|Q(\mathbf{w},\mu)) + \ln\frac{m+1}{\boxed{\delta}}}{m}$$

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{\mathrm{KL}(P\|Q(\mathbf{w},\mu)) + \ln\frac{m+1}{\boxed{\delta}}}{m}$$

- δ is the confidence

PAC-Bayes Bound for SVM (2/2)

Linear classifiers performance may be bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w}, \mu) \| Q_{\mathcal{D}}(\mathbf{w}, \mu)) \leqslant \frac{\mathrm{KL}(P \| Q(\mathbf{w}, \mu)) + \ln \frac{m+1}{\boxed{\delta}}}{m}$$

- δ is the confidence
- The bound holds with probability $1 - \delta$ over the random i.i.d. selection of the training data.

Form of the SVM bound

- Note that bound holds for all posterior distributions so that we can choose μ to optimise the bound

Form of the SVM bound

- Note that bound holds for all posterior distributions so that we can choose μ to optimise the bound
- If we define the inverse of the KL by

$$\mathrm{KL}^{-1}(q, A) = \max\{p : \mathrm{KL}(q \| p) \leqslant A\}$$

then have with probability at least $1 - \delta$

$$Pr\left(\langle \mathbf{w}, \phi(\mathbf{x})\rangle \neq y\right) \leqslant 2 \min_{\mu} \mathrm{KL}^{-1}\left(\mathbb{E}_m[\tilde{F}(\mu\gamma(\mathbf{x}, y))], \frac{\mu^2/2 + \ln \frac{m+1}{\delta}}{m}\right)$$

Gives SVM Optimisation

- Primal form:

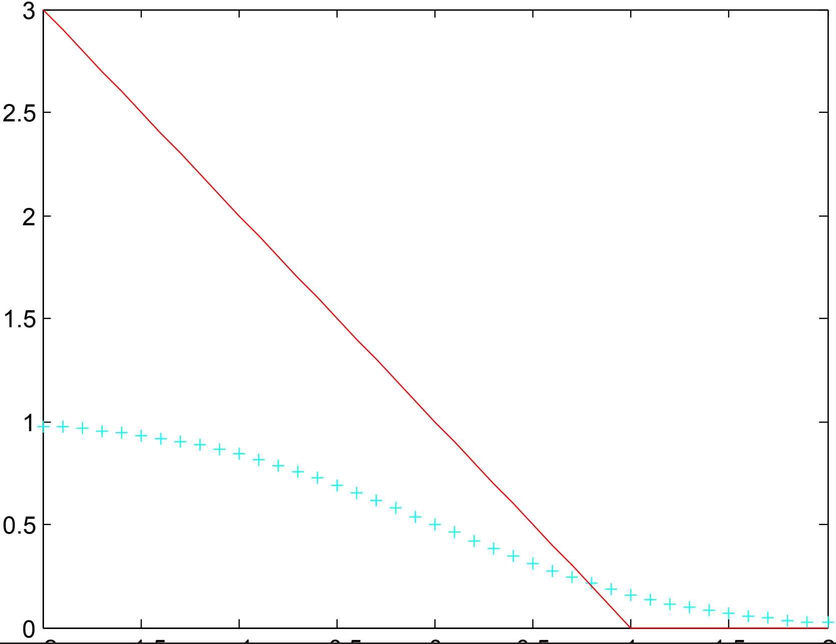
$$\begin{aligned} & \min_{\mathbf{w},\xi_i} \left[\tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \xi_i\right] & \\ \text{s.t.} \quad & y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geqslant 1 - \xi_i & i = 1, \ldots, m \\ & \xi_i \geqslant 0 & i = 1, \ldots, m \end{aligned}$$

- Dual form:

$$\begin{aligned} & \max_\alpha \left[\sum_{i=1}^m \alpha_i - \tfrac{1}{2}\sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j)\right] \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, m \end{aligned}$$

where $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ and $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle = \sum_{i=1}^m \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x})$.

Slack variable conversion



Data- or distribution-dependent priors

- An important component in the PAC-Bayes analysis is the choice of the **prior distribution**

Data- or distribution-dependent priors

- An important component in the PAC-Bayes analysis is the choice of the **prior distribution**
- The results hold whatever the choice of prior, provided that it is chosen *before* seeing the data sample

Data- or distribution-dependent priors

- An important component in the PAC-Bayes analysis is the choice of the **prior distribution**
- The results hold whatever the choice of prior, provided that it is chosen *before* seeing the data sample
- Are there ways we can choose a ‘better’ prior?

Data- or distribution-dependent priors

- An important component in the PAC-Bayes analysis is the choice of the **prior distribution**
- The results hold whatever the choice of prior, provided that it is chosen *before* seeing the data sample
- Are there ways we can choose a ‘better’ prior?
- Will explore:

Data- or distribution-dependent priors

- An important component in the PAC-Bayes analysis is the choice of the **prior distribution**
- The results hold whatever the choice of prior, provided that it is chosen *before* seeing the data sample
- Are there ways we can choose a ‘better’ prior?
- Will explore:
    - using part of the data to *learn the prior* for SVMs, but also more interestingly and more generally

Data- or distribution-dependent priors

- An important component in the PAC-Bayes analysis is the choice of the **prior distribution**
- The results hold whatever the choice of prior, provided that it is chosen *before* seeing the data sample
- Are there ways we can choose a ‘better’ prior?
- Will explore:
  - using part of the data to *learn the prior* for SVMs, but also more interestingly and more generally
  - defining the prior in terms of the *data generating distribution* *(aka localised PAC-Bayes)*.

Learning the prior (1/3)

- Bound depends on the **distance between prior and posterior**

Learning the prior (1/3)

- Bound depends on the **distance between prior and posterior**
- Better prior (closer to posterior) would lead to **tighter bound**

Learning the prior (1/3)

- Bound depends on the **distance between prior and posterior**
- Better prior (closer to posterior) would lead to **tighter bound**
- **Learn** the prior P with part of the data

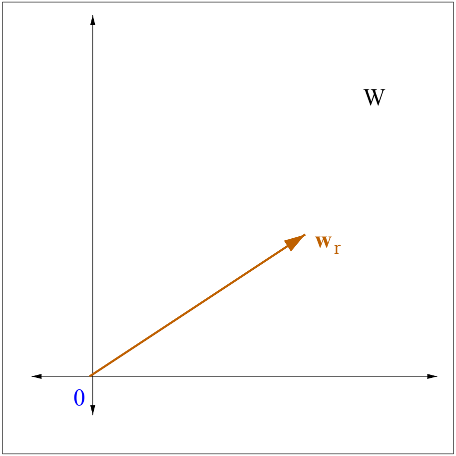
Learning the prior (1/3)

- Bound depends on the **distance between prior and posterior**
- Better prior (closer to posterior) would lead to **tighter bound**
- **Learn** the prior P with part of the data
- Introduce the learnt prior **in the bound**

Learning the prior (1/3)

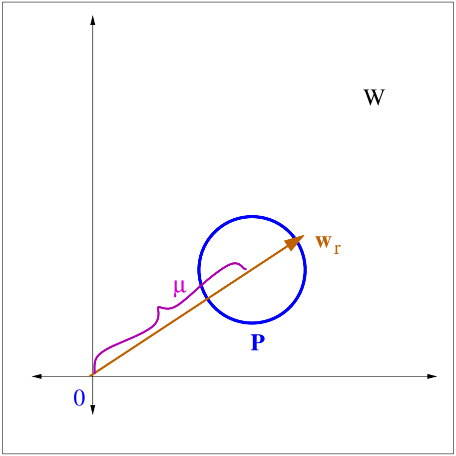
- Bound depends on the **distance between prior and posterior**
- Better prior (closer to posterior) would lead to **tighter bound**
- **Learn** the prior P with part of the data
- Introduce the learnt prior **in the bound**
- Compute stochastic error with **remaining data**

New prior for the SVM (3/3)



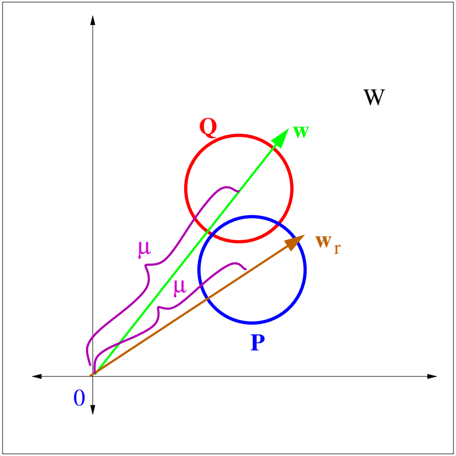
- Solve SVM with **subset of patterns**
-
-
-

New prior for the SVM (3/3)



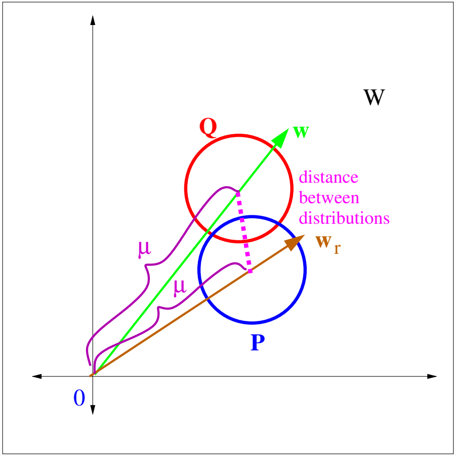
- Solve SVM with **subset of patterns**
- Prior in the **direction $\mathbf{w}_r$**
-
-

New prior for the SVM (3/3)



- Solve SVM with **subset of patterns**
- Prior in the **direction** $\mathbf{w}_r$
- **Posterior** like PAC-Bayes Bound
-

New prior for the SVM (3/3)



- Solve SVM with **subset of patterns**
- Prior in the **direction** $\mathbf{w}_r$
- **Posterior** like PAC-Bayes Bound
- **New bound** depends on $\mathrm{KL}(P\|Q)$

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu) \,\|\, \boxed{Q_{\mathcal{D}}(\mathbf{w},\mu)}) \leqslant \frac{0.5\|\mu\mathbf{w} - \eta\mathbf{w}_r\|^2 + \ln\frac{(m-r+1)J}{\delta}}{m-r}$$

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}\left(\hat{Q}_S(\mathbf{w},\mu)\,\middle\|\,\boxed{Q_{\mathcal{D}}(\mathbf{w},\mu)}\right) \leqslant \frac{0.5\|\mu\mathbf{w}-\eta\mathbf{w}_r\|^2 + \ln\frac{(m-r+1)J}{\delta}}{m-r}$$

- $Q_{\mathcal{D}}(\mathbf{w},\mu)$ true performance of the classifier

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}\left(\boxed{\hat{Q}_S(\mathbf{w},\mu)} \,\middle\|\, Q_{\mathcal{D}}(\mathbf{w},\mu)\right) \leqslant \frac{0.5\|\mu\mathbf{w} - \eta\mathbf{w}_r\|^2 + \ln\frac{(m-r+1)J}{\delta}}{m-r}$$

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}(\boxed{\hat{Q}_S(\mathbf{w}, \mu)} \| Q_{\mathcal{D}}(\mathbf{w}, \mu)) \leqslant \frac{0.5\|\mu\mathbf{w} - \eta\mathbf{w}_r\|^2 + \ln \frac{(m-r+1)J}{\delta}}{m-r}$$

- $\hat{Q}_S(\mathbf{w}, \mu)$ stochastic measure of the training error on remaining data

$$\hat{Q}(\mathbf{w}, \mu)_S = \mathbb{E}_{m-r}[\tilde{F}(\mu\gamma(\mathbf{x}, y))]$$

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{\boxed{0.5\|\mu\mathbf{w}-\eta\mathbf{w}_r\|^2} + \ln\frac{(m-r+1)J}{\delta}}{m-r}$$

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{\boxed{0.5\|\mu\mathbf{w}-\eta\mathbf{w}_r\|^2} + \ln\frac{(m-r+1)J}{\delta}}{m-r}$$

- $0.5\|\mu\mathbf{w}-\eta\mathbf{w}_r\|^2$ distance between prior and posterior

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{0.5\|\mu\mathbf{w}-\eta\mathbf{w}_r\|^2 + \ln\frac{(m-r+1)J}{\delta}}{\boxed{m-r}}$$

New Bound for the SVM (2/3)

SVM performance may be **tightly** bounded by

$$\mathrm{KL}(\hat{Q}_S(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{0.5\|\mu\mathbf{w}-\eta\mathbf{w}_r\|^2+\ln\frac{(m-r+1)J}{\delta}}{\boxed{m-r}}$$

- Penalty term only dependent on the remaining data $m-r$

Prior-SVM

- New bound proportional to $\|\mu \mathbf{w} - \eta \mathbf{w}_r\|^2$

Prior-SVM

- New bound proportional to $\|\mu \mathbf{w} - \eta \mathbf{w}_r\|^2$
- Classifier that **optimises the bound**

Prior-SVM

- New bound proportional to $\|\mu \mathbf{w} - \eta \mathbf{w}_r\|^2$
- Classifier that **optimises the bound**
- Optimisation problem to determine the **p-SVM**

$$
\begin{array}{lcr}
 & \min_{\mathbf{w},\xi_i} \left[\frac{1}{2}\|\mathbf{w} - \mathbf{w}_r\|^2 + C\sum_{i=1}^{m-r} \xi_i\right] & \\
\text{s.t.} & y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geqslant 1 - \xi_i & i = 1, \dots, m-r \\
 & \xi_i \geqslant 0 & i = 1, \dots, m-r
\end{array}
$$

Prior-SVM

- New bound proportional to $\|\mu\mathbf{w} - \eta\mathbf{w}_r\|^2$
- Classifier that **optimises the bound**
- Optimisation problem to determine the **p-SVM**

$$
\begin{aligned}
& \min_{\mathbf{w},\xi_i} \left[\tfrac{1}{2}\|\mathbf{w} - \mathbf{w}_r\|^2 + C\sum_{i=1}^{m-r}\xi_i\right] & \\
\text{s.t.} \quad & y_i\mathbf{w}^T\phi(\mathbf{x}_i) \geqslant 1 - \xi_i & i = 1,\dots,m-r \\
& \xi_i \geqslant 0 & i = 1,\dots,m-r
\end{aligned}
$$

- The p-SVM is only solved with the **remaining points**

Bound for p-SVM

1. Determine the **prior** with a subset of the training examples to obtain $\mathbf{w}_r$

Bound for p-SVM

1. Determine the **prior** with a subset of the training examples to obtain $\mathbf{w}_r$
2. Solve **p-SVM** and obtain $\mathbf{w}$

Bound for p-SVM

1. Determine the **prior** with a subset of the training examples to obtain $\mathbf{w}_r$
2. Solve **p-SVM** and obtain $\mathbf{w}$
3. **Margin** for the stochastic classifier $\hat{Q}_s$

$$\gamma(\mathbf{x}_j, y_j) = \frac{y_j \mathbf{w}^T \phi(\mathbf{x}_j)}{\|\phi(\mathbf{x}_j)\| \|\mathbf{w}\|} \qquad j = 1, \ldots, m - r$$

Bound for p-SVM

1. Determine the **prior** with a subset of the training examples to obtain $\mathbf{w}_r$
2. Solve **p-SVM** and obtain $\mathbf{w}$
3. **Margin** for the stochastic classifier $\hat{Q}_s$

$$\gamma(\mathbf{x}_j, y_j) = \frac{y_j \mathbf{w}^T \phi(\mathbf{x}_j)}{\|\phi(\mathbf{x}_j)\| \|\mathbf{w}\|} \qquad j = 1, \ldots, m - r$$

4. **Linear search** to obtain the optimal value of μ. This introduces an insignificant extra penalty term

Bound for η-prior-SVM

- Prior is elongated along the line of $\mathbf{w}_r$ but spherical with variance 1 in other directions

Bound for η-prior-SVM

- Prior is elongated along the line of $\mathbf{w}_r$ but spherical with variance 1 in other directions
- Posterior again on the line of $\mathbf{w}$ at a distance μ chosen to optimise the bound.

Bound for η-prior-SVM

- Prior is elongated along the line of $\mathbf{w}_r$ but spherical with variance 1 in other directions
- Posterior again on the line of $\mathbf{w}$ at a distance μ chosen to optimise the bound.
- Resulting bound depends on a benign parameter τ determining the variance in the direction $\mathbf{w}_r$

$$\mathrm{KL}(\hat{Q}_{S\setminus R}(\mathbf{w},\mu)\|Q_{\mathcal{D}}(\mathbf{w},\mu)) \leqslant \frac{0.5(\ln(\tau^2)+\tau^{-2}-1+P^{\|}_{\mathbf{w}_r}(\mu\mathbf{w}-\mathbf{w}_r)^2/\tau^2+P^{\perp}_{\mathbf{w}_r}(\mu\mathbf{w})^2)+\ln(\frac{m-r+1}{\delta})}{m-r}$$

η-Prior-SVM

- Consider using a prior distribution P that is elongated in the direction of $\mathbf{w}_r$

η-Prior-SVM

- Consider using a prior distribution P that is elongated in the direction of $\mathbf{w}_r$
- This will mean that there is low penalty for large projections onto this direction

η-Prior-SVM

- Consider using a prior distribution P that is elongated in the direction of $\mathbf{w}_r$
- This will mean that there is low penalty for large projections onto this direction
- Translates into an optimisation:

$$\min_{\mathbf{v},\eta,\xi_i} \left[\frac{1}{2}\|\mathbf{v}\|^2 + C\sum_{i=1}^{m-r} \xi_i \right]$$

η-Prior-SVM

- Consider using a prior distribution P that is elongated in the direction of $\mathbf{w}_r$
- This will mean that there is low penalty for large projections onto this direction
- Translates into an optimisation:

$$\min_{\mathbf{v},\eta,\xi_i} \left[\frac{1}{2}\|\mathbf{v}\|^2 + C \sum_{i=1}^{m-r} \xi_i \right]$$

- subject to

$$\begin{aligned} y_i(\mathbf{v} + \eta \mathbf{w}_r)^T \phi(\mathbf{x}_i) &\geqslant 1 - \xi_i \qquad i = 1, \ldots, m - r \\ \xi_i &\geqslant 0 \qquad i = 1, \ldots, m - r \end{aligned}$$

Model Selection with the new bound: setup

- Comparison of 10-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound

Model Selection with the new bound: setup

- Comparison of 10-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets

Model Selection with the new bound: setup

- Comparison of 10-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets
- Select C and σ that lead to minimum Classification Error (CE)

Model Selection with the new bound: setup

- Comparison of 10-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets
- Select C and σ that lead to minimum Classification Error (CE)
  - For 10-F XV select the pair that minimize the validation error

Model Selection with the new bound: setup

- Comparison of 10-fold Xvalidation, PAC-Bayes Bound and the Prior PAC-Bayes Bound
- UCI datasets
- Select C and σ that lead to minimum Classification Error (CE)
  - For 10-F XV select the pair that minimize the validation error
  - For PAC-Bayes Bound and Prior PAC-Bayes Bound select the pair that minimize the bound

Results

		Classifier					
		SVM				ηPrior SVM	
Problem		2FCV	10FCV	PAC	PrPAC	PrPAC	τ-PrPAC
digits	Bound	–	–	0.175	0.107	0.050	**0.047**
	TE	**0.007**	**0.007**	**0.007**	0.014	0.010	0.009
waveform	Bound	–	–	0.203	0.185	0.178	**0.176**
	TE	0.090	0.086	**0.084**	0.088	0.087	0.086
pima	Bound	–	–	0.424	0.420	0.428	**0.416**
	TE	0.244	0.245	**0.229**	**0.229**	0.233	0.233
ringnorm	Bound	–	–	0.203	0.110	0.053	**0.050**
	TE	**0.016**	**0.016**	0.018	0.018	**0.016**	**0.016**
spam	Bound	–	–	0.254	0.198	0.186	**0.178**
	TE	0.066	**0.063**	0.067	0.077	0.070	0.072
Average	TE	0.0846	0.0834	**0.081**	0.0852	0.0832	0.0832

Take home messages

- Bounds are remarkably tight: for final column average factor between bound and TE is under 3.

Take home messages

- Bounds are remarkably tight: for final column average factor between bound and TE is under 3.
- Model selection from the bounds is as good as 10FCV: in fact all but one of the PAC-Bayes model selections give better averages for TE.

Take home messages

- Bounds are remarkably tight: for final column average factor between bound and TE is under 3.
- Model selection from the bounds is as good as 10FCV: in fact all but one of the PAC-Bayes model selections give better averages for TE.
- The better bounds do not appear to give better model selection - best model selection is from the simplest bound.

  - A. Ambroladze, E. Parrado-Hernández, and J. Shawe-Taylor. Tighter PAC-Bayes bounds. In *Advances in Neural Information Processing Systems* 18, (2006) Pages 9-16.
  - P. Germain, A. Lacasse, F. Laviolette and M. Marchand. PAC-Bayesian learning of linear classifiers, in *Proceedings of the 26nd International Conference on Machine Learning* (ICML'09, Montréal, Canada.). ACM Press (2009), 382, Pages 453-460.

Distribution-defined priors

Distribution-defined priors

- Consider P and Q are Gibbs-Boltzmann distributions

$$P(h) := \frac{1}{Z'} e^{-\gamma \operatorname{risk}(h)} \qquad Q(h) := \frac{1}{Z} e^{-\gamma \hat{\operatorname{risk}}_S(h)}$$

Distribution-defined priors

- Consider P and Q are Gibbs-Boltzmann distributions

$$P(h) := \frac{1}{Z'} e^{-\gamma \, \mathrm{risk}(h)} \qquad Q(h) := \frac{1}{Z} e^{-\gamma \, \hat{\mathrm{risk}}_S(h)}$$

- These distributions are hard to work with since we cannot apply the bound to a single weight vector, but the bounds can be very tight:

$$KL_+(\hat{Q}_S(\gamma) \| Q_{\mathcal{D}}(\gamma)) \leqslant \frac{1}{m}\left(\frac{\gamma}{\sqrt{m}} \sqrt{\ln \frac{8\sqrt{m}}{\delta}} + \frac{\gamma^2}{4m} + \ln \frac{4\sqrt{m}}{\delta} \right)$$

with the only uncertainty the dependence on γ.

Distribution-defined priors

- Consider P and Q are Gibbs-Boltzmann distributions

$$P(h) := \frac{1}{Z'} e^{-\gamma \operatorname{risk}(h)} \qquad Q(h) := \frac{1}{Z} e^{-\gamma \hat{\operatorname{risk}}_S(h)}$$

- These distributions are hard to work with since we cannot apply the bound to a single weight vector, but the bounds can be very tight:

$$KL_+(\hat{Q}_S(\gamma) \| Q_{\mathcal{D}}(\gamma)) \leqslant \frac{1}{m}\left(\frac{\gamma}{\sqrt{m}} \sqrt{\ln \frac{8\sqrt{m}}{\delta}} + \frac{\gamma^2}{4m} + \ln \frac{4\sqrt{m}}{\delta} \right)$$

with the only uncertainty the dependence on γ.

- O. Catoni. A PAC-Bayesian approach to adaptive classification. Preprint n.840, Laboratoire de Probabilités et Modèles Aléatoires, Universités Paris 6 and Paris 7, 2003.
- G. Lever, F. Laviolette, J. Shawe-Taylor. Distribution-Dependent PAC-Bayes Priors. Proceedings of the 21st International Conference on Algorithmic Learning Theory (ALT 2010), 119-133.

Observations

- We cannot compute the prior distribution P or even sample from it:

Observations

- We cannot compute the prior distribution P or even sample from it:
  - Note that this would not be possible to consider in normal Bayesian inference;

Observations

- We cannot compute the prior distribution P or even sample from it:
  - Note that this would not be possible to consider in normal Bayesian inference;
  - Trick here is that the error measures only depend on the posterior Q, while the bound depends on KL between posterior and prior: an estimate of this KL is made without knowing the prior explicitly

Observations

- We cannot compute the prior distribution P or even sample from it:
  - Note that this would not be possible to consider in normal Bayesian inference;
  - Trick here is that the error measures only depend on the posterior Q, while the bound depends on KL between posterior and prior: an estimate of this KL is made without knowing the prior explicitly
- the Gibbs distributions are hard to sample from so not easy to work with this bound.

Other distribution defined priors

- An alternative distribution defined prior for an SVM is to place symmetrical Gaussian at the weight vector: $\mathbf{w}_p = \mathbb{E}_{(\mathbf{x},y)\sim D}(y\,\boldsymbol{\phi}(\mathbf{x}))$ to give distributions that are easier to work with, but results not impressive...

Other distribution defined priors

- An alternative distribution defined prior for an SVM is to place symmetrical Gaussian at the weight vector: $\mathbf{w}_p = \mathbb{E}_{(\mathbf{x},y)\sim D}(y\,\boldsymbol{\phi}(\mathbf{x}))$ to give distributions that are easier to work with, but results not impressive...
- What if we were to take the expected weight vector returned from a random training set of size m: then the KL between posterior and prior is related to the concentration of weight vectors from different training sets

Other distribution defined priors

- An alternative distribution defined prior for an SVM is to place symmetrical Gaussian at the weight vector: $\mathbf{w}_p = \mathbb{E}_{(\mathbf{x},y)\sim D}(y\,\boldsymbol{\phi}(\mathbf{x}))$ to give distributions that are easier to work with, but results not impressive...
- What if we were to take the expected weight vector returned from a random training set of size m: then the KL between posterior and prior is related to the concentration of weight vectors from different training sets
- This is connected to stability...

Outline

- stability

Stability

Uniform hypothesis sensitivity β at sample size m:

$$\| \mathsf{A}(z_{1:m}) - \mathsf{A}(z'_{1:m}) \| \leqslant \beta \sum_{i=1}^{m} \mathbf{1}[z_i \neq z'_i]$$

$(z_1, \ldots, z_m)$

- $\mathsf{A}(z_{1:m}) \in \mathcal{H}$ normed space
- $w_m = \mathsf{A}(z_{1:m})$ 'weight vector'

$(z'_1, \ldots, z'_m)$

- Lipschitz
- smoothness

Uniform loss sensitivity β at sample size m:

$$|\ell(\mathsf{A}(z_{1:m}), z) - \ell(\mathsf{A}(z'_{1:m}), z)| \leqslant \beta \sum_{i=1}^{m} \mathbf{1}[z_i \neq z'_i]$$

- worst-case
- data-insensitive
- distribution-insensitive
- Open: data-dependent?

Generalization from Stability

If A has sensitivity β at sample size m, then for any $\delta \in (0, 1)$,

w.p. $\geqslant 1 - \delta$, $\quad R_{\mathrm{out}}(h) \leqslant R_{\mathrm{in}}(h) + \epsilon(\beta, m, \delta)$

Generalization from Stability

If A has sensitivity β at sample size m, then for any $\delta \in (0, 1)$,

w.p. $\geqslant 1 - \delta$, $\quad R_{\mathrm{out}}(h) \leqslant R_{\mathrm{in}}(h) + \epsilon(\beta, m, \delta)$

(e.g. Bousquet & Elisseeff)

Generalization from Stability

If A has sensitivity β at sample size m, then for any $\delta \in (0,1)$,

w.p. $\geqslant 1-\delta$, $\quad R_{\text{out}}(h) \leqslant R_{\text{in}}(h) + \epsilon(\beta, m, \delta)$

(e.g. Bousquet & Elisseeff)

- the intuition is that if individual examples do not affect the loss of an algorithm then it will be concentrated

Generalization from Stability

If A has sensitivity β at sample size m, then for any $\delta \in (0, 1)$,

w.p. $\geqslant 1 - \delta$, $\quad R_{\text{out}}(h) \leqslant R_{\text{in}}(h) + \epsilon(\beta, m, \delta)$

(e.g. Bousquet & Elisseeff)

- the intuition is that if individual examples do not affect the loss of an algorithm then it will be concentrated
- can be applied to kernel methods where β is related to the regularisation constant, but bounds are quite weak

Generalization from Stability

If A has sensitivity β at sample size m, then for any $\delta \in (0, 1)$,

w.p. $\geqslant 1 - \delta$, $\quad R_{\text{out}}(h) \leqslant R_{\text{in}}(h) + \epsilon(\beta, m, \delta)$

(e.g. Bousquet & Elisseeff)

- the intuition is that if individual examples do not affect the loss of an algorithm then it will be concentrated
- can be applied to kernel methods where β is related to the regularisation constant, but bounds are quite weak
- question: algorithm output is highly concentrated
  $\Longrightarrow$ stronger results?

Stability + PAC-Bayes I

If A has uniform hypothesis stability β at sample size n, then for any $\delta \in (0,1)$, w.p. $\geqslant 1 - 2\delta$,

$$\mathrm{KL}\big(R_{\mathrm{in}}(Q) \| R_{\mathrm{out}}(Q)\big) \leqslant \frac{\frac{n\beta^2}{2\sigma^2}\left(1 + \sqrt{\frac{1}{2}\log\left(\frac{1}{\delta}\right)}\right)^2 + \log\left(\frac{n+1}{\delta}\right)}{n}$$

Gaussian randomization

- $P = \mathcal{N}(\mathbb{E}[W_n], \sigma^2 I)$
- $Q = \mathcal{N}(W_n, \sigma^2 I)$
- $\mathrm{KL}(Q \| P) = \frac{1}{2\sigma^2} \| W_n - \mathbb{E}[W_n] \|^2$

Main proof components:

- w.p. $\geqslant 1 - \delta$, $\quad \mathrm{KL}\big(R_{\mathrm{in}}(Q) \| R_{\mathrm{out}}(Q)\big) \leqslant \frac{\mathrm{KL}(Q \| Q_0) + \log\left(\frac{n+1}{\delta}\right)}{n}$
- w.p. $\geqslant 1 - \delta$, $\quad \| W_n - \mathbb{E}[W_n] \| \leqslant \sqrt{n}\, \beta \left(1 + \sqrt{\frac{1}{2}\log\left(\frac{1}{\delta}\right)}\right)$

Performance of deep NNs

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights
- If we have a deep network solution with a wide basin of good performance we can take a similar approach using PAC-Bayes with a broad posterior around the solution

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights
- If we have a deep network solution with a wide basin of good performance we can take a similar approach using PAC-Bayes with a broad posterior around the solution
- (Dziugaite and Roy + Neyshabur) have derived some of the tightest deep learning bounds in this way

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights
- If we have a deep network solution with a wide basin of good performance we can take a similar approach using PAC-Bayes with a broad posterior around the solution
- (Dziugaite and Roy + Neyshabur) have derived some of the tightest deep learning bounds in this way
  - by training to expand the basin of attraction

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights
- If we have a deep network solution with a wide basin of good performance we can take a similar approach using PAC-Bayes with a broad posterior around the solution
- (Dziugaite and Roy + Neyshabur) have derived some of the tightest deep learning bounds in this way
  - by training to expand the basin of attraction
  - hence not measuring good generalisation of normal training

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights
- If we have a deep network solution with a wide basin of good performance we can take a similar approach using PAC-Bayes with a broad posterior around the solution
- (Dziugaite and Roy + Neyshabur) have derived some of the tightest deep learning bounds in this way
  - by training to expand the basin of attraction
  - hence not measuring good generalisation of normal training
  - D&R have also tried to apply the Lever et al. bound but observed cannot measure generalisation correctly for deep networks as has no way of distinguishing between successful fitting of true and random labels

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights
- If we have a deep network solution with a wide basin of good performance we can take a similar approach using PAC-Bayes with a broad posterior around the solution
- (Dziugaite and Roy + Neyshabur) have derived some of the tightest deep learning bounds in this way
    - by training to expand the basin of attraction
    - hence not measuring good generalisation of normal training
    - D&R have also tried to apply the Lever et al. bound but observed cannot measure generalisation correctly for deep networks as has no way of distinguishing between successful fitting of true and random labels
- There have also been suggestions that stability of SGD is important in obtaining good generalization

Performance of deep NNs

- Deep learning has thrown down a challenge to SLT: very good performance with extremely complex hypothesis classes
- For SVMs we can think of the margin as capturing an accuracy with which we need to estimate the weights
- If we have a deep network solution with a wide basin of good performance we can take a similar approach using PAC-Bayes with a broad posterior around the solution
- (Dziugaite and Roy + Neyshabur) have derived some of the tightest deep learning bounds in this way
  - by training to expand the basin of attraction
  - hence not measuring good generalisation of normal training
  - D&R have also tried to apply the Lever et al. bound but observed cannot measure generalisation correctly for deep networks as has no way of distinguishing between successful fitting of true and random labels
- There have also been suggestions that stability of SGD is important in obtaining good generalization

Deep Network Training Experiments

Deep Network Training Experiments

- Use part of the data for training a prior (as with SVM experiments)

Deep Network Training Experiments

- Use part of the data for training a prior (as with SVM experiments)
- Use second part of data to perform an optimisation of a PAC-Bayes bound

Deep Network Training Experiments

- Use part of the data for training a prior (as with SVM experiments)
- Use second part of data to perform an optimisation of a PAC-Bayes bound
- Different ways to choose approximations to the KL term between empirical and true risk: the relaxed Pinsker inequality reads:

$$\mathrm{kl}(\hat{p}\|p) \geqslant 2(p-\hat{p})^2 \quad \text{for } \hat{p}, p \in (0,1), \qquad (f_{\mathrm{classic}}) \tag{2}$$

while the refined Pinsker inequality takes the form:

$$\mathrm{kl}(\hat{p}\|p) \geqslant \frac{(p-\hat{p})^2}{2p} \quad \text{for } \hat{p}, p \in (0,1),\ \hat{p} < p. \quad (f_{\mathrm{quad}}) \tag{3}$$

- f_λ based on the λ bound and f_{bbb} based on variational inference.

Model Selection Results

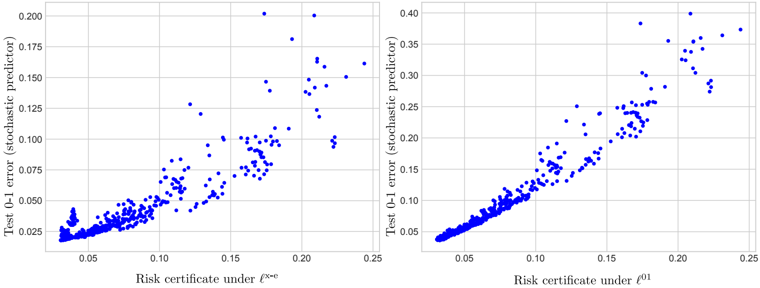


Figure: Model selection results from more than 600 runs with different hyper-parameters. The architecture used is a CNN with Gaussian data-dependent priors. We use a reduced subset of MNIST for these experiments (10% of training data).

Training and Generalisation Results

Setup			Risk cert.		Stch. pred.		Det. pred.		Ens. pred.		Prior
Arch.	Prior	Obj.	$\ell^{\text{x-e}}$	ℓ^{01}	x-e	01 err.	x-e	01 err.	x-e	01 err.	01 err.
FCN	Rand.Init. (Gaussian)	f_{quad}	.2033	.3155	.0268	.0921	.0137	.0558	.0007	.0572	.8792
		f_{lambda}	.2326	.3275	.0211	.0732	.0077	.0429	.0004	.0448	.8792
		f_{classic}	.1749	.3304	.0407	.1411	.0204	.0851	.0009	.0868	.8792
		f_{bbb}	.5163	.5516	.0088	.0293	.0038	.0172	.0003	.0178	.8792
	Learnt (Gaussian)	f_{quad}	.0146	.0279	.0084	.0202	.0032	.0186	.0002	.0189	.0202
		f_{lambda}	.0201	.0354	.0082	.0196	.0071	.0185	.0001	.0185	.0202
		f_{classic}	.0141	.0284	.0101	.0230	.0089	.0189	.0002	.0191	.0202
		f_{bbb}	.0788	.0968	.0063	.0179	.0066	.0153	.0001	.0153	.0202
	-	f_{erm}	-	-	-	-	.0101	.0152	-	-	-
CNN	Rand.Init. (Gaussian)	f_{quad}	.1453	.2165	.0143	.0513	.0062	.0257	.0003	.0261	.9478
		f_{lambda}	.1583	.2202	.0109	.0397	.0056	.0207	.0003	.0211	.9478
		f_{classic}	.1260	.2277	.0253	.0869	.0111	.0425	.0006	.0421	.9478
		f_{bbb}	.3400	.3645	.0039	.0154	.0016	.0088	.0001	.0092	.9478
	Learnt (Gaussian)	f_{quad}	.0078	.0155	.0045	.0104	.0003	.0105	.0001	.0104	.0104
		f_{lambda}	.0095	.0186	.0044	.0106	.0047	.0098	.0000	.0100	.0104
		f_{classic}	.0083	.0166	.0049	.0123	.0048	.0103	.0001	.0103	.0104
		f_{bbb}	.0447	.0538	.0040	.0104	.0043	.0082	.0002	.0082	.0104
	-	f_{erm}	-	-	-	-	.0081	.0092	-	-	-

Table: MNIST using Gaussian priors. The table includes two architectures (FCN and CNN), two priors (a data-free prior , and a data-dependent prior) and four training objectives.

Training and Generalisation Results

Setup			Risk cert.		Stch. pred.		Det. pred.		Ens. pred.		Prior
Arch.	Prior	Obj.	$\ell^{\text{x-e}}$	ℓ^{01}	x-e	01 err.	x-e	01 err.	x-e	01 err.	01 err.
CNN (9 layers)	Learnt (50% data)	f_{quad}	.1296	.3034	.0903	.2452	.0726	.2439	.0024	.2413	.2518
		f_{lambda}	.1742	.3730	.0689	.2307	.0609	.2225	.0018	.2133	.2518
		f_{classic}	.1173	.2901	.0931	.2537	.0952	.2437	.0025	.2332	.2518
		f_{bbb}	.8096	.8633	.0715	.2198	.0735	.2160	.0017	.2130	.2518
	Learnt (70% data)	f_{quad}	.1017	.2502	.0816	.2137	.0928	.2137	.0023	.2100	.2169
		f_{lambda}	.1414	.3128	.0708	.2081	.0767	.2061	.0021	.2049	.2169
		f_{classic}	.0957	.2377	.0862	.2161	.0827	.2167	.0021	.2135	.2169
		f_{bbb}	.6142	.6965	.0708	.1979	.0562	.1992	.0019	.1944	.2169
	-	f_{erm}	-	-	-	-	.1400	.1946	-	-	-
CNN (15 layers)	Learnt (50% data)	f_{quad}	.0867	.2174	.0584	.1668	.0538	.1662	.0014	.1653	.1688
		f_{lambda}	.1217	.2707	.0506	.1618	.0417	.1639	.0015	.1622	.1688
		f_{classic}	.0782	.1954	.0652	.1686	.0594	.1692	.0013	.1674	.1688
		f_{bbb}	.6069	.7066	.0468	.1553	.0412	.1530	.0012	.1517	.1688
	Learnt (70% data)	f_{quad}	.0756	.1806	.0559	.1463	.0391	.1469	.0016	.1449	.1490
		f_{lambda}	.0922	.2121	.0500	.1437	.0507	.1449	.0012	.1438	.1490
		f_{classic}	.0703	.1667	.0615	.1475	.0551	.1480	.0010	.1476	.1490
		f_{bbb}	.4481	.5572	.0455	.1413	.0395	.1405	.0008	.1409	.1490
	-	f_{erm}	-	-	-	-	.0957	.1413	-	-	-

Table: Train and test set results on CIFAR-10 using Gaussian priors, three deep CNN architectures and two percentages of data used to build the data-dependent prior (50% and 70%, i.e. 25.000 and 35.000 examples).

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 *Audibert and Bousquet [6], Catoni [9, 10], Guedj [25], Guedj and Pujol [27], Maurer [39], McAllester [41, 42, 44, 45], Mhammedi et al. [46], Seeger [51, 52], Shawe-Taylor and Williamson [56], Thiemann et al. [58]*

SVMs & linear classifiers *Germain et al. [19], Langford and Shawe-Taylor [32], McAllester [44]*

Supervised learning algorithms reinterpreted as bound minimizers *Ambroladze et al. [5], Germain et al. [22], Shawe-Taylor and Hardoon [57]*

High-dimensional regression *Alquier and Biau [1], Alquier and Lounici [2], Guedj and Robbiano [24], Guedj and Alquier [26], Li et al. [35]*

Classification *Catoni [9, 10], Lacasse et al. [30], Langford and Shawe-Taylor [32], Parrado-Hernández et al. [49]*

A flexible framework

Transductive learning, domain adaptation *Bégin et al. [7], Derbeko et al. [12], Germain et al. [20], Nozawa et al. [48]*

Non-iid or heavy-tailed data *Alquier and Guedj [3], Holland [29], Lever et al. [34], Seldin et al. [54, 55]*

Density estimation *Higgs and Shawe-Taylor [28], Seldin and Tishby [53]*

Reinforcement learning *Fard and Pineau [16], Fard et al. [17], Ghavamzadeh et al. [23], Seldin et al. [54, 55]*

Sequential learning *Gerchinovitz [18], Li et al. [36]*

Algorithmic stability, differential privacy *Dziugaite and Roy [13, 14], London [37], London et al. [38], Rivasplata et al. [50]*

Deep neural networks *Dziugaite and Roy [15], Letarte et al. [33], Neyshabur et al. [47], Zhou et al. [60]*

...

Conclusions

- One of key questions in learning is generalisation

Conclusions

- One of key questions in learning is generalisation
- Modern machine learning appears to contradict many of the conclusions of statistical learning theory

Conclusions

- One of key questions in learning is generalisation
- Modern machine learning appears to contradict many of the conclusions of statistical learning theory
- Modelling learning in a more refined way leads to bounds that overcome this contradiction and throw light on different ingredients in achieving good test performance

Conclusions

- One of key questions in learning is generalisation
- Modern machine learning appears to contradict many of the conclusions of statistical learning theory
- Modelling learning in a more refined way leads to bounds that overcome this contradiction and throw light on different ingredients in achieving good test performance
- Can drive algorithms to give improved bounds and state of the art performance

Conclusions

- One of key questions in learning is generalisation
- Modern machine learning appears to contradict many of the conclusions of statistical learning theory
- Modelling learning in a more refined way leads to bounds that overcome this contradiction and throw light on different ingredients in achieving good test performance
- Can drive algorithms to give improved bounds and state of the art performance
- Many other aspects of deep learning still remain to be captured by theoretical analysis

References I

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

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

[3] Pierre Alquier and Benjamin Guedj. Simpler PAC-Bayesian bounds for hostile data. *Machine Learning*, 107(5):887–902, 2018.

[4] Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of Gibbs posteriors. *ArXiv e-prints*, 2015. URL `http://arxiv.org/abs/1506.04091`.

[5] 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.

[6] Jean-Yves Audibert and Olivier Bousquet. Combining PAC-Bayesian and generic chaining bounds. *Journal of Machine Learning Research*, 2007.

[7] Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. PAC-Bayesian theory for transductive learning. In *AISTATS*, 2014.

[8] Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. PAC-Bayesian bounds based on the Rényi divergence. In *AISTATS*, 2016.

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

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

[11] Olivier Catoni. *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*, volume 56. Inst. of Mathematical Statistic, 2007.

[12] Philip Derbeko, Ran El-Yaniv, and Ron Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *J. Artif. Intell. Res. (JAIR)*, 22, 2004.

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

[14] 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, 2018.

References II

[15] Gintare K. Dziugaite and Daniel 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.

[16] Mahdi Milani Fard and Joelle Pineau. PAC-Bayesian model selection for reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[17] Mahdi Milani Fard, Joelle Pineau, and Csaba 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.

[18] 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.

[19] 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, ICML*, 2009.

[20] 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.

[21] Pascal Germain. *Généralisations de la théorie PAC-bayésienne pour l'apprentissage inductif, l'apprentissage transductif et l'adaptation de domaine.* PhD thesis, Université Laval, 2015.

[22] Pascal Germain, Alexandre Lacasse, Mario Marchand, Sara Shanian, and François Laviolette. From PAC-Bayes bounds to KL regularization. In *Advances in Neural Information Processing Systems*, pages 603–610, 2009.

[23] 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.

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

[25] Benjamin Guedj. A primer on PAC-Bayesian learning. *arXiv:1901.05353*, 2019. To appear in the Proceedings of the French Mathematical Society.

[26] Benjamin Guedj and Pierre Alquier. PAC-Bayesian estimation and prediction in sparse additive models. *Electron. J. Statist.*, 7: 264–291, 2013.

[27] Benjamin Guedj and Louis Pujol. Still no free lunches: the price to pay for tighter PAC-Bayes bounds. *arXiv preprint arXiv:1910.04460*, 2019.

References III

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

[29] Matthew J Holland. PAC-Bayes under potentially heavy tails. *arXiv:1905.07900*, 2019. To appear in NeurIPS.

[30] 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.

[31] John Langford and Matthias Seeger. Bounds for averaging classifiers. Technical report, Carnegie Mellon, Departement of Computer Science, 2001.

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

[33] Gaël Letarte, Pascal Germain, Benjamin Guedj, and François Laviolette. Dichotomize and Generalize: PAC-Bayesian Binary Activated Deep Neural Networks. *arXiv:1905.10259*, 2019. To appear at NeurIPS.

[34] 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.

[35] 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.

[36] Le Li, Benjamin Guedj, and Sébastien Loustau. A quasi-Bayesian perspective to online clustering. *Electron. J. Statist.*, 12(2): 3071–3113, 2018.

[37] 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.

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

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

[40] D. A. McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1):5–21, 2003.

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

References IV

[42] David McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37, 1999.

[43] David McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3), 1999.

[44] David McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51(1), 2003.

[45] David McAllester. Simplified PAC-Bayesian margin bounds. In *COLT*, 2003.

[46] Zakaria Mhammedi, Peter D. Grunwald, and Benjamin Guedj. PAC-Bayes Un-Expected Bernstein Inequality. *arXiv preprint arXiv:1905.13367*, 2019. Accepted at NeurIPS 2019.

[47] 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.

[48] Kento Nozawa, Pascal Germain, and Benjamin Guedj. PAC-Bayesian contrastive unsupervised representation learning. *arXiv preprint arXiv:1910.04464*, 2019.

[49] 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.

[50] 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.

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

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

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

[54] 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.

[55] Yevgeny Seldin, Peter Auer, François Laviolette, John Shawe-Taylor, and Ronald Ortner. PAC-Bayesian analysis of contextual bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

References V

[56] 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.

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

[58] Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A Strongly Quasiconvex PAC-Bayesian Bound. In *International Conference on Algorithmic Learning Theory, ALT*, pages 466–492, 2017.

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

[60] Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. In *ICLR*, 2019.