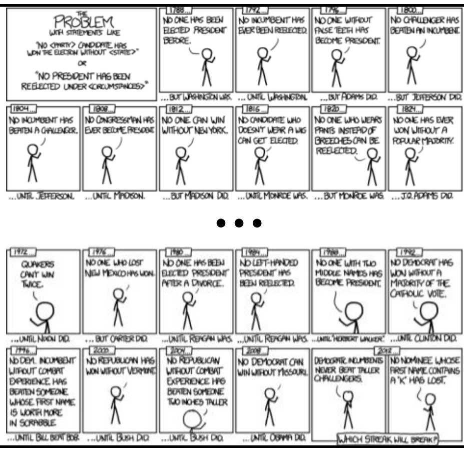


http://imgs.xkcd.com/comics/electoral_precedent.png

Statistical Learning Theory

CS4780/5780 – Machine Learning
Fall 2013

Thorsten Joachims
Cornell University

Reading: Mitchell Chapter 7 (not 7.4.4 and 7.5)

Outline

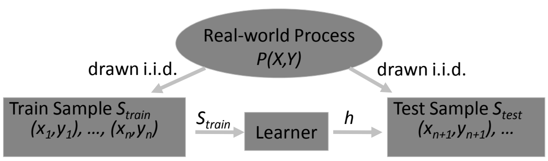
Questions in Statistical Learning Theory:
- How good is the learned rule after n examples?
- How many examples do I need before the learned rule is accurate?
- What can be learned and what cannot?
- Is there a universally best learning algorithm?

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
- Infinite hypothesis spaces and VC dimension

Can you Convince me of your Psychic Abilities?

- Game
  - I think of n bits
  - If somebody in the class guesses my bit sequence, that person clearly has telepathic abilities – right?
- Question:
  - If at least one of |H| players guesses the bit sequence correctly, is there any significant evidence that he/she has telepathic abilities?
  - How large would n and |H| have to be?

Discriminative Learning and Prediction Reminder



- Goal: Find *h* with small prediction error $Err_P(h)$ over *P(X,Y)*.
- Discriminative Learning: Given H, find *h* with small error $Err_{S_{train}}(h)$ on training sample S_{train}.

- Training Error: Error $Err_{S_{train}}(h)$ on training sample.
- Test Error: Error $Err_{S_{test}}(h)$ on test sample is an estimate of $Err_P(h)$

Review of Definitions

Definition: *A particular* **instance of a learning problem** *is described by a probability distribution* $P(X,Y)$.

Definition: *A* **sample** $S = ((\vec{x}_1, y_1), ..., (\vec{x}_n, y_n))$ *is independently identically distributed (i.i.d.) according to* $P(X,Y)$.

Definition: *The* **error on sample** S $Err_S(h)$ *of a hypothesis* h *is* $Err_S(h) = \frac{1}{n}\sum_{i=1}^{n} \Delta(h(\vec{x}_i), y_i)$.

Definition: *The* **prediction/generalization/true error** $Err_P(h)$ *of a hypothesis* h *for a learning task* $P(X,Y)$ *is*

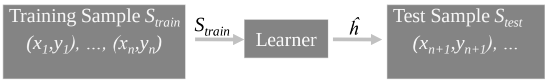
$$Err_P(h) = \sum_{\vec{x} \in X, y \in Y} \Delta(h(\vec{x}), y) P(X = \vec{x}, Y = y).$$

Definition: *The* **hypothesis space H** *is the set of all possible classification rules available to the learner.*

Generalization Error Bound: Finite H, Zero Error

- Setting
  - Sample of n labeled instances S_{train}
  - Learning Algorithm *L* with a finite hypothesis space *H*
  - At least one $h \in H$ has zero prediction error $Err_P(h)=0$ (→ $Err_{S_{train}}(h)=0$)
  - Learning Algorithm *L* returns zero training error hypothesis $\hat{h}$
- What is the probability 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

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

- Union Bound:

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

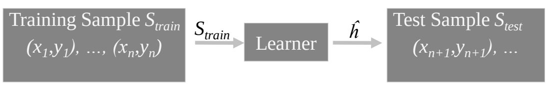
- Unnamed:

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

Sample Complexity: Finite H, Zero Error

- Setting
  - Sample of n labeled instances S_{train}
  - Learning Algorithm *L* with a finite hypothesis space *H*
  - At least one $h \in H$ has zero prediction error (→ $Err_{S_{train}}(h)=0$)
  - Learning Algorithm *L* returns zero training error hypothesis $\hat{h}$
- How many training examples does L need so that with probability at least (1-δ) it learns an $\hat{h}$ with prediction error less than ε?

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



Probably Approximately Correct Learning

Definition: *C is* **PAC-learnable** *by learning algorithm $\mathcal{L}$ using H and a sample S of n examples drawn i.i.d. from some fixed distribution $P(X)$ and labeled by a concept $c \in C$, if for sufficiently large n*

$$P(Err_P(h_{\mathcal{L}(S)}) \leq \epsilon) \geq (1-\delta)$$

for all $c \in C, \epsilon > 0, \delta > 0$, and $P(X)$. $\mathcal{L}$ is required to run in polynomial time dependent on $1/\epsilon, 1/\delta, n$, the size of the training examples, and the size of c.