Neuronale Netze (SS 2002), 12.6.

COLT theory

- In principle learning scenario:
  - unknown f is to be learned from examples
  - available: parameterized models with increasing complexity
  - choose a model $\rightarrow$ bound the structural risk, i.e. error due to too many free parameters which are not determined by the data
  - fit the parameters $\rightarrow$ bound the empirical risk, i.e. the training error
  - risk $=$ empirical risk $+$ structural risk

    So far, the structural risk has not been limited!
- COLT theory offers precise notations of generalization ability/learnability, guarantees, and bounds
- We restrict to binary function classes $\mathcal{F}$, inputs from X, probability P on X, functions to learn in $\mathcal{F}$, no noise.
- **Learning algorithm**

$$h : \bigcup_{m=1}^{\infty} (X \times \{0,1\})^m \rightarrow \mathcal{F}, \quad \vec{x} \mapsto h_m(\vec{x}, f)$$

- $d_P(f,g) := E_P(|f-g|)$, $\hat{d}_m(f,g,\vec{x}) := \sum |f(x_i) - g(x_i)|/m$
- A learning algorithm is PAC iff for all $\epsilon > 0$ the following holds

$$\sup_f P^m(\vec{x} \mid d_P(f, h_m(\vec{x}, f)) > \epsilon) \rightarrow 0 (m \rightarrow \infty)$$

- Finite function classes are PAC learnable, every algorithm without errors will do, and $|\mathcal{F}|^2(1-\epsilon)^m \leq \delta$ gives the number of training examples.

- example: $\{f : [0,1] \to \{0,1\} \mid f$ is almost surely constant$\}$ with the uniform distribution is PAC learnable.

  PAC algorithm: output the constant function which corresponds to the majority of the training examples

  Attention – the following algorithm does not make errors, but it is not PAC: output the function $f(x) = f(x_i)$ for $x = x_i$ for some training example, and $f(x) = 0$, otherwise. This cannot learn the constant function 1.