Generalization Bounds

Overview

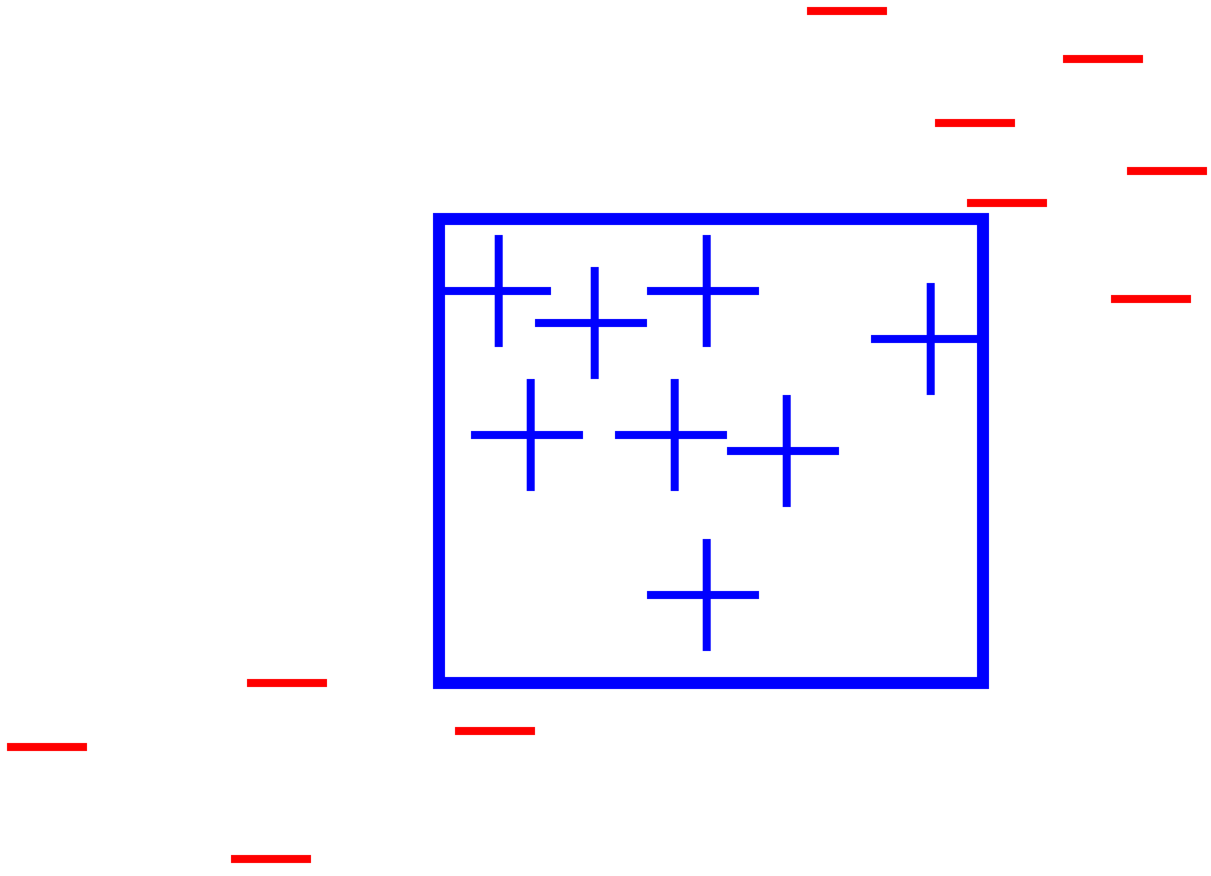
- Probably Approximately Correct (PAC) model
- Basic generalization bounds
  - finite hypothesis class
  - infinite hypothesis class
- Model Selection

Good-Turing problem

- Assume you have access to a large data set of words
- Words are drawn i.i.d from some distribution
- You observe a sample S of size m
- QUESTION: what is the probability of the words you did not observe?

Motivating Example (PAC)

- Concept: Average body-size person
- Inputs: for each person:
  - height
  - weight
- Sample: labeled examples of persons
  - label + : average body-size
  - label - : not average body-size
- Two dimensional inputs



Motivating Example (PAC)

- Assumption: target concept is a rectangle.
- Goal:
  - Find a rectangle that “approximate” the target.
- Formally:
  - With high probability
  - output a rectangle such that
  - its error is low.

Example (Modeling)

- Assume:
  - Fixed distribution over persons.
- Goal:
  - Low error with respect to THIS distribution!!!
- How does the distribution look like?
  - Highly complex.
  - Each parameter is not uniform.
  - Highly correlated.

Model Based approach

- First try to model the distribution.
- Given a model of the distribution:
  - find an optimal decision rule.

- Bayesian Learning

PAC approach

- Assume that the distribution is fixed.
- Samples are drawn are i.i.d.
  - independent
  - identical
- Concentrate on the decision rule rather than distribution.

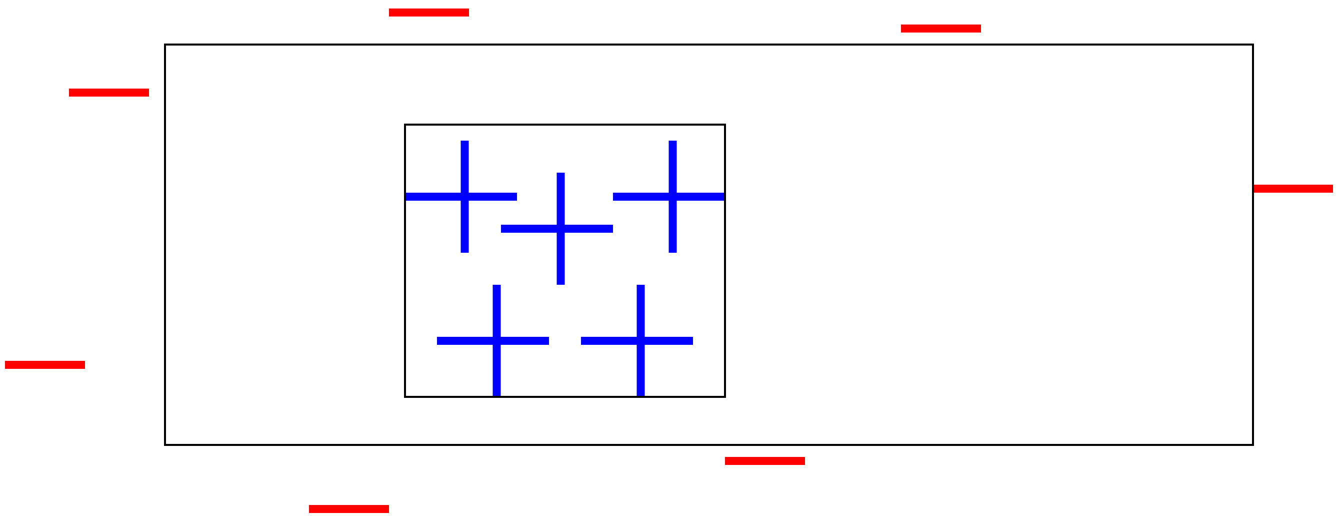
PAC Learning

- Task: learn a rectangle from examples.
- Input: point (x,y) and classification + or -
  - classifies by a rectangle R
- Goal:
  - in the fewest examples
  - compute R' efficiently
  - R' is a good approximation for R

PAC Learning: Accuracy

- Testing the accuracy of a hypothesis:
  - using the distribution D of examples.
- Error = R Δ R'
- Pr[Error] = D(Error) = D(R Δ R')
- We would like Pr[Error] to be controllable.
- Given a parameter ε:
  - Find R' such that Pr[Error] < ε.

PAC Learning: Hypothesis



- Which Rectangle should we choose?
- Latter we show it is not that important.

PAC model: Setting

- A distribution: D (unknown)
- Target function: c_t from C
  - $c_t : X \rightarrow \{0,1\}$
- Hypothesis: h from H
  - $h: X \rightarrow \{0,1\}$
- Error probability:
  - $error(h) = Prob_D[h(x) \neq c_t(x)]$
- Oracle: $EX(c_t,D)$

PAC Learning: Definition

- C and H are concept classes over X.
- C is PAC learnable by H if
- There Exist an Algorithm A such that:
  - For any distribution D over X and c_t in C
  - for every input ε and δ:
  - outputs a hypothesis h in H,
  - while having access to $EX(c_t,D)$
  - with probability $1-\delta$ we have $error(h) < \varepsilon$
- Complexities.

PAC: comments

- We only assumed that examples are i.i.d.
- We have two independent parameters:
  - Accuracy ε
  - Confidence δ
- No assumption about the likelihood of concepts.
  - no prior
- Hypothesis is tested on the same distribution as the sample.

Finite Concept class

- Assume $C=H$ and finite.
  - realizable case
- h is ε-bad if error(h)> ε.
- Algorithm:
  - Sample a set S of $m(\varepsilon,\delta)$ examples.
  - Find h in H which is consistent.
- Algorithm fails if h is ε-bad.

Analysis

- Assume hypothesis g is ε-bad.
- The probability that g is consistent:
  - $\Pr[g \text{ consistent}] \leq (1-\varepsilon)^m < e^{-\varepsilon m}$
- The probability that there exists:
  - g is ε-bad and consistent:
  - $|H| \Pr[g \text{ consistent and } \varepsilon\text{-bad}] \leq |H|\, e^{-\varepsilon m}$
- Sample size:
  - $m > (1/\varepsilon) \ln (|H|/\delta)$

PAC: non-realizable case

- What happens if c_t not in H
- Needs to redefine the goal.
- Let h^* in H minimize the error $\beta = error(h^*)$
- Goal: find h in H such that
  - $error(h) \leq error(h^*) + \varepsilon = \beta + \varepsilon$
- Algorithm ERM
  - Empirical Risk Minimization

Concentration Bounds

- Markov inequality

  $\Pr[X > a] < E[X]/a, \quad X>0$

- Chebyshev:

  $\Pr[X > a] < E[X^2]/a^2$

- Chernoff: (X_i are Bernoulli r.v.)

  $\Pr[\Sigma_{i=1,n} X_i > \mu + \lambda] < \exp(-\lambda^2/n)$

Analysis

- For each h in H:
  - let obs-error(h) be the error on the sample S.
- Compute the probability that:
  - |obs-error(h) - error(h) | < $\varepsilon/2$
  - Chernoff bound: $\exp(-(\varepsilon/2)^2 m)$
- Consider entire H : $|H| \exp(-(\varepsilon/2)^2 m)$
- Sample size
  - ***$m > (4/\varepsilon^2) \ln (|H|/\delta)$***

Correctness

- Assume that for all h in H:
  - $|\text{obs-error}(h) - \text{error}(h)| < \varepsilon/2$
- In particular:
  - $\text{obs-error}(h^*) < \text{error}(h^*) + \varepsilon/2$
  - $\text{error}(h) - \varepsilon/2 < \text{obs-error}(h)$
- For the output h:
  - $\text{obs-error}(h) < \text{obs-error}(h^*)$
- Conclusion: $\text{error}(h) < \text{error}(h^*) + \varepsilon$

Example: Learning OR of literals

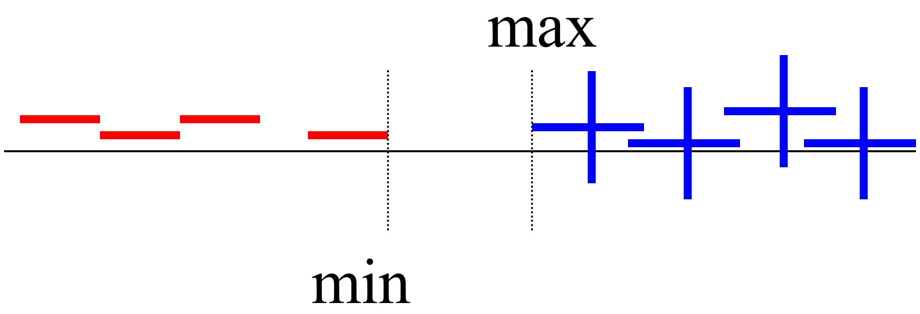
- Inputs: $x_1, \ldots, x_n$
- Literals : $x_1, \bar{x}_1$
- OR functions: $x_1 \vee \bar{x}_4 \vee x_7$
- Number of functions? $\mathbf{3^n}$

ELIM: Algorithm for learning OR

- Keep a list of all literals
- For every example whose classification is 0:
  - Erase all the literals that are 1.
- Example
- Correctness:
  - Our hypothesis h: An OR of our set of literals.
  - Our set of literals includes the target OR literals.
  - Every time h predicts zero: we are correct.
- Sample size: $m > (1/\varepsilon)\ \ln (3^n/\delta)$

Infinite Concept class

- $X=[0,1]$ and $H=\{c_\theta \mid \theta \text{ in } [0,1]\}$
- $c_\theta(x) = 0$ iff $x < \theta$
- Assume C=H:



- Which c_θ should we choose in [min,max]?

Proof I

- Show that the probability that
  - $\Pr[\, D([min,max]) > \varepsilon \,] < \delta$
- Proof: By Contradiction.
  - The probability that x in [min,max] at least ε
  - The probability we do not sample from [min,max]

  Is $(1-\varepsilon)^m$
  - Needs $m > (1/\varepsilon) \ln (1/\delta)$

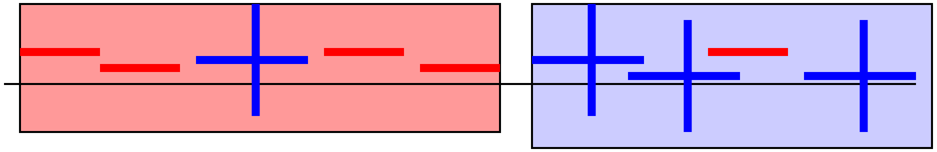
What's WRONG ?!

Proof II (correct):

- Let min' be : $D([min',\theta])=\varepsilon/2$
- Let max' be : $D([\theta,max'])=\varepsilon/2$
- Goal: Show that with high probability
  - X_- in $[min',\theta]$ and
  - X_+ in $[\theta,max']$
- In such a case any value in $[x_-,x_+]$ is good.
- Compute sample size!

Non-Feasible case

- Suppose we sample:



- Algorithm:
  - Find the function h with lowest error!

Analysis

- Define: z_i as a $\varepsilon/4$ - net (w.r.t. D)
- For the optimal h* and our h there are
  - z_j : $|error(h[z_j]) - error(h^*)| < \varepsilon/4$
  - z_k : $|error(h[z_k]) - error(h)| < \varepsilon/4$
- Show that with high probability:
  - $|obs\text{-}error(h[z_i]) - error(h[z_i])| < \varepsilon/4$
- Completing the proof.
- Computing the sample size.

General ε-net approach

- Given a class H define a class G
  - For every h in H
  - There exist a g in G such that
  - $D(g \Delta h) < \varepsilon/4$
- Algorithm: Find the best g in G.
- Computing the confidence and sample size.

Polynomials

- Polynomials of degree d:
  - parameters $a_{0,} \ldots, a_d, \theta$
  - computation: $\Sigma\, a_i x^i \geq \theta$
- Effective log class size:
  - $(d+1)\log(1/\varepsilon)$

Hyperplanes

- Domain $[0,1]^d$
- Concept class:
  - parameters $w \varepsilon [0,1]^d$ and θ
  - computation $\langle w,x \rangle \geq \theta$
- Effective log-class size:
  - $d \log (1/\varepsilon)$

VC dimension

- Overcoming the discritization
- Intuitively, the number of parameters.
  - VC-dim(hyperplans)=d+1
- Avoids the need of discritization
- A necessary and sufficient condition.

Model selection - Outline

- Motivation
- Overfitting
- Structural Risk Minimization
- Hypothesis Validation
- Minimum Description Length

Motivation:

- We have too few examples
- We have a very rich hypothesis class
- How can we find the best hypothesis
- Alternatively,
- Usually we choose the hypothesis class
- How should we go about doing it?

Overfitting

- Concept class: Intervals on a line
- Can classify any training set
- Zero training error: The only goal?!

Overfitting: Intervals

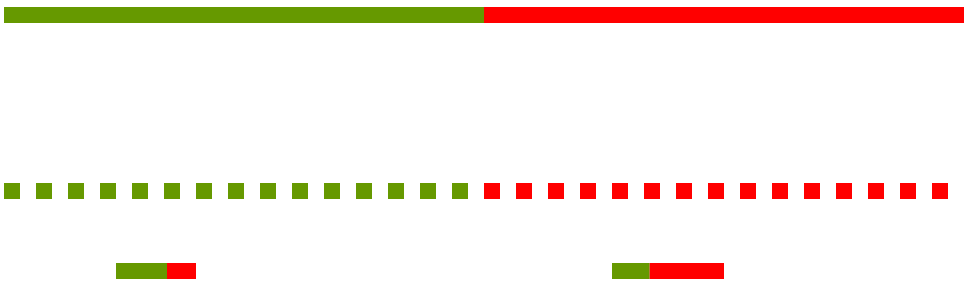
- Can always get zero error
- Are we interested?!

Overfitting: Intervals

intervals	0	1	2	3	4
errors	7	3	2	1	0

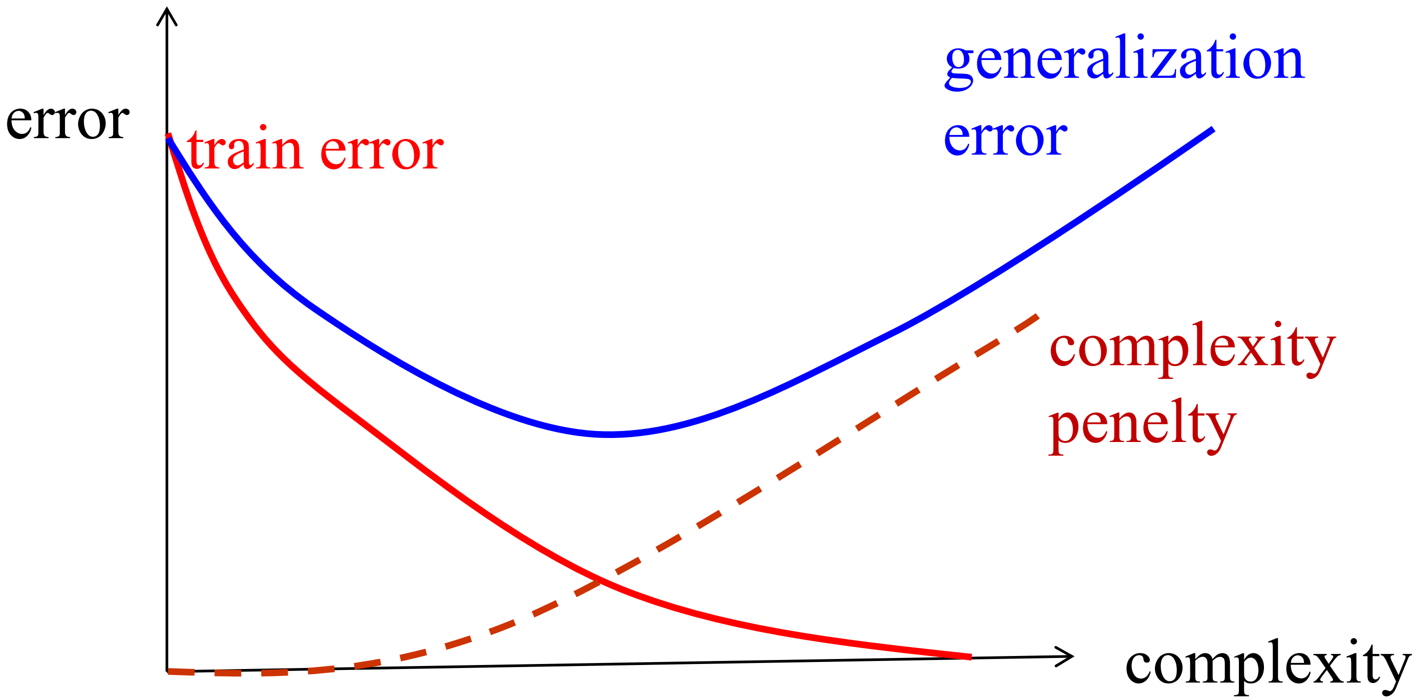
Overfitting

- Simple concept plus noise
- A very complex concept
  - insufficient number of examples



+ noise 1/3

Model Selection



Theoretical Model

- Nested Hypothesis classes
  - $H_1 \subseteq H_2 \subseteq H_3 \subseteq \ldots \subseteq H_i \subseteq$
  - For simplicity $|H_i| = 2^i$
- There is a target function $c(x)$,
  - For some i, $c \in H_i$
  - $\varepsilon(h) = Pr[\, h \neq c]$
  - $\varepsilon_i = \mathbf{min}_{h \in Hi}\, e(h)$
  - $\varepsilon^* = \mathbf{min}_i\, e_i$

Theoretical Model

- Training error
  - $obs(h) = Pr\,[\, h \neq c]$
  - $obs_i = \mathbf{min}_{h \in H_i}\, obs(h)$
- Complexity of h
  - $d(h) = \mathbf{min}_i \{h \in H_i\}$
- Add a penalty for *d(h)*
- minimize: *obs(h)+penalty(h)*

Structural Risk Minimization

- Penalty based.
- Chose the hypothesis which minimizes:
  - *obs(h)+penalty(h)*
- SRM penalty:

$$obs(h)+\sqrt{\frac{[d(h)+1]\ln 2/\delta}{m}} \approx \sqrt{\frac{d(h)}{m}\ln \frac{1}{\delta}}$$

SRM: Performance

- THEOROM
  - With probability $1\text{-}\delta$
  - h^* : best hypothesis
  - g^* : SRM choice
  - $\varepsilon(h^*) \leq \varepsilon(g^*) \leq \varepsilon(h^*) + 2\ penalty(h^*)$
- Claim: The theorem is "tight"
  - H_i includes 2^i coins

Proof

- Bounding the error in H_i
- Bounding the error across H_i

Hypothesis Validation

- Separate sample to training and selection.
- Using the training
  - Select from each H_i a candidate g_i
- Using the selection sample
  - select between $g_1, \dots, g_m$
- The split size
  - $(1-\gamma)m$ training set
  - γm selection set

Hypo.Validation: Performance

- Errors
  - $\varepsilon_{hv}(m)$, $\varepsilon_A(m)$
- Theorem: with probability 1-δ

$$\varepsilon_{hv}(m) \leq \varepsilon_A((1-\gamma)m) + \sqrt{\frac{\ln(m/\delta)}{\gamma m}}$$

- Is HV always near-optimal ?!

Minimum Description length

- Penalty: size of *h*
- Related to MAP
  - size of h: *log(Pr[h])*
  - errors: *log(Pr[D|h])*
- *Selection rule*
  - minimize errors + size(h)

Summary

- PAC model
- Generalization bounds
  - Empirical Risk Minimization
- Model Selection