CS-E4710 Machine Learning: Supervised Methods

Lecture 2: Statistical learning theory

Juho Rousu

September 13, 2022

Department of Computer Science
Aalto University

Generalization

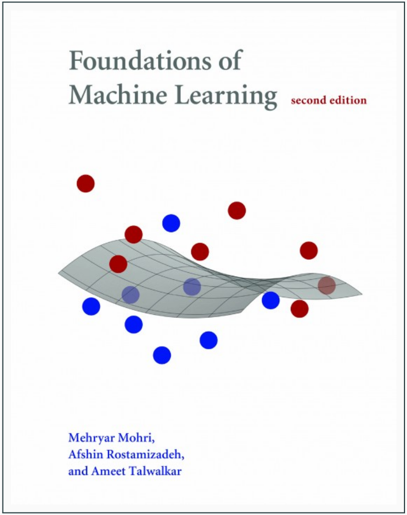
- Our aim is to predict as well as possible the outputs of future examples, not only for training sample
- We would like to minimize the **generalization error**, or the (true) **risk**

  $$R(h) = \mathbb{E}_{(\mathbf{x},y)\sim D}\left[\, L(h(\mathbf{x}), y) \,\right],$$

  where $L(y, y')$ is a suitable loss function (e.g. zero-one loss)
- Assuming future examples are independently drawn from the same distribution D that generated the training examples (i.i.d assumption)
- But we do not know D!
- What can we say about $R(h)$ based on training examples and the hypothesis class $\mathcal{H}$ alone? Two possibilities:
  - Empirical evaluation through testing
  - **Statistical learning theory (Lectures 2 and 3)**

Additional reading

- Lectures 2-4 are mostly based on Mohri et al book: chapters 2-4
- Available online in Aalto eBookAalto Central: `https://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?pq-origsite=primo&docID=6246520`
- The book goes much deeper in the theory (e.g. proofs of theorems) than what we do in the course



Probably approximately correct learning

Probably Approximate Correct Learning framework

- Probably Approximate Correct (PAC) Learning framework formalizes the notion of generalization in machine learning
- Ingredients:
    - input space X containing all possible inputs x
    - set of possible labels $\mathcal{Y}$ (in binary classification $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, +1\}$)
    - Concept class $\mathcal{C}$ containing concepts $C : X \mapsto \mathcal{Y}$ (to be learned), concept C gives a label $C(x)$ for each input x
    - unknown probability distribution D
    - training sample $S = (x_1, C(x_1)), \dots, (x_m, C(x_m))$ drawn independently from D
    - hypothesis class $\mathcal{H}$, in the basic case $\mathcal{H} = \mathcal{C}$ but this assumption can be relaxed
- The goal in PAC learning is to learn a hypothesis with a low generalization error

$$R(h) = \mathbb{E}_{x \sim D}\left[\, L_{0/1}(h(x), C(x)) \,\right] = \Pr_{x \sim D}(h(x) \neq C(x))$$

PAC learnability

- A class $\mathcal{C}$ is **PAC-learnable**, if there exist an algorithm $\mathcal{A}$ that given a training sample S outputs a hypothesis $h_S \in \mathcal{H}$ that has generalization error satisfying

$$Pr(R(h_S) \leq \epsilon) \geq 1 - \delta$$

  - for **any** distribution D, for arbitrary $\epsilon, \delta > 0$ and sample size $m = |S|$ that grows polynomially in $1/\epsilon$,$1/\delta$
  - for **any** concept $C \in \mathcal{C}$
- In addition, if $\mathcal{A}$ runs in time polynomial in m,$1/\epsilon$,and $1/\delta$ the class is called **efficiently PAC learnable**

Interpretation

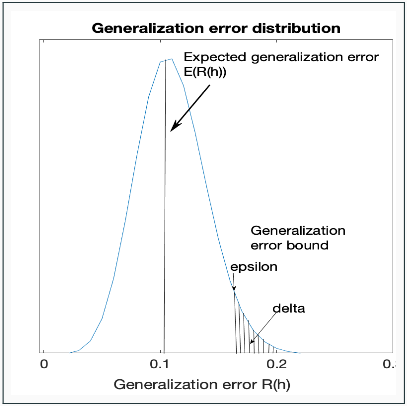
Let us interpret the bound

$$Pr(R(h_S) \leq \epsilon) \geq 1 - \delta$$

- ϵ sets the level of generalization error that is of interest to us, say we are content with predicting incorrectly 10% of the new data points: $\epsilon = 0.1$
- $1 - \delta$ sets a level of confidence, if we are content of the training algorithm to fail 5% of the time to provide a good hypothesis: $\delta = 0.05$
- Samples size and running time should **not explode** when we make ϵ and δ stricter: requirement of polynomial growth
- The event "low generalization error", $\{R(h_S) \leq \epsilon\}$ is considered as a random variable because we cannot know beforehand which hypothesis $h_S \in \mathcal{H}$ will be selected by the algorithm

Generalization error bound vs. test error

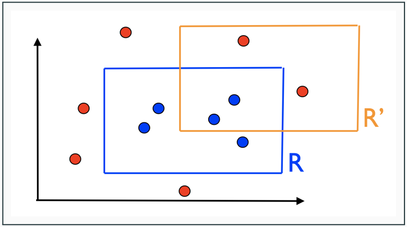
- Generalization error bounds concern the tail of the error distribution
    - We wish a high generalization error to be a **rare event**
- Expected generalization error which might be considerably lower
    - Analyzing average behaviour where most data distributions and concepts are "not bad"
- We expect there be a gap between the expected error and the tail
    - The smaller the failure probability δ, the larger the gap



Example: learning axis-aligned rectangles

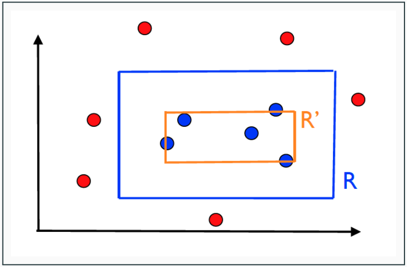
Learning setup

- The goal is to learn a rectangle R (representing the true concept) that includes all blue points and excludes all red points
- The hypotheses also will be rectangles (here R'), which will in general have both false positive predictions (predicting blue when true label is red) and false negative predictions (predicting red when true label is blue).



Example: learning axis-aligned rectangles

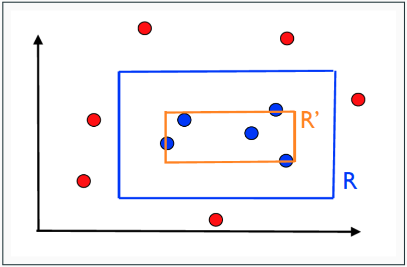
- We will use a simple algorithm: choose the tightest rectangle that contains all blue points
- Note that this will be a consistent hypothesis: no errors on the training data
- The hypothesis R' will only make false negative errors compared to the true concept R, no false positive errors (Q: Why is that?)



Example: learning axis-aligned rectangles

Questions:

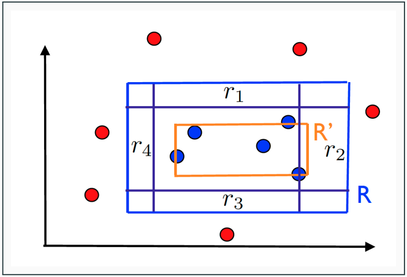
- Is the class of axis-aligned rectangles PAC-learnable?
- How much training data will be needed to learn?
- Need to bound the risk of outputting a bad hypothesis $R(\mathrm{R}') > \epsilon$ with high probability $1 - \delta$
- We can assume $Pr_D(R) > \epsilon$ (otherwise $R(\mathrm{R}') \leq \epsilon$ trivially)



Example: learning axis-aligned rectangles

Let r_1, r_2, r_3, r_4 be rectangles along the sides of R such that $Pr_D(r_i) = \epsilon/4$

- Their union satisfies $Pr_D(r_1 \cup r_2 \cup r_3 \cup r_4) \leq \epsilon$
- Errors can only occur within $R - R'$ (false negative predictions)
- If R' intersects all of the four regions $r_1, \ldots, r_4$ then we know that

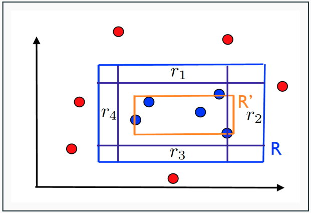


$$R(\mathrm{R}') \leq \epsilon$$

Thus, if $R(\mathrm{R}') > \epsilon$ then R' must miss **at least one** of the four regions

Example: learning axis-aligned rectangles

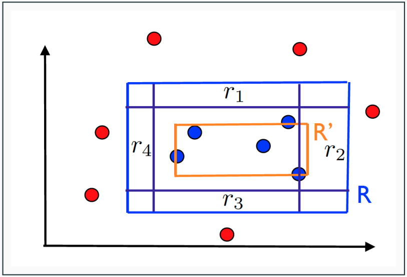
- Events $A = \{\mathrm{R}' \text{ intersects all four rectangles } r_1, \ldots, r_4\}$, $B = \{R(\mathrm{R}') < \epsilon\}$, satisfy $A \subseteq B$
- Complement events $A_C = \{\mathrm{R}' \text{ misses at least one rectangle }\}$, $B_C = \{R(\mathrm{R}') \geq \epsilon\}$ satisfy $B_C \subseteq A_C$
- B_C is the bad event (high generalization error), we want it to have low probability
- In probability space, we have $Pr(B_C) \leq Pr(A_C)$
- Our task is to upper bound $Pr(A_C)$



Example: learning axis-aligned rectangles

- Each r_i has probability mass $\epsilon/4$ by our design
- Probability of one example missing one rectangle: $1 - \epsilon/4$
- Probability of m examples missing one rectangle: $(1 - \epsilon/4)^m$ (m times repeated trial with replacement)
- Probability of all examples missing at least one of the rectangles:

$$Pr(A_C) \leq 4(1 - \epsilon/4)^m$$



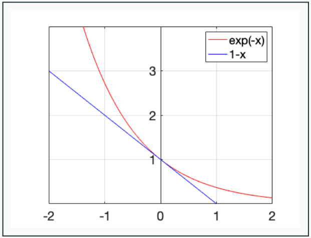
Example: learning axis-aligned rectangles

- We can use a general inequality $\forall x : (1 - x) < \exp(-x)$ to obtain:

$$Pr(R(h) \geq \epsilon) \leq 4(1-\epsilon/4)^m \leq 4\exp(-m\epsilon/4)$$

- We want this probability to be small $(< \delta)$:

$$4\exp(-m\epsilon/4) < \delta$$
$$\Leftrightarrow m \geq 4/\epsilon \log 4/\delta$$

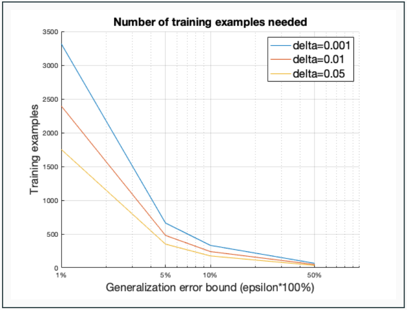


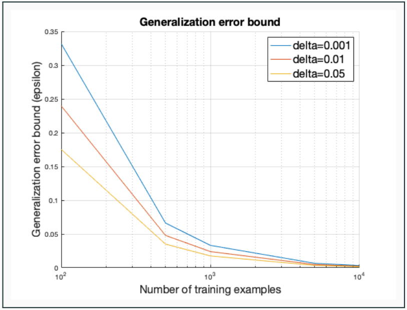
- The last inequality is our first generalization error bound, a **sample complexity** bound to be exact

Note: corresponding to Mohri et al (2018) log denotes the natural logarithm: $\log(\exp(x)) = x$

Plotting the behaviour of bound

- Left, the sample complexity, the number of examples needed to reach a given generalization error level is shown $m(\epsilon, \delta) = 4/\epsilon \log 4/\delta$
- Right, the generalization bound is plotted as a function of training sample size $\epsilon(m, \delta) = 4/m \log 4/\delta$
- Three different confidence levels (δ) are plotted

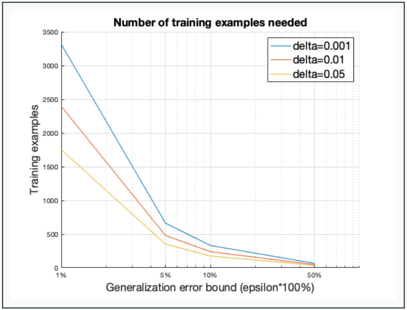


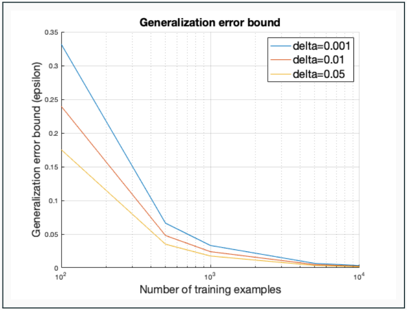


Plotting the behaviour of the bound

Typical behaviour of ML learning algorithms is revealed:

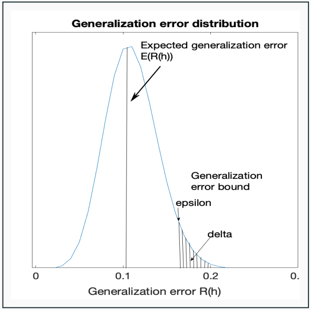
- increase of sample size decreases generalization error
- extra data gives less and less additional benefit as the sample size grows (law of diminishing returns)
- requiring high level of confidence (small δ) for obtaining low error requires more data for the same level of error





Generalization error bound vs. expected test error

- The error bounds hold for any concepts from the class
    - including concepts that are harder to learn than "average concept"
- They hold for **any** distribution *D* generating the data
    - Including adversially generated distributions (aiming to make learning harder)
- We bound the probability of being in the high error tail of the distribution (not the convergence to the mean or median generalization error)



For these reasons empirically estimated test errors might be considerably lower than the bounds suggest

Half-time poll: Personalized email spam filtering system

Company is developing a personalized email spam filtering system. The system is tuned personally for each customer using the customers data on top of the existing training data. The company has a choice of three machine learning algorithms, with different performance characteristics. So far, the company has tested three different algorithms on a small set of test users.

Which algorithm should the company choose?

1. Algorithm 1, which guarantees error rate of less than 10% for 99% of the future customer base
2. Algorithm 2, which guarantees error rate of less than 5% for 90% of the future customer base
3. Algorithm 3, which empirically has error rate of 1% on the current user base

Answer to the poll in Mycourses by 11:15: Go to Lectures page and scroll down to "Lecture 2 poll": `https://mycourses.aalto.fi/course/view.php?id=37029&section=2`

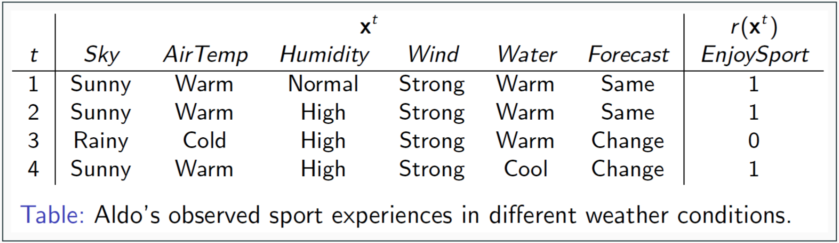
Guarantees for finite hypothesis sets

Finite hypothesis classes

- Finite concept classes arise when:
  - Input variables have finite domains or they are converted to such in preprocessing (e.g. discretizing real values), and
  - The representations of the hypotheses have finite size (e.g. the number of times a single variable can appear)
  - Dealing with subclasses of Boolean formulae, expressions binary input variables (literals) combined with logical operators (AND, OR, NOT,...)
- Finite concept classes have been thoroughly analyzed hypothesis classes in statistical learning theory

Example: Boolean conjunctions

- Aldo likes to do sport only when the weather is suitable
- Also has given examples of suitable and not suitable weather
- Let us build a classifier for Aldo to decide whether to do sports today
- As the classifier we use rules in the form of boolean conjunctions (boolean formulae containing AND, and NOT, but not OR operators): e.g. if (Sky=Sunny) AND NOT(Wind=Strong) then (EnjoySport=1)



			$\mathbf{x}^t$				$r(\mathbf{x}^t)$
t	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	1
2	Sunny	Warm	High	Strong	Warm	Same	1
3	Rainy	Cold	High	Strong	Warm	Change	0
4	Sunny	Warm	High	Strong	Cool	Change	1

Table: Aldo's observed sport experiences in different weather conditions.

Finite hypothesis class - consistent case

- Sample complexity bound relying on the size of the hypothesis class (Mohri et al, 2018): $Pr(R(h_s) \leq \epsilon) \geq 1 - \delta$ if

$$m \geq \frac{1}{\epsilon}(\log(|\mathcal{H}|) + \log(\frac{1}{\delta}))$$

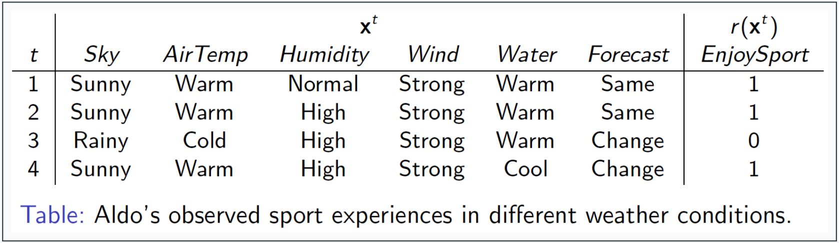
- An equivalent generalization error bound:

$$R(h) \leq \frac{1}{m}(\log(|\mathcal{H}|) + \log(\frac{1}{\delta}))$$

- Holds for any finite hypothesis class assuming there is a consistent hypothesis, one with zero empirical risk
- Extra term compared to the rectangle learning example is the term $\frac{1}{\epsilon}(\log(|\mathcal{H}|))$
- The more hypotheses there are in $\mathcal{H}$, the more training examples are needed

Example: Boolean conjunctions

- How many different conjunctions can be built ($=|\mathcal{H}|$)
- Each variable can appear with or without "NOT" or can be excluded from the rule = 3 possibilities
- The total number of hypotheses is thus 3^d, where d is the number of variables
- We have six variables in total, giving us $|\mathcal{H}| = 3^6 = 729$ different hypotheses

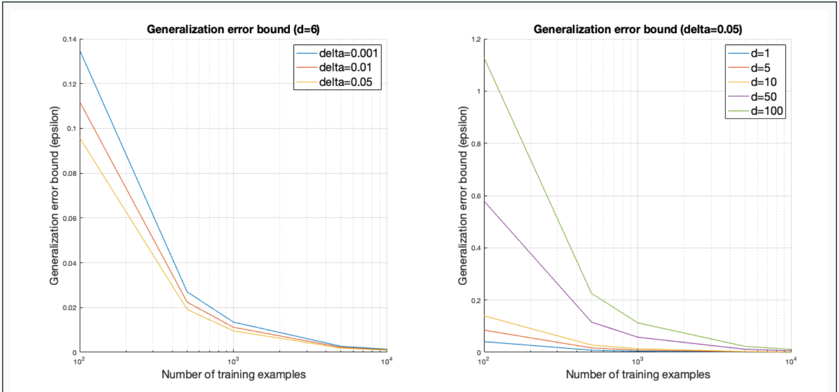


			$\mathbf{x}^t$				$r(\mathbf{x}^t)$
t	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	1
2	Sunny	Warm	High	Strong	Warm	Same	1
3	Rainy	Cold	High	Strong	Warm	Change	0
4	Sunny	Warm	High	Strong	Cool	Change	1

Table: Aldo's observed sport experiences in different weather conditions.

Plotting the bound for Aldo's problem using boolean conjunctions

- On the left, the generalization bound is shown for different values of δ, using $d = 6$ variables
- On the right, the bound is shown for increasing number of input variables d, using $\delta = 0.05$



Arbitrary boolean formulae

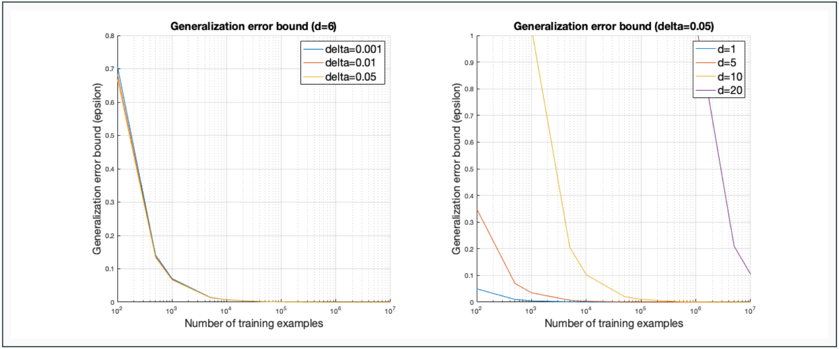
- What about using **arbitrary** boolean formulae?
- How many boolean formulae of d variables there are?
- There are 2^d possible input vectors, size of the input space is $|X| = 2^d$
- We can define a boolean formula that outputs 1 for an arbitrary subset of $S \subset X$ and zero outside that subset: $f_S(\mathbf{x}) = (\mathbf{x} = \mathbf{x}_1) OR (\mathbf{x} = \mathbf{x}_2) OR \cdots OR (\mathbf{x} = \mathbf{x}_{|S|})$
- We can pick the subset in $2^{|X|}$ ways (Why?)
- Thus we have $|\mathcal{H}| = 2^{2^d}$ different boolean formula
- Our generalization bound gives

$$m \geq \frac{1}{\epsilon}(2^d \log 2 + \log(\frac{1}{\delta}))$$

- Thus we need exponential number of examples with respect to the number of variables; the hypothesis class is considered not PAC-learnable!

Plotting the bound for Arbitrary boolean formulae

- With $d = 6$ variables we need ca. 500 examples to get bound below 0.07 (left picture)
- Increase of number of variables quickly raises the sample complexity to 10^6 and beyond (right picture)



Proof outline of the PAC bound for finite hypothesis classes

Proof outline (Mohri et al., 2018)

- Consider any hypothesis $h \in \mathcal{H}$ with $R(h) > \epsilon$
- For h to be consistent $\hat{R}(h) = 0$, all training examples need to miss the region where h is making an error.
- The probability of this event is

$$Pr(\hat{R}(h) = 0 | R(h) > \epsilon) \leq (1 - \epsilon)^m$$

- m times repeated trial with success probability ϵ
- This is the probability that one consistent hypothesis has high error

Proof outline

- But we do not need which consistent hypothesis h is selected by our learning algorithm
- Hence our result will need to hold for all consistent hypotheses
  - This is an example of **uniform convergence** bound
- We wish to upper bound the probability that some $h \in \mathcal{H}$ is consistent $\hat{R}(h) = 0$ and has a high generalization error $R(h) > \epsilon$ for a fixed $\epsilon > 0$:

$$Pr(\exists h \in \mathcal{H} | \hat{R}(h) = 0 \wedge R(h) > \epsilon)$$

- Above $\wedge$ is the logical "and"

Proof outline

- We can replace $\exists$ by enumerating all hypotheses in $\mathcal{H}$ using logical-or ($\vee$)

$$Pr(\exists h \in \mathcal{H} | \hat{R}(h) = 0 \wedge R(h) > \epsilon) =$$
$$Pr\big(\{\hat{R}(h_1) = 0 \wedge R(h_1) > \epsilon\} \vee \{\hat{R}(h_2) = 0 \wedge R(h_2) > \epsilon\} \vee \cdots\big)$$

- Using the the fact that $Pr(A \cup B) \leq Pr(A) + Pr(B)$ and $Pr(A \cap C) \leq Pr(A|C)$ for any events A,B and C the above is upper bounded by

$$\leq \sum_{h \in \mathcal{H}} Pr(\hat{R}(h) = 0 \wedge R(h) > \epsilon) \leq \sum_{h \in \mathcal{H}} Pr(\hat{R}(h) = 0 | R(h) > \epsilon)$$
$$\leq |\mathcal{H}|(1 - \epsilon)^m$$

- Last inequality follows from using the $Pr(\hat{R}(h) = 0 | R(h_1) > \epsilon) \leq (1 - \epsilon)^m$ for the $|\mathcal{H}|$ summands

Proof outline

- We have established

$$Pr(\exists h \in \mathcal{H} | \hat{R}(h) = 0 \wedge R(h) > \epsilon) \leq |\mathcal{H}|(1-\epsilon)^m \leq |\mathcal{H}| \exp(-m\epsilon)$$

- Set the right-hand side equal to δ and solve for m to obtain the bound:

$$\begin{aligned} \delta &= |\mathcal{H}| \exp\left(-m\epsilon\right) \\ \log \delta &= \log |\mathcal{H}| - m\epsilon \\ m &= \frac{1}{\epsilon}(\log(|\mathcal{H}|) + \log(1/\delta)) \end{aligned}$$

Finite hypothesis class - inconsistent case

- So far we have assumed that there is a consistent hypothesis $h \in \mathcal{H}$, one that achieves zero empirical risk on training sample
- In practise this is often not the case
- However as long as the empirical risk $\hat{R}(h)$ is small, a low generalization error can still be achieved
- Generalization error bound (Mohri, et al. 2018): Let $\mathcal{H}$ be a finite hypothesis set. Then for any $\delta > 0$ with probability at least $1 - \delta$ we have for all $h \in \mathcal{H}$:

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{\log(|\mathcal{H}|) + \log(2/\delta)}{2m}}$$

- We see the dependency from $\log |\mathcal{H}|/m$ as in the consistent case but now under square root
    - Slower convergence w.r.t number of examples

Summary

- Probably approximately correct learning is a theoretical framework for analysing the generalization performance of machine learning algorithms
- PAC theory is concerned about upper bounding the probability δ of "bad" events, those of high generalization error (ϵ)
- In finite hypothesis classes, (the logarithm of) number of hypothesis $\log |\mathcal{H}|$ in the hypothesis class affects the number of examples needed to obtain a given level of risk with high probability