

## Introduction to learning

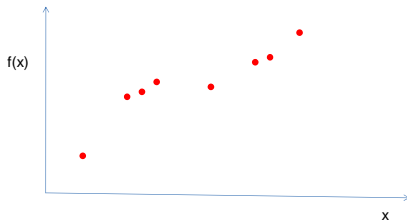
Devika Subramanian  
Comp 540  
(adapted from Lecture 2 of T. Poggio's  
9.250 course at MIT)

## Outline

- Learning = function approximation
- Empirical risk minimization
- Generalization and well-posedness
- Regularization

## Supervised Learning

- Supervised learning is a problem of approximating a function from sparse data.



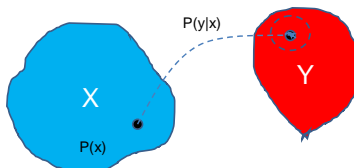
## Input to supervised learning

- Given a training set  $D$

$$D = \{(x^{(i)}, y^{(i)}) \mid x^{(i)} \in X, y^{(i)} \in Y, i = 1..m\}$$

- where training examples  $(x^{(i)}, y^{(i)})$  are drawn i.i.d. from a fixed but unknown probability distribution  $P(z)$  on  $Z = X \times Y$

## Data generation view



Data is generated by a fixed but unknown probability distribution.

## Output of supervised learning

- We need to define a hypothesis space  $H$  of functions that we use to represent the relationship between the  $y$ 's and the  $x$ 's, i.e.,  
 $y = f(x)$ , where  $f$  can be a deterministic or stochastic function.

## Learning = function approximation from data

- The goal of supervised learning is to use training set  $D$  to “learn” a function  $f$ , that can take a new  $x$  value, and predict the associated value of  $y$ .

$$y_{pred} = f(x_{new})$$

- If  $y$  is a real-valued random variable, we have a **regression** problem.
- If  $y$  takes values from an unordered finite set, we have a **classification** problem.

## Loss functions

- To measure how good our learned function  $f$  is, we need a loss function  $L$ . A loss function denotes the price we pay when we guess  $f(x)$  for input  $x$  and its true value is  $y$ .

## Loss functions for regression

- L2 loss or squared error

$$L(f(x), y) = (f(x) - y)^2$$

- L1 loss or absolute error

$$L(f(x), y) = |f(x) - y|$$

## Loss functions for classification

- For binary classification, we use 0-1 loss

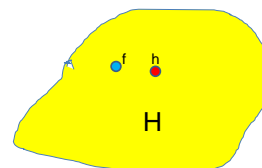
$$L(f(x), y) = \begin{cases} 1 & \text{if } f(x) \neq y \\ 0 & \text{otherwise} \end{cases}$$

## Learning problem: summary

- There is a fixed, but unknown probability distribution  $P$  on the product space  $Z = X \times Y$
- We assume that  $X$  is a compact domain in Euclidean space and that  $Y$  is a closed subset of the reals.
- The training set  $D = \{(x^{(i)}, y^{(i)}) \mid x^{(i)} \in X, y^{(i)} \in Y, i = 1..m\}$  consists of  $m$  samples drawn i.i.d. from  $P$ .
- The hypothesis space  $H$  is a set of functions  $f: X \rightarrow Y$ .

## Learning algorithm

- A learning algorithm uses  $D$  to select from  $H$  a function  $h: X \rightarrow Y$  such that  $h(x)$  is approximately equal to  $y$  for new pairs  $(x, y)$ .



## True error and empirical error

- Given a function  $f$ , a probability distribution  $P$ , and a loss function  $L$ , the expected or true error of  $f$  is:

$$E[f] = \int_{x,y} L(f(x), y) dP(x, y)$$

- Since we do not know  $P$ , we settle for error on the training set, or empirical error

$$E_D[f] = \frac{1}{m} \sum_{i=1}^m L(f(x^{(i)}), y^{(i)})$$

## Requirements for a solution

- [Distribution independent generalization]:** The empirical error of learned function  $h$  must converge to the expected error, and thus be a proxy for it.

$$\forall P, \lim_{n \rightarrow \infty} |E_D[h] - E[h]| = 0 \text{ in probability}$$

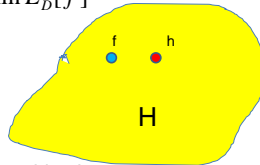
- [Universal consistency]:** error of learned function  $h$  must be "close" to that of true function  $f$ .

$$\forall \varepsilon > 0, \lim_{n \rightarrow \infty} \sup_P \Pr \{ E[h] > \inf_{f \in H} E[f] + \varepsilon \}$$

## Empirical risk minimization (ERM)

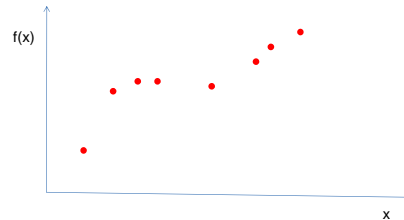
- Given a training set  $D$ , a function space  $H$ , a loss function  $L$  (which determines empirical error), empirical risk minimization selects  $h$  as

$$h = \arg \min_{f \in H} E_D[f]$$

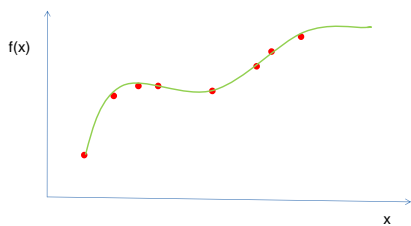


Problem: an ill-posed optimization problem!

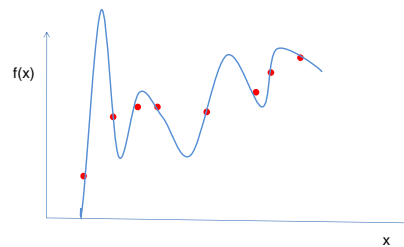
## An example



## True solution



## ERM solution



How can we guarantee that given a sufficient number of examples the ERM solution will converge to the true one?

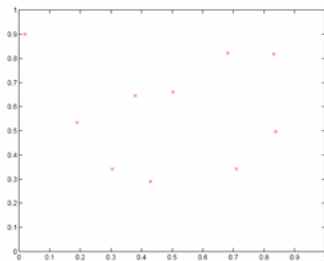
## Conditions for consistency of ERM

- A proper choice of hypothesis space  $H$  will ensure ERM produces reasonable solutions.
- Heuristic strategy: start with simplest choice of  $H$ , and introduce complexity incrementally.
- Heuristic strategy formalizable in terms of theory of regularization.

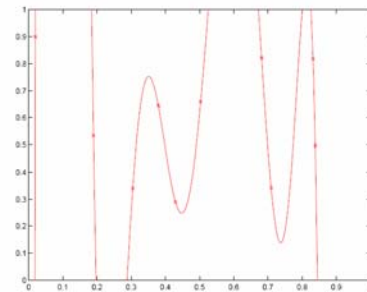
## Well and ill posed problems

- A problem is well-posed if its solution exists, is unique and depends continuously on the data (i.e., small changes in training data cause small changes in learned function).
- A problem is ill-posed if it is not well-posed.

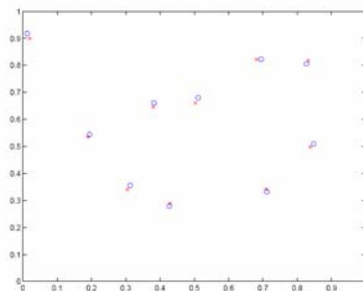
## Example of stability



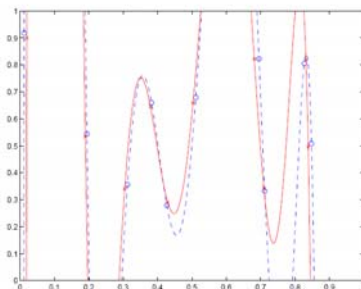
## A 10<sup>th</sup> degree polynomial fit



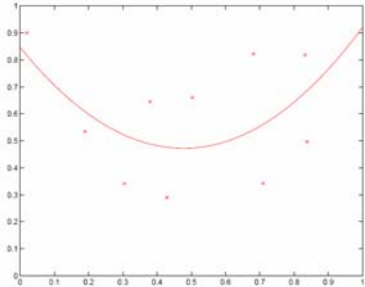
## Perturb the points slightly ...



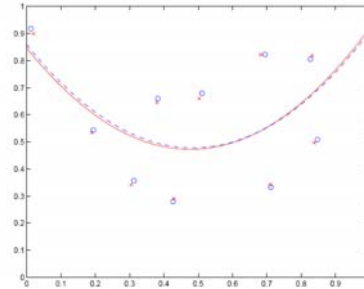
## ... the solution changes a lot



## A second degree polynomial fit



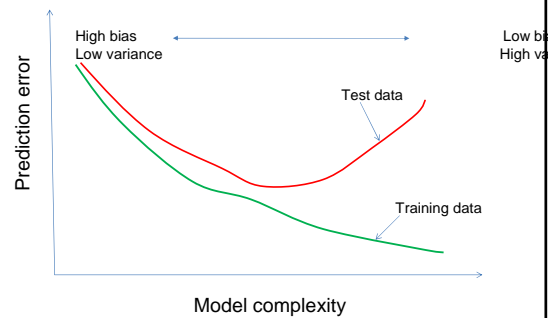
## ... solution varies very little



## Bias/variance tradeoff

- 10<sup>th</sup> degree polynomial class: low bias, high variance
- 2<sup>nd</sup> degree polynomial class: high bias, low variance

## Model complexity choice



Adapted from Figure 2.11, HTF

## Regularization

- ERM finds the function in  $H$  which minimizes

$$\frac{1}{m} \sum_{i=1}^m L(f(x^{(i)}, y^{(i)}))$$

- This is not a well-posed problem. Tikhonov regularization instead finds a function in  $H$  which minimizes

$$\frac{1}{m} \sum_{i=1}^m L(f(x^{(i)}, y^{(i)})) + \underbrace{\gamma \|f\|^2}_{\text{Penalty term}}$$

Penalty term which encodes a notion of smoothness of  $f$