Foundations of Artificial Intelligence

Statistical Learning Theory

CS472 – Fall 2007
Thorsten Joachims

Outline

Questions in Statistical Learning Theory:
- How good is the learned rule after n examples?
- How many examples do I need before the learned rule is accurate?
- What can be learned and what cannot?
- Is there a universally best learning algorithm?

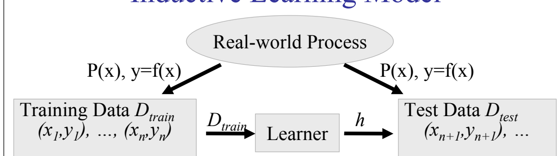
In particular, we will address:
What is the true error of h if we only know the training error of h?
- Finite hypothesis spaces and zero training error
- (Finite hypothesis spaces and non-zero training error)

Game: Randomized 20-Questions

Game: 20-Questions
- I think of object *f*
- For *i* = 1 to 20
  - You get to ask 20 yes/no questions about *f* and I have to answer truthfully
- You make a guess *h*
- You win, if *f*=*h*

Game: Randomized 20-Questions
- I pick function $f \in H$, where f: $X \rightarrow \{-1,+1\}$
- For *i* = 1 to 20
  - World delivers instances $x \in X$ with probability *P(x)* and I have to tell you *f(x)*
- You form hypothesis $h \in H$ trying to guess my $f \in H$
- You win if *f(x)=h(x)* with probability at least 1-ε for *x* drawn according to *P(x)*.

Inductive Learning Model



- **Probably Approximately Correct (PAC) Learning Model:**
  - Take any function *f* from *H*
  - Draw *n* Training examples D_{train} from P(x), label as *y=f(x)*
  - Run learning algorithm on D_{train} to produce *h* from *H*
  - Gather Test Examples D_{test} from P(x)
  - Apply *h* to D_{test} and measure fraction (probability) of h(x)≠f(x)
  - How likely is it that error probability is less than some threshold ε (for any f from H)?

Measuring Prediction Performance

Definition: *The* **training error** $Err_{D_{train}}(h)$ *on training data* $D_{train} = ((\vec{x}_1,y_1),\ldots,(\vec{x}_n,y_n))$ *of a hypothesis h is* $Err_{D_{train}}(h) = \frac{1}{n}\sum_{i=1}^{n}\Delta(h(\vec{x}_i),y_i)$.

Definition: *The* **test error** $Err_{D_{test}}(h)$ *on test data* $D_{test} = ((\vec{x}_1,y_1),\ldots,(\vec{x}_k,y_k))$ *of a hypothesis h is* $Err_{D_{test}}(h) = \frac{1}{n}\sum_{i=1}^{k}\Delta(h(\vec{x}_i),y_i)$.

Definition: *The* **prediction/generalization/true error** $Err_P(h)$ *of a hypothesis h for a target function f over distribution P(X) is*

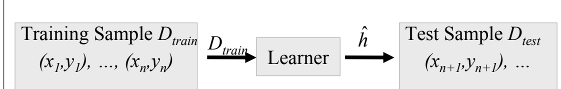
$$Err_P(h) = \sum_{\vec{x}\in X}\Delta(h(\vec{x}),f(\vec{x}))P(X=\vec{x}).$$

Definition: *The* zero/one-loss function $\Delta(a,b)$ *returns* 1 *if* $a \neq b$ *and* 0 *otherwise.*

Generalization Error Bound: Finite H, Zero Training Error

- **Model and Learning Algorithm**
  - Learning Algorithm *A* with a finite hypothesis space *H*
  - Sample of *n* labeled examples D_{train} drawn according to *P(x)*
  - Target function $f \in H$
    → At least one $h \in H$ has zero training error $Err_{D_{train}}(h)$
  - Learning Algorithm *A* returns zero training error hypothesis $\hat{h}$
- **What is the probability δ that the prediction error of $\hat{h}$ is larger than ε?**

$$P(Err_P(\hat{h}) \geq \epsilon) \leq |H|e^{-\epsilon n}$$



Useful Formulas

- **Binomial Distribution:** The probability of observing x heads in a sample of n independent coin tosses, where in each toss the probability of heads is p, is

$$P(X=x|p,n)=\frac{n!}{r!(n-x)!}p^x(1-p)^{n-x}$$

- **Union Bound:**

$$P(X_1=x_1 \vee X_2=x_2 \vee \ldots \vee X_n=x_n) \leq \sum_{i=1}^{n} P(X_i=x_i)$$

- **Unnamed:**

$$(1-\epsilon) \leq e^{-\epsilon}$$

Sample Complexity: Finite H, Zero Training Error

- **Model and Learning Algorithm**
  - Sample of n labeled examples D_{train}
  - Learning Algorithm A with a finite hypothesis space H
  - At least one $h \in H$ has zero training error $Err_{D_{train}}(h)$
  - Learning Algorithm L returns zero training error hypothesis $\hat{h}$
- **How many training examples does A need so that with probability at least $(1-\delta)$ it learns an $\hat{h}$ with prediction error less than ε?**

$$n \geq \frac{1}{\epsilon}(\ln(|H|) - \ln(\delta))$$

Training Sample D_{train} $(x_1,y_1), \ldots, (x_n,y_n)$ → D_{train} → Learner → $\hat{h}$ → Test Sample D_{test} $(x_{n+1},y_{n+1}), \ldots$

Example: Smart Investing

Task: Pick stock analyst based on past performance.

Experiment:
- Have analyst predict "next day up/down" for 10 days.
- Pick analyst that makes the fewest errors.

Situation 1:
- 1 stock analyst {A1}, A1 makes 5 errors

Situation 2:
- 3 stock analysts {A1,B1,B2}, B2 best with 1 error

Situation 3:
- 1003 stock analysts {A1,B1,B2,C1,…,C1000}, C543 best with 0 errors

Which analysts are you most confident in, A1, B2, or C543?

Useful Formula

- **Hoeffding/Chernoff Bound:**

For any distribution P(X) where X can take the values 0 and 1, the probability that an average of an i.i.d. sample deviates from its mean p by more than ε is bounded as

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} x_i - p\right| > \epsilon\right) \leq 2e^{-2n\epsilon^2}$$

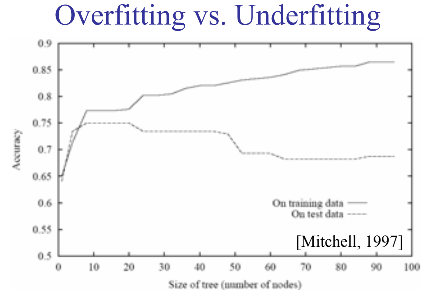
Generalization Error Bound: Finite H, Non-Zero Training Error

- **Model and Learning Algorithm**
  - Sample of n labeled examples D_{train}
  - Unknown (random) fraction of examples in D_{train} is mislabeled (noise)
  - Learning Algorithm A with a finite hypothesis space H
  - A returns hypothesis $\hat{h}=A(S)$ with lowest training error
- **What is the probability δ that the prediction error of $\hat{h}$ exceeds the fraction of training errors by more than ε?**

$$P\left(\left|Err_{D_{train}}(h_{\mathcal{A}(D_{train})}) - Err_P(h_{\mathcal{A}(D_{train})})\right| \geq \epsilon\right) \leq 2|H|e^{-2\epsilon^2 n}$$

Training Sample D_{train} $(x_1,y_1), \ldots, (x_n,y_n)$ → D_{train} → Learner → $\hat{h}$ → Test Sample D_{test} $(x_{n+1},y_{n+1}), \ldots$

Overfitting vs. Underfitting



With probability at least $(1-\delta)$:

$$Err_P(h_{\mathcal{A}(D_{train})}) \leq Err_{D_{train}}(h_{\mathcal{A}(D_{train})}) + \sqrt{\frac{1}{2n}(\ln(2|H|) - \ln(\delta))}$$

Generalization Error Bound: Infinite H, Non-Zero Training Error

- **Model and Learning Algorithm**
  - Sample of *n* labeled examples D_{train}
  - Learning Algorithm *A* with a hypothesis space *H* with *VCDim(H)=d*
  - *A* returns hypothesis $\hat{h}=A(S)$ with lowest training error
- **Definition:** *The* **VC-Dimension of** *H is equal to the maximum number d of examples that can be split into two sets in all* 2^d *ways using functions from H (shattering).*
- **Given hypothesis space *H* with *VCDim(H)* equal to *d* and a training sample D_{train} of size *n*, with probability at least $(1-\delta)$ it holds that**

$$Err_P(h_{\mathcal{A}(D_{train})}) \leq Err_{D_{train}}(h_{\mathcal{A}(D_{train})}) + \sqrt{\frac{d\left(\ln\frac{2n}{d}+1\right)-\ln\frac{\delta}{4}}{n}}$$

This slide is not relevant for exam.