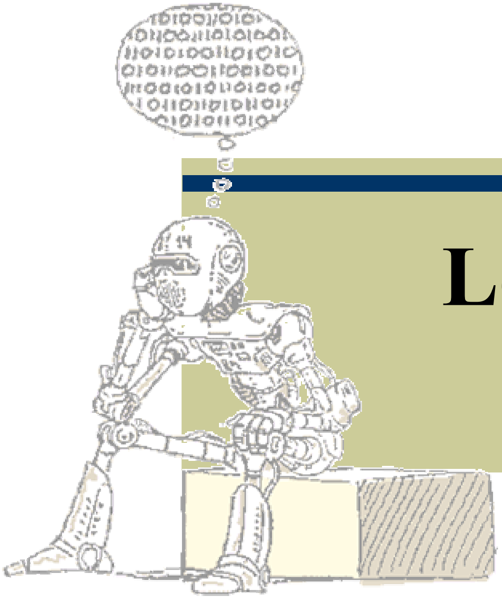


Lecture V: Learning for Control

- *Model Selection*

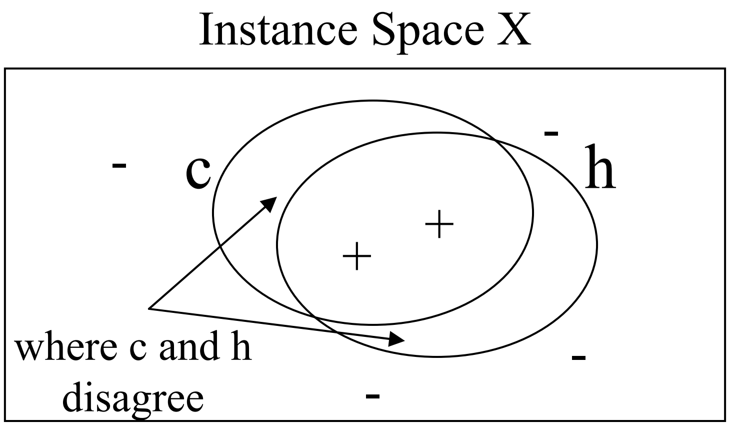
Overview

- Model Complexity
- Model Selection & Regularization Theory
  - Crossvalidation & Other Techniques
  - MDL, Occam’s Razor
  - Bayesian Model Selection

Concept Learning Example

- **Given.**
  - **Instances X** : Possible days, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast.*
  - **Target Function c** in ***C***: EnjoySport : X->{0,1}
  - **Hypotheses *H*:** Conjunction of literals. E.g. $< Cold, High, ?, ?, ? >$
  - **Training Examples *D***: Positive and negative examples of target function.<{x1,c(x1)},…{xm, c(xm)}
- **Determine.**
  - A hypothesis ***h*** in ***H*** such that ***h****(x)*=***c****(x)* for all x in ***D*** ?
  - A hypothesis ***h*** in ***H*** such that ***h****(x)*=***c****(x)* for all x in ***X*** ?

True Error of Hypothesis



Definition: The **true error** (denoted $error_D(h)$) of hypothesis h with respect to target concept ***c*** and distribution ***D*** is the probability that ***h*** will misclassify an instance drawn at random according to ***D***

$$error_D(h) \equiv \Pr_{x \in D}[c(x) \neq h(x)]$$

Two notions of error

- ***Training error*** of hypothesis ***h*** with respect to target concept ***c***

  $How\ often\ h(x) \neq c(x)\ over\ training\ instances.$

- ***True error*** of hypothesis ***h*** with respect to target concept ***c***

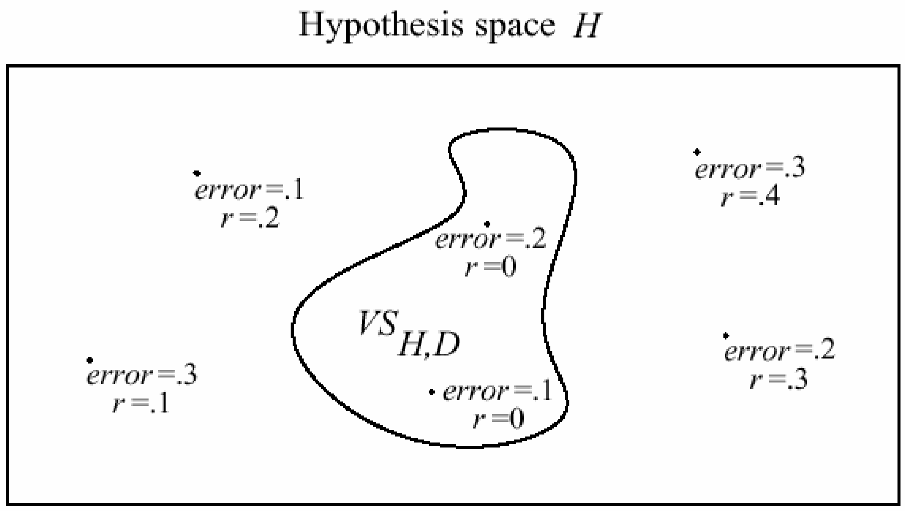
  $How\ often\ h(x) \neq c(x)\ over\ future\ random\ instances.$

Our concern

- Can we bound the true error of ***h*** given the training error of ***h*** ?
- Consider the case when the training error of h is zero, or h belongs to the version space of D i.e. $h \in VS(D)$

$$VS_{H,D} = \left\{h \in H \mid (\forall \langle x, c(x) \rangle \in D)(h(x) = c(x))\right\}$$

Exhausting the Version Space



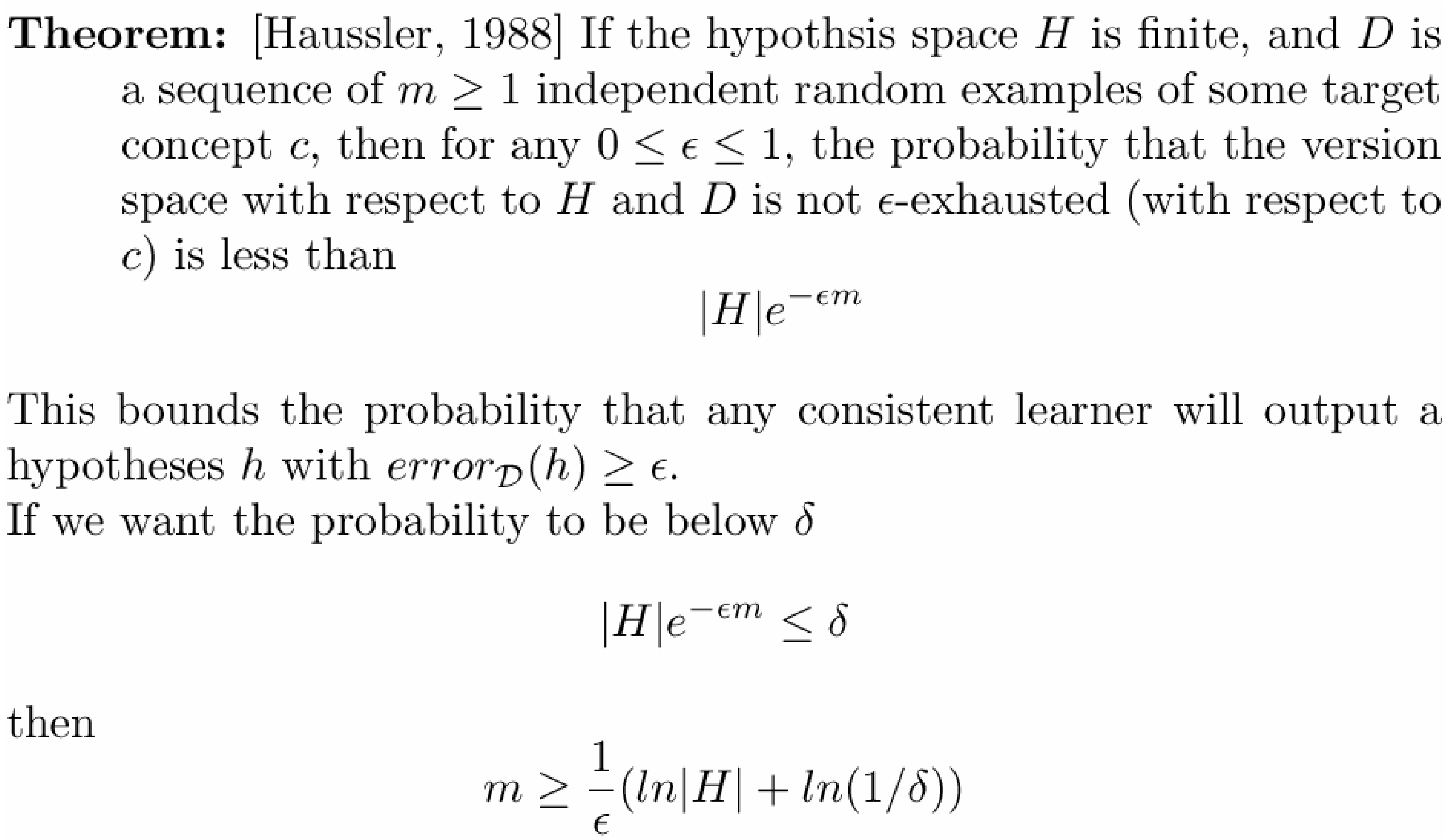
(r = training error, $error$ = true error)

Definition: The version space $VS_{H,D}$ is said to be ϵ-exhausted with respect to c and $\mathcal{D}$, if every hypothesis h in $VS_{H,D}$ has error less than ϵ with respect to c and $\mathcal{D}$.

$$(\forall h \in VS_{H,D}) error_{\mathcal{D}}(h) < \epsilon$$

Sample Complexity

What is the number of samples needed to ε-exhaust the VS ??



Theorem: [Haussler, 1988] If the hypothsis space H is finite, and D is a sequence of $m \geq 1$ independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to H and D is not ϵ-exhausted (with respect to c) is less than

$$|H|e^{-\epsilon m}$$

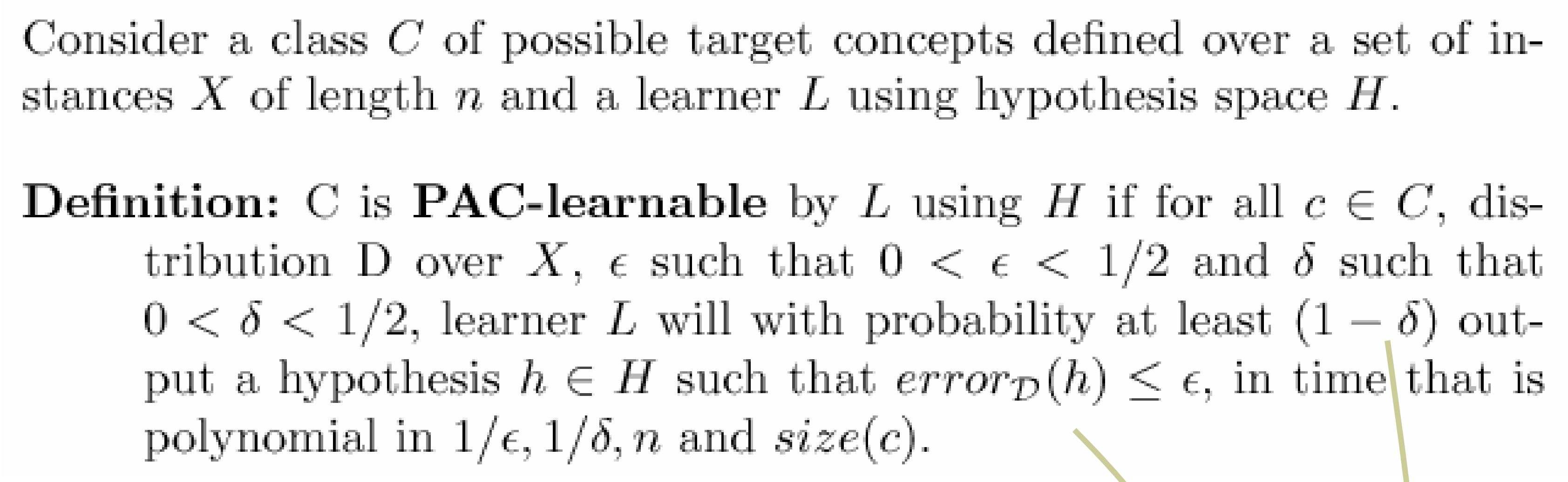
This bounds the probability that any consistent learner will output a hypotheses h with $error_{\mathcal{D}}(h) \geq \epsilon$.
If we want the probability to be below δ

$$|H|e^{-\epsilon m} \leq \delta$$

then

$$m \geq \frac{1}{\epsilon}(ln|H| + ln(1/\delta))$$

PAC Learnability



Consider a class C of possible target concepts defined over a set of instances X of length n and a learner L using hypothesis space H.

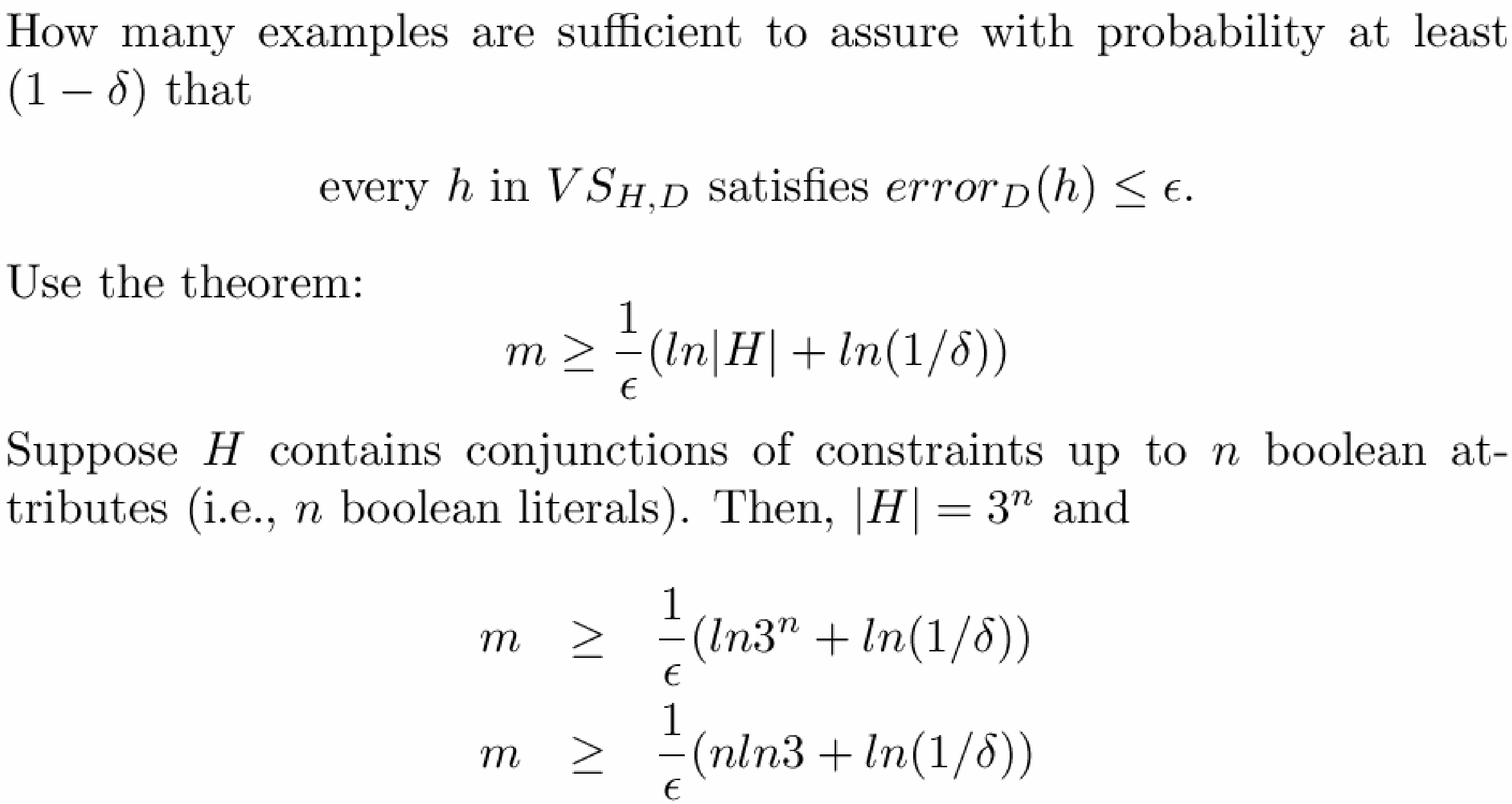
Definition: C is **PAC-learnable** by L using H if for all $c \in C$, distribution D over X, ϵ such that $0 < \epsilon < 1/2$ and δ such that $0 < \delta < 1/2$, learner L will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon, 1/\delta, n$ and $size(c)$.

Approximately

Probably

PAC=Probably Approximately Correct

PAC learnability of Boolean literals



How many examples are sufficient to assure with probability at least $(1-\delta)$ that

every h in $VS_{H,D}$ satisfies $error_D(h) \leq \epsilon$.

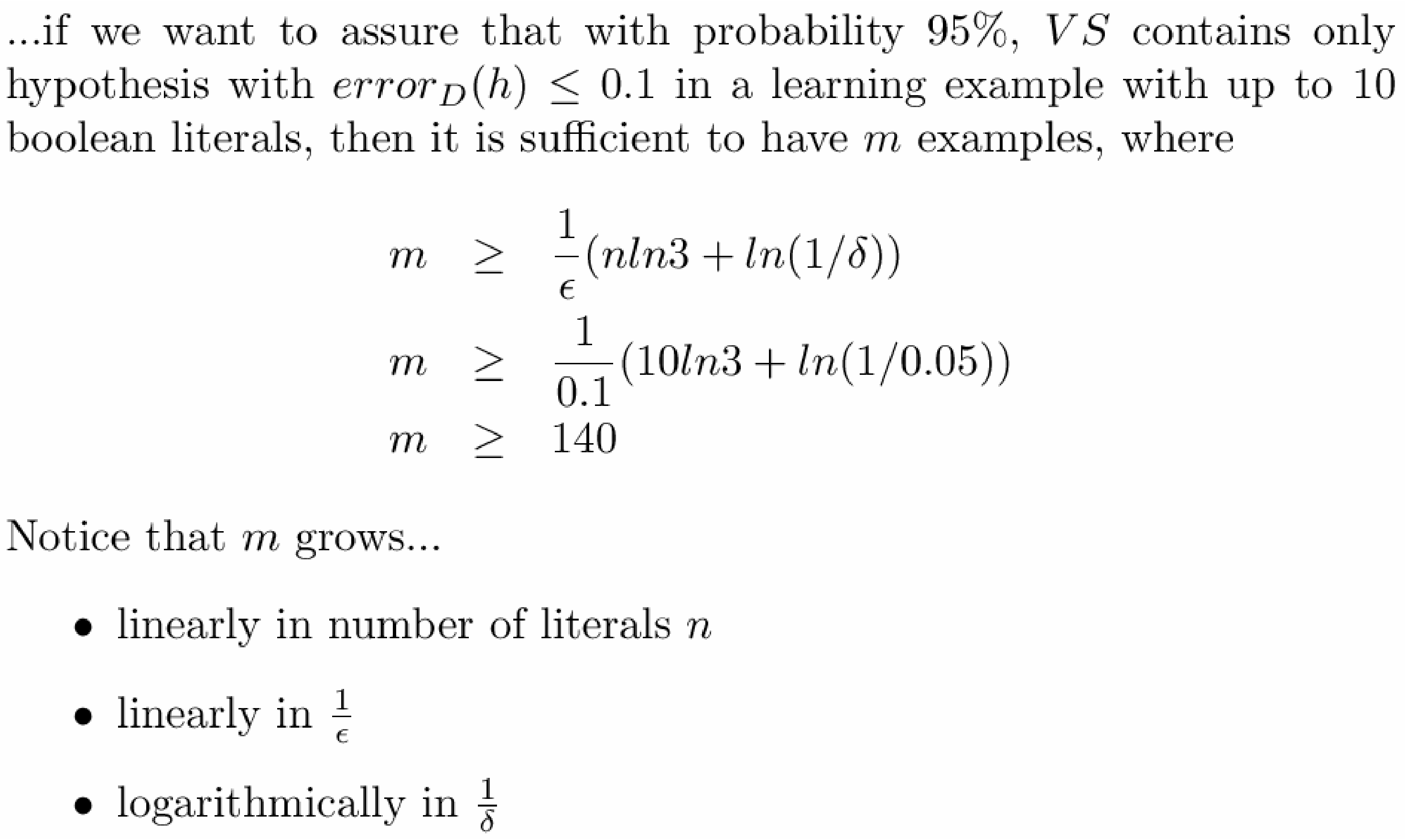
Use the theorem:

$$m \geq \frac{1}{\epsilon}(ln|H| + ln(1/\delta))$$

Suppose H contains conjunctions of constraints up to n boolean attributes (i.e., n boolean literals). Then, $|H| = 3^n$ and

$$\begin{aligned} m &\geq \frac{1}{\epsilon}(ln3^n + ln(1/\delta)) \\ m &\geq \frac{1}{\epsilon}(nln3 + ln(1/\delta)) \end{aligned}$$

PAC learnability of Boolean literals: An example



...if we want to assure that with probability 95%, VS contains only hypothesis with $error_D(h) \leq 0.1$ in a learning example with up to 10 boolean literals, then it is sufficient to have m examples, where

$$\begin{aligned} m &\geq \frac{1}{\epsilon}(nln3 + ln(1/\delta)) \\ m &\geq \frac{1}{0.1}(10ln3 + ln(1/0.05)) \\ m &\geq 140 \end{aligned}$$

Notice that m grows...

- linearly in number of literals n
- linearly in $\frac{1}{\epsilon}$
- logarithmically in $\frac{1}{\delta}$

Other Measures of Model Complexity

- **VC Dimension [Vapnik-Chernovenkis]**
  - Provides a general measure of complexity of a learning system.
  - Provides bounds on optimism (on errors expected from the system).

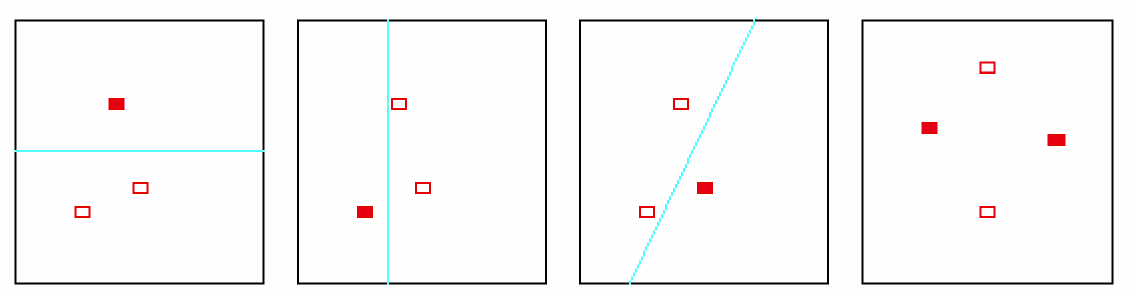
Assume a class of indicator function: $\{f(x,\alpha)\}$ *with parameters* α *and* $x \in \Re^p$

Definition: ***The VC dimension of the class*** **{f(x,α)}** ***is defined to be the largest number of points (in some configuration) that can be shattered by members of*** **{f(x,α)} .**

Shatter: A set of points is said to be shattered by a class of functions if, no matter how we assign a binary label to each point, a member of the class can perfectly separate them.

Note: Hence, if VC dimension is ***d***, then there exists a set of ***d*** points that can be shattered but there is no set of ***d+1*** points that can be shattered

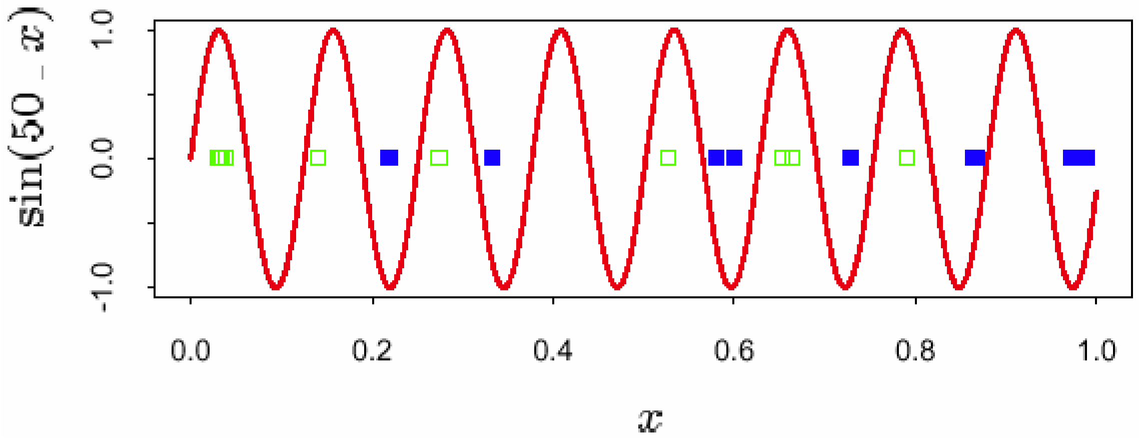
VC Dimension (cont'd)



Example: Class of lines in a plane can shatter 3 points with arbitrary labeling. However, any configuration of 4 points that cannot be shattered with the labeling shown in panel four.

Hence, VC dimension of a line in a plane is 3.

How is it different from number of parameters ?? Check out the next example !!!



VC dimension of a indicator function sin(αx) with one parameter is **infinite**.

Sample Complexity and VC dim.

How many randomly drawn training examples suffice to PAC learn any target concept C ?

Or how many examples suffice to ε–exhaust the version space with probability (1-δ) ?

Using VC dim. as a measure of complexity of H [Blumer et al., 89] :

$$m \geq \frac{1}{\varepsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\varepsilon))$$

Recall and compare bounds using |H| (size of hypothesis space)**:**

$$m \geq \frac{1}{\varepsilon}(\ln|H| + \ln(1/\delta))$$

Comparisons:

- VC(H) bounds are defined even for infinite hypothesis spaces
- Usually, the bounds using VC(H) are tighter than those using |H|

VC Dimension (cont'd)

VC dimension of real valued functions {g(x,α)} : is the VC dimension of the indicator class {I(g(x,α)-β>0)} , where β takes the values over the range of g.

Estimation of Bounds on Prediction Error based on VC dimension (h)

$$L(\alpha) = \int \frac{1}{2} |y - f(x,\alpha)| dP(x,y)$$: Generalization Error

$$L_{emp}(\alpha) = \frac{1}{2N} \sum_{i=1}^{N} |y_i - f(x_i,\alpha)|$$: Empirical (training) Error

For a machine with VC dim=h, the following bound holds with probability1-η

$$L(\alpha) \leq L_{emp}(\alpha) + \sqrt{\frac{h(\log\frac{2N}{h}+1) - \log\frac{\eta}{4}}{N}}$$: Error Bounds

Model Selection/Assessment

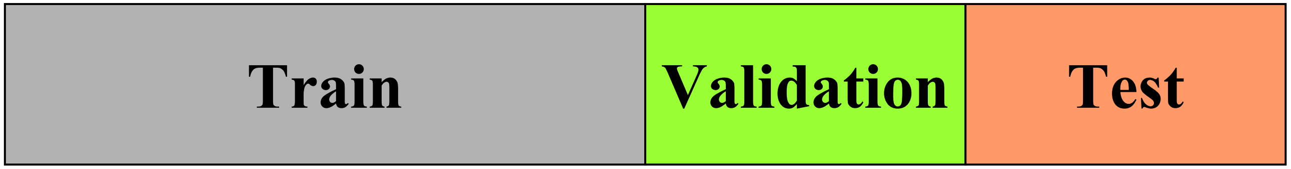
Model Selection

Estimating the performance of different models in order to choose the (approximate) best one.

Model Assessment

Having chosen a final model, estimating it's prediction (generalization) error

If we are in a data rich environment, then divide the data set into :



Train | Validation | Test

Otherwise,

- *we **recycle** the validation set as the test set (**efficient sample reuse**) - at the expense of underestimating the true test error of the chosen model.*
- *Use **analytical methods** to approximate the validation step*

Model Selection Procedures

- **Crossvalidation**
- **Bootstrap**

(***Efficient sample reuse methods***)
Makes no prior assumption about the models

- **Regularization**
- **Structural Risk Minimization**
- **Minimum Description Length**
- **Bayesian Model Selection / BIC**

(***Analytical + some sample reuse***)
Incorporates prior knowledge about the models under consideration and aims for simpler models

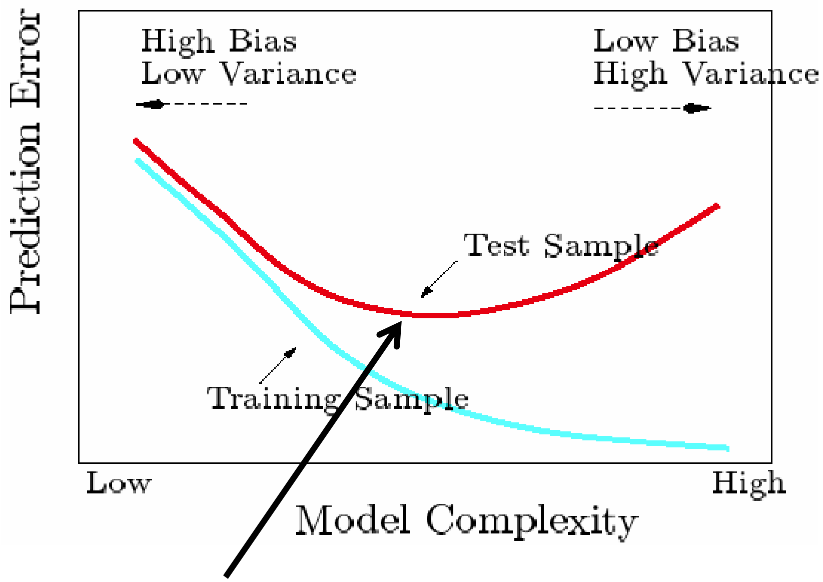
Model Selection Procedures (I)

- **Crossvalidation**

Divide a given data set into two parts – a *training* set and *testing (validation)* test. Train models of various complexity and test their *generalization* on the validation set.

+ ***n-fold*** crossvalidation and ***leave-one-out*** crossvalidation

A typical performance curve of the [errors vs. model complexity] looks like this:

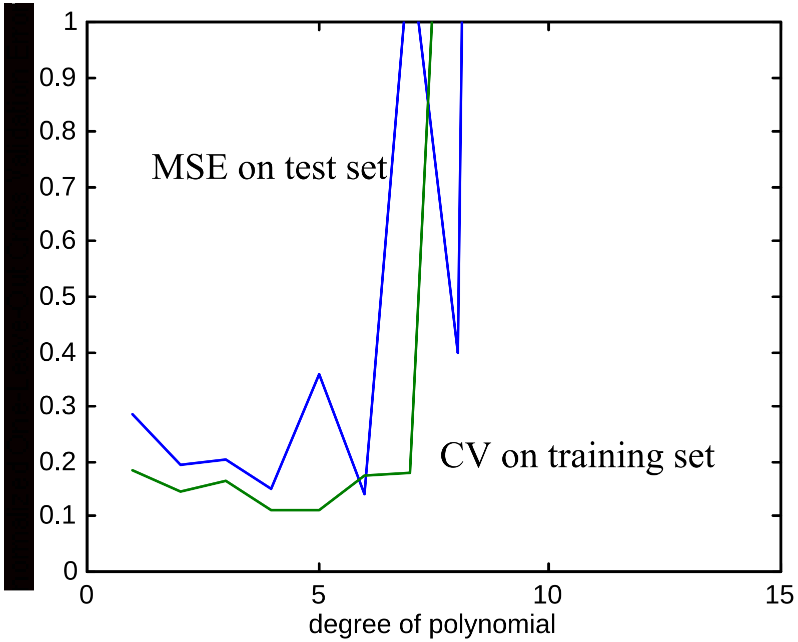


Training set error keeps decreasing as the model complexity increases. Testing/Validation set error (typically) decreases initially and then, starts to increase again.

The “elbow” corresponds to the optimal complexity

Example: Leave-one-out Crossvalidation

- **Notation for leave-one-out CV error:** $J_{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_{i,-i}\right)^2$



Note that this can be computationally very expensive for some network, and very cheap for others.

=> optimal degree of polynomial is 4 or 5

Model Selection Procedures (II)

- **Regularization**

In this approach, we write an ***augmented error function*** :

E = error on data + λ • model complexity

The second term penalizes the complex models with large variance. λ is the parameter specifying the weight of the penalty. Too large λ results in strong bias. λ is optimized (usually) using crossvalidation.

For instance: add penalty term to MSE cost function, e.g., a smoothness prior

$$\tilde{J} = J + \gamma P$$

$$P = \int \left(\frac{d^2 y}{dx^2} \right)^2 dx$$

We will see more instances and examples of various types of regularization at appropriate junctures along the course…

Model Selection Procedures (III)

Structural Risk Minimization (SRM) [Vapnik]

- Assumes we have a set of models ordered in terms of their complexity.
  - Polynomials of increasing degree (complexity = number of free parameters)
  - Models ordered according to VC dimension ($h_1 < h_2 < \ldots$)
- SRM corresponds to finding the model that is ***simplest in the order of complexity*** while while performing the best in terms of empirical (training) error.

*A very successful application of the SRM principle is the **Support Vector Machines (SVM)** – a paradigm based on the principle of **maximizing margins** & in turn **reducing the VC dimension** of approximation candidates.*

Model Selection Procedures (IV)

Minimum Description Length (MDL) [Rissanen,1978]

- Motivated from the field of Information theory – Coding length.
- Out of all the models that describe the data well, we want to have the ***simplest models*** since that lends to the shortest description length (coding) for representation.

To transmit a random variable *z* having probability density function Pr(z), we require about –log Pr(z) bits of information

Shannon’s Theorem

Applied to Model Selection: model **M**, parameter **θ**, data **D**=(**X**,**y**)

$$length = -\log \Pr(\mathbf{y} \mid \boldsymbol{\theta}, M, \mathbf{X}) - \log \Pr(\boldsymbol{\theta} \mid M)$$

Bits for transmitting discrepancy from model parameters

Bits for transmitting model parameters

Note*: The length here corresponds to the negative log posterior probability*

Minimum Description Length (cont'd)

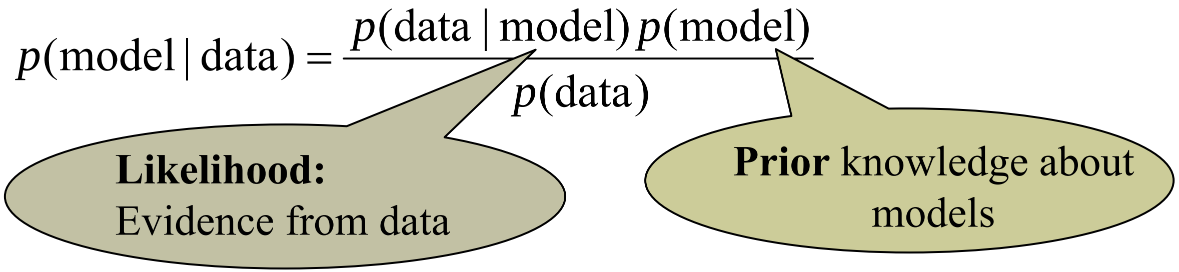
- **Possible Coding Schemes:**
  - code every data point in the data set
    - this assumes data points are independent and there is no structure in the data
  - find a code that takes advantage of the structure in the data, and that should thus be more efficient:
    - then we need to first transmit the information about the model of the data: L(M)
    - and second for every data point how much it differs from the model: L(D|M)
- **Description Length**

$$\text{description length} = \underbrace{L(D \mid M)}_{\text{error}} + \underbrace{L(M)}_{\text{complexity}}$$

- **Relationship to Machine Learning:**
    - the model of the data is the learned information
    - the error is the remaining approximation error of the learning system
    - we automatically seek the least complex model to account for the data:
      - bias & variance tradeoff

Model Selection Procedures (V)

Bayesian Model Selection and BIC



$$p(\text{model}\,|\,\text{data}) = \frac{p(\text{data}\,|\,\text{model})\,p(\text{model})}{p(\text{data})}$$

Likelihood:
Evidence from data

Prior knowledge about models

- For a given model, we can evaluate the likelihood of observing the data

- Here, we incorporate the our prior knowledge about the likelihood of the models

Bayesian Model Selection involves either choosing the model with the ***largest*** posterior probability or taking an ***average over all models*** weighted by their posterior probabilities.

Posterior probabilities

$$\begin{aligned}\Pr(M_m\,|\,\mathbf{D}) &\propto \Pr(M_m)\cdot\Pr(\mathbf{D}\,|\,M_m)\\ &\propto \Pr(M_m)\cdot\int \Pr(\mathbf{D}\,|\,\theta_M, M_m)\Pr(\theta_M\,|\,M_m)d\theta_M\end{aligned}$$

Bayesian Information Criterion (cont'd)

- **Bayesian Information Criterion**

$$\Pr(M_m \mid \mathbf{D}) \propto \Pr(M_m)\cdot\Pr(\mathbf{D}\mid M_m)$$
$$\propto \Pr(M_m)\cdot\int \Pr(\mathbf{D}\mid\theta_M, M_m)\Pr(\theta_M \mid M_m)d\theta_M$$

To compare two models:

$$\frac{\Pr(M_m\mid\mathbf{D})}{\Pr(M_l\mid\mathbf{D})} = \frac{\Pr(M_m)}{\Pr(M_l)}\cdot\frac{\Pr(\mathbf{D}\mid M_m)}{\Pr(\mathbf{D}\mid M_l)}$$

Denoted as BF(**D**) and called the ***Bayes factor*** : the contribution of the data towards the posterior odds.

Usually, the prior probabilities are treated as uniform in the absence of any other biases.

We can approximate Pr(D|M) using the Laplace approximation to the integral and get:

$$\log\Pr(\mathbf{D}\mid M_m) = \log\Pr(\mathbf{D}\mid\hat{\theta}_M, M_m) - \frac{d_m}{2}\log N$$

$$d_m = \mathit{number\ of\ free\ parameters\ in\ model}\ M_m,\quad \hat{\theta}_M = \mathit{max. likelihood\ estimate}$$

BIC: Maximize posterior = minimize BIC

$$\boxed{\max.\left\{\log\Pr(\mathbf{D}\mid\hat{\theta}_M, M_m) - \frac{d_m}{2}\log N\right\} = \min.\left\{-2\cdot\log\Pr(\mathbf{D}\mid\hat{\theta}_M, M_m) + d_m\log N\right\}}$$