# Foundations of Deep Learning

# Lecture 08

# Generalization II: PAC-Bayes Bounds

Aurelien Lucchi

Fall 2024

# A new approach to generalization

First attempt at deriving a generalization bound: we first derived an upper bound for a fixed hypothesis $f$ and then apply a union bound for all $f \in \mathcal{F}$.

**This lecture:** discuss alternative approach using randomized hypotheses.

# Section 1

# General formulation PAC-Bayes bounds

# Recall problem of interest

**Hypothesis space** Set $\mathcal{F}$ of functions from $\mathcal{X} \subseteq \mathbb{R}^d$ to $\mathcal{Y} \subseteq \mathbb{R}$

**Risk** Given a data distribution $\mathcal{D}$ over $(\mathcal{X}, \mathcal{Y})$, we define the true risk as

$$R(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l((\mathbf{x}, y), f)]$$

We will mostly discuss a classification setting where $l((\mathbf{x}, y), f) = \mathbf{1}[f(\mathbf{x}) \neq y]$.

**Empirical risk** Typically do not have access to the exact data distribution but only a sample set $S = \{(\mathbf{x}_i, y_i\}_{i=1}^n \sim D \implies$ define the empirical risk as

$$R_S(f) = \sum_{(x,y)\in S} [l((\mathbf{x}, y), f)].$$

# What's a generalization bound?

The typical form of a generalization bound we are interested in is:

$$\underbrace{R(f)}_{\text{True risk}} \leq \underbrace{R_S(f)}_{\text{Empirical risk}} + \underbrace{m(\text{complexity of class of functions}, n)}_{\text{(a function that approaches 0 as n approaches infinity)}},$$

where $n$ is the number of training datapoints and $m$ is a function that measures the complexity of a class of functions.

**Generalization gap:**

$$\delta_f^S := \max(0, R(f) - R_S(f)) \leq |R(f) - R_S(f)|.$$

# Recall: Prior Distribution

**Definition:** The prior distribution, $P(\theta)$, represents our belief about the parameters $\theta$ **before observing any data**.

**Key Points:**

- Encodes domain knowledge or assumptions about the parameters.
- Examples: Gaussian, Uniform, Beta, etc.

# Recall: Posterior Distribution

**Definition:** The posterior distribution, $P(\theta \mid \mathcal{D})$, combines prior beliefs with observed data $\mathcal{D}$ to update our knowledge about $\theta$.

**Key Points:**

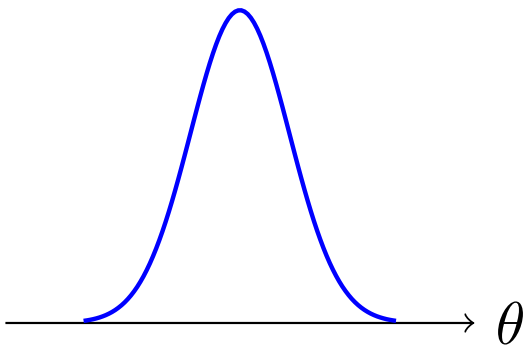
- Calculated using Bayes' theorem:

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})}$$

- Reflects updated beliefs after incorporating evidence.
- Dependent on likelihood $P(\mathcal{D} \mid \theta)$.

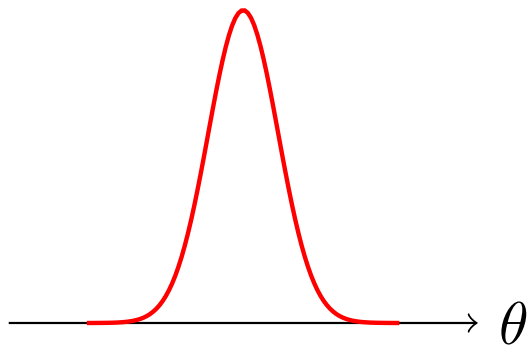
# Illustration

Prior $P(\theta)$



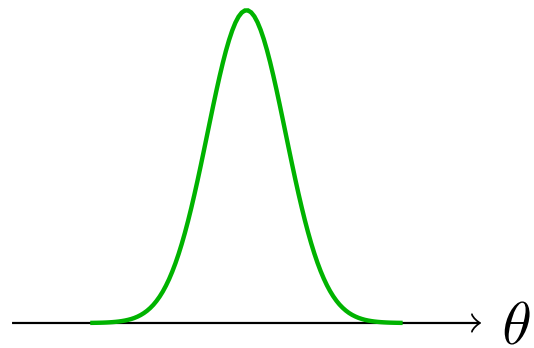

Prior knowledge

Likelihood $P(\mathcal{D}|\theta)$




Evidence

Posterior $P(\theta|\mathcal{D})$




Updated belief

# Prior and Posterior Distributions

We will define two distributions:

1. **Prior** $P$ on a hypothesis set $\mathcal{F}$ that **does not depend on the data** $S$. This prior is typically "wide", i.e. we won't exclude many functions unless we have a precise idea of what functions $f \in \mathcal{F}$ should be allowed.
2. **Posterior** $Q := Q(S)$ on the hypothesis set $\mathcal{F}$ that can depend on the data $S$. As we will see shortly, our model (e.g. a deep neural network) will be assumed to be drawn from the distribution $Q$.

Note that we will typically assume that $P \gg Q$, i.e. the support of $P$ dominates the support of $Q$.

# General idea PAC-Bayes bounds

- Construct a stochastic classifier via the distribution $Q$
- Bound the expected generalization gap $\mathbb{E}_{f\sim Q}[\delta_f^S]$

$\rightsquigarrow$ we do not choose a specific classifier $f \in \mathcal{F}$ anymore, but a distribution $Q$ over the space of functions $\mathcal{F}$.

# Recall KL divergence

The Kullback-Leibler (KL) divergence measures how one probability distribution diverges from a second probability distribution.

For discrete distributions:

$$\mathrm{KL}(P||Q) = \sum_i P(i) \cdot \ln\left(\frac{P(i)}{Q(i)}\right). \tag{1}$$

For continuous distributions:

$$\mathrm{KL}(P||Q) = \int P(x) \cdot \ln\left(\frac{P(x)}{Q(x)}\right) dx. \tag{2}$$

One can for instance check that if $P = Q$, then the KL divergence is zero. Note however that this metric is not symmetric in general, i.e. $\mathrm{KL}(Q||P) \neq \mathrm{KL}(P||Q)$.

# Recall KL divergence

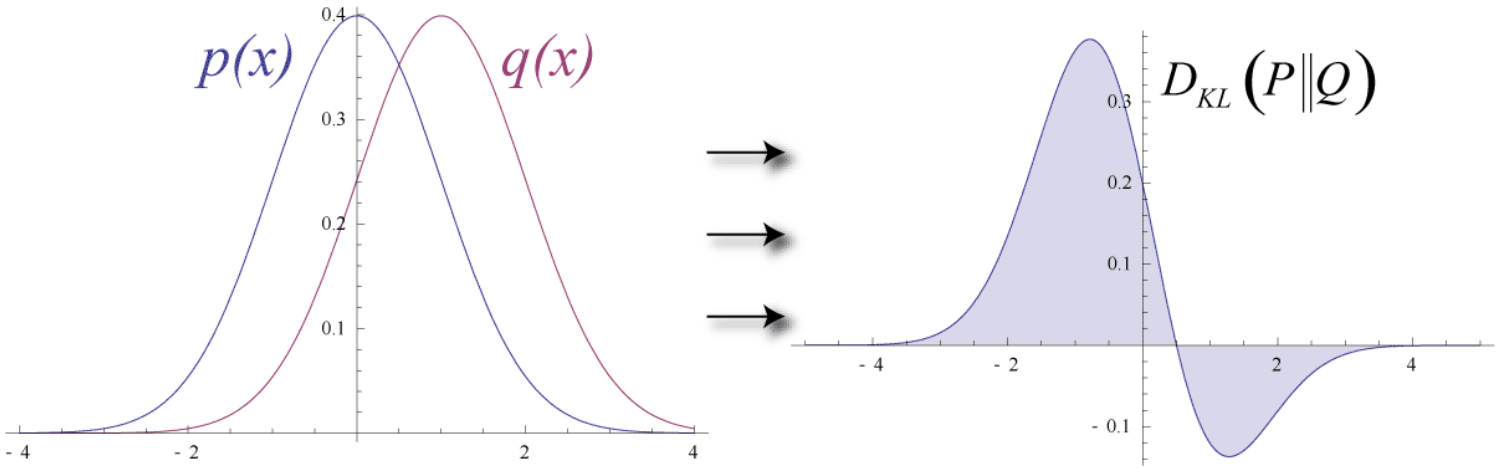



Figure: Illustration KL divergence for continuous distributions. Source: Wikipedia.

# Change of measure

**Problem:** Distribution $Q$ depends on $S$, potentially in a complicated way

$\rightsquigarrow$ A key step in the PAC-Bayes analysis is to convert the expectation under $Q$ to an expectation under $P$.

Lemma 1 (Change of measure inequality [DV75])

*For any $P \gg Q$, $P$-measurable function $\phi$ (random variable)*

$$\mathbb{E}_{f\sim Q}[\phi(f)] \leq KL(Q||P) + \ln \mathbb{E}_{f\sim P}\left[e^{\phi(f)}\right].$$

Note that for the bound to provide a tight estimate, we need $\mathrm{KL}(Q||P)$ to be small, i.e., $Q$ and $P$ should be close to each other.

# Proof

# PAC-Bayesian Theorem

We will first prove a general PAC bound that makes use of a **general function** $\Delta$ to measure the distance between $\mathbb{E}_Q[R(f)]$ and $\mathbb{E}_Q[R_S(f)]$
$\rightsquigarrow$ we will later use an **explicit** function $\Delta$.

Before we state the theorem, note that since we consider the 0-1 loss and the samples are taken to be i.i.d., then $NR_S(f)$ can only take values from the set $\{0, \dots N\}$ and it follows a binomial distribution:

$$\text{Bin}(k; N, R(f)) = \binom{N}{k} (R(f))^k (1 - R(f))^{N-k}.$$

*(this is the probability of getting exactly k successes in N independent Bernoulli trials (with the same rate $R(f)$)*

# PAC-Bayesian Theorem

Theorem 2 ( [BGLR14])

*For fixed $P$ and any $Q$, $\epsilon \in (0,1)$ and for any $\Delta : [0,1] \times [0,1] \to \mathbb{R}$ convex function, the following holds with probability greater than $1-\epsilon$ over sample sets $S$ of size $N$:*

$$\Delta(\mathbb{E}_{f\sim Q}[R(f)], \mathbb{E}_{f\sim Q}[R_S(f)]) \leq \frac{1}{N}\left(KL(Q||P) + \ln \frac{J_\Delta(N)}{\epsilon}\right),$$

*where*
$J_\Delta(N) = \left(\sup_{m\in[0,1]} \sum_{k=0}^{N} Bin(k;N,m) \exp\left(N\Delta\left(m, \frac{k}{N}\right)\right)\right).$

Recall: $\text{Bin}(k;N,R(f)) = \binom{N}{k}(R(f))^k(1-R(f))^{N-k}$

# Proof

# Theorem [McA03]

Variants of the theorem can be derived for specific choices of $\Delta$.
For instance, in the case of the square function...

Theorem 3 ( [McA03])

*For fixed $P$ and any $Q$, $\epsilon \in (0;1)$ with probability greater than $1-\epsilon$ over sample sets $S$ of size $N$:*

$$\mathbb{E}_{f\sim Q}[R(f)] - \mathbb{E}_{f\sim Q}[R_S(f)] \leq \sqrt{\frac{2}{N}\left[KL(Q||P) + \ln\left(\frac{2\sqrt{N}}{\epsilon}\right)\right]}$$

Proof idea.
The proof is a combination of Theorem 2 where $\Delta = (R(f) - R_S(f))^2$ is the square function. □

# Comparison to VC theory

## VC bounds

- Depend on the complexity of the model (through the number of parameters)
- Do not consider any property of the underlying data distribution we are trying to approximate
- Deep learning models have a large number of parameters $\Longrightarrow$ VC bounds yield so-called vacuous bounds (i.e. these bounds do not explain generalization)

## PAC-Bayes

- Do not depend at all on the model complexity of the underlying function class
- Instead, they depend on a prior $P$ and the output of the algorithm encoded in $Q$, which depends on the data distribution

Section 2

# PAC-Bayesian for DNNs

# General recipe [DR17]

1. Choose a Gaussian $P = \mathcal{N}(\theta_0, \lambda \mathbf{I})$, e.g. a simple way should be $\theta_0 = \mathbf{0}$, $\lambda = 1$ but one could also cross-validate
2. Choose a Gaussian $Q = \mathcal{N}(\theta, \text{diag}(\sigma))$, where $\theta$ are the parameters of our model (e.g. a fully-trained DNN) and where $\sigma_i$ is the variance in the $i$-th weight. Choosing $\sigma_i$ small means that we need more precision for this specific parameter
3. With our choice of $P$ and $Q$, we can derive a simple expression for $\text{KL}(Q||P)$
4. Minimize PAC-Bayes bound (typically using a surrogate loss)

$$\mathbb{E}_{f\sim Q}[R(f)] - \mathbb{E}_{f\sim Q}[R_S(f)] \leq \sqrt{\frac{2}{N}\left[\text{KL}(Q||P) + \ln\left(\frac{2\sqrt{N}}{\epsilon}\right)\right]}$$

- achieve small error on sample
- find wide minima: robust to large parameter perturbations

# Experimental evaluation [DR17]

## Binary version of MNIST

| Experiment | T-600 | T-1200 | T-$300^2$ | T-$600^2$ | T-$1200^2$ | T-$600^3$ | R-600 |
|---|---|---|---|---|---|---|---|
| Train error | 0.001 | 0.002 | 0.000 | 0.000 | 0.000 | 0.000 | 0.007 |
| Test error | 0.018 | 0.018 | 0.015 | 0.016 | 0.015 | 0.013 | 0.508 |
| SNN train error | 0.028 | 0.027 | 0.027 | 0.028 | 0.029 | 0.027 | 0.112 |
| SNN test error | 0.034 | 0.035 | 0.034 | 0.033 | 0.035 | 0.032 | 0.503 |
| PAC-Bayes bound | 0.161 | 0.179 | 0.170 | 0.186 | 0.223 | 0.201 | 1.352 |
| KL divergence | 5144 | 5977 | 5791 | 6534 | 8558 | 7861 | 201131 |
| # parameters | 471k | 943k | 326k | 832k | 2384k | 1193k | 472k |
| VC dimension | 26m | 56m | 26m | 66m | 187m | 121m | 26m |

Table 1: Results for experiments on binary class variant of MNIST. SGD is either trained on (T) true labels or (R) random labels. The network architecture is expressed as $N^L$, indicating $L$ hidden layers with $N$ nodes each. Errors are classification error. The reported VC dimension is the best known upper bound (in millions) for ReLU networks. The SNN error rates are tight upper bounds (see text for details). The PAC-Bayes bounds upper bound the test error with probability 0.965.

Section 3

# A short note on Double Descent

# Double Descent

**Traditonal view in machine learning:** Larger models tend to overfit to the training data, yielding to poor generalization to unseen data.
$\rightsquigarrow$ incomplete picture as shown by [BHMM19]

**Modern view:** As one increases the size of a model (entering the so called over-parametrized regime where the number of parameters exceeds the number of training datapoints), one observes that the test error starts decreasing, even though the training error goes to zero.

# Double Descent: illustration

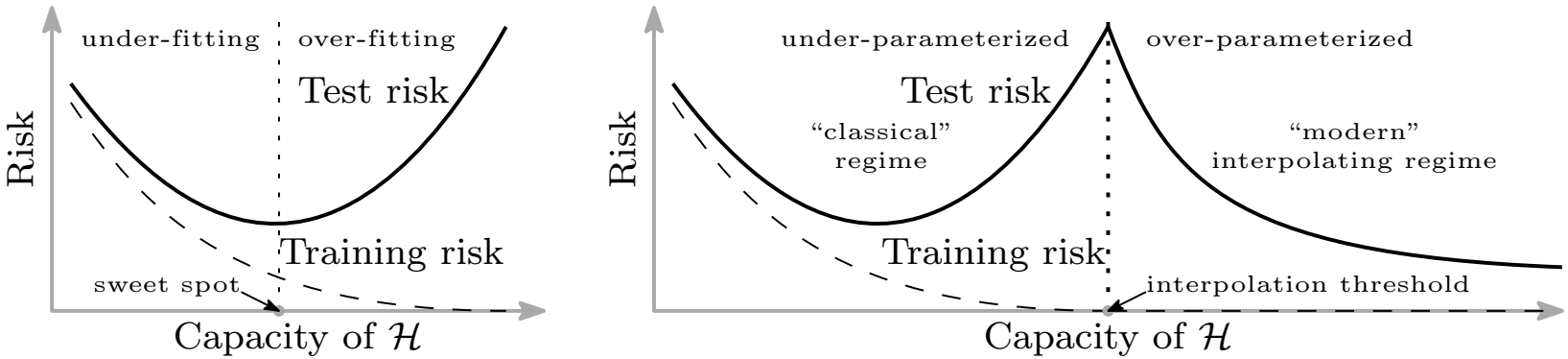



Figure: **Curves for training risk (dashed line) and test risk (solid line). Figure from [BHMM19]** (**a**) The classical **U-shaped risk curve** arising from the bias-variance trade-off. (**b**) The **double descent risk curve**, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behavior from using high capacity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.

Luc Bégin, Pascal Germain, Franccois Laviolette, and Jean-Francis Roy.

Pac-bayesian theory for transductive learning.

In *Artificial Intelligence and Statistics*, pages 105–113. PMLR, 2014.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal.

Reconciling modern machine-learning practice and the classical bias–variance trade-off.

*Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Gintare Karolina Dziugaite and Daniel M Roy.

Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data.

*arXiv preprint arXiv:1703.11008*, 2017.

Monroe D Donsker and SR Srinivasa Varadhan.

Asymptotics for the wiener sausage.

*Communications on Pure and Applied Mathematics*, 28(4):525–565, 1975.


David McAllester.

Simplified pac-bayesian margin bounds.

In *Learning theory and Kernel machines*, pages 203–215. Springer, 2003.