Predictive Learning from Data

LECTURE SET 2

Inductive Learning, Basic Learning Problems and Inductive Principles

Cherkassky, Vladimir, and Filip M. Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.

Source: Dr. Vladimir Cherkassky (revised by Dr. Hsiang-Han Chen)

OUTLINE

2.0 Objectives + Background

- formalization of inductive learning
- classical statistics vs predictive approach

2.1 Terminology and Learning Problems

2.2 Basic Learning Methods and Complexity Control

2.3 Inductive Principles

2.4 Alternative Learning Formulations

2.5 Summary

2.0 Objectives

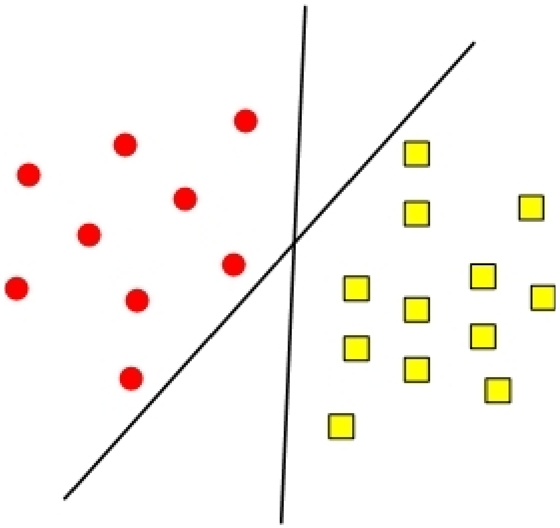
- To quantify the notions of explanation, prediction and model
- Introduce terminology
- Describe common learning problems

Example: classification problem

training samples, model

Goal 1: explanation of training data

Goal 2: generalization (for future data)



- Learning (model estimation) is ill-posed

Well posed

Three fundamental conditions:

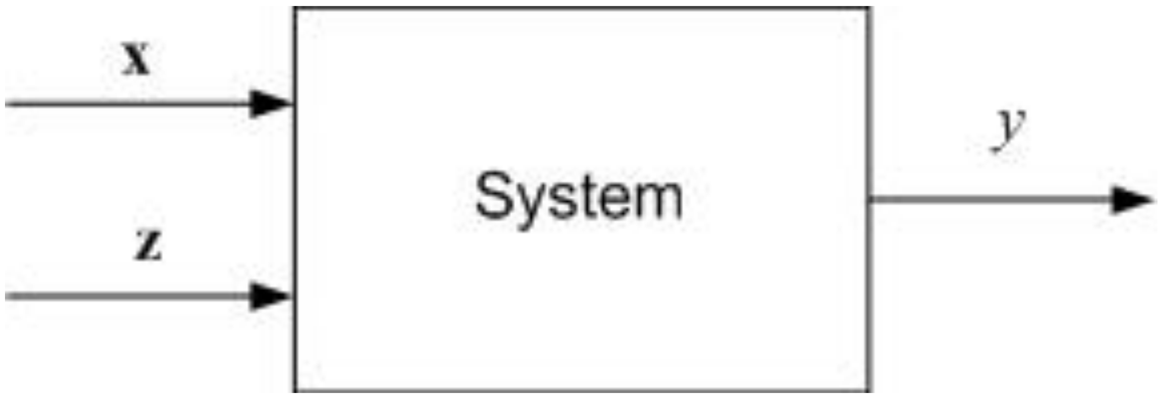
- The problem must have a unique solution.
- The solution must depend continuously on the data or the parameters.
- The solution must be stable against small changes in the data or the parameters.

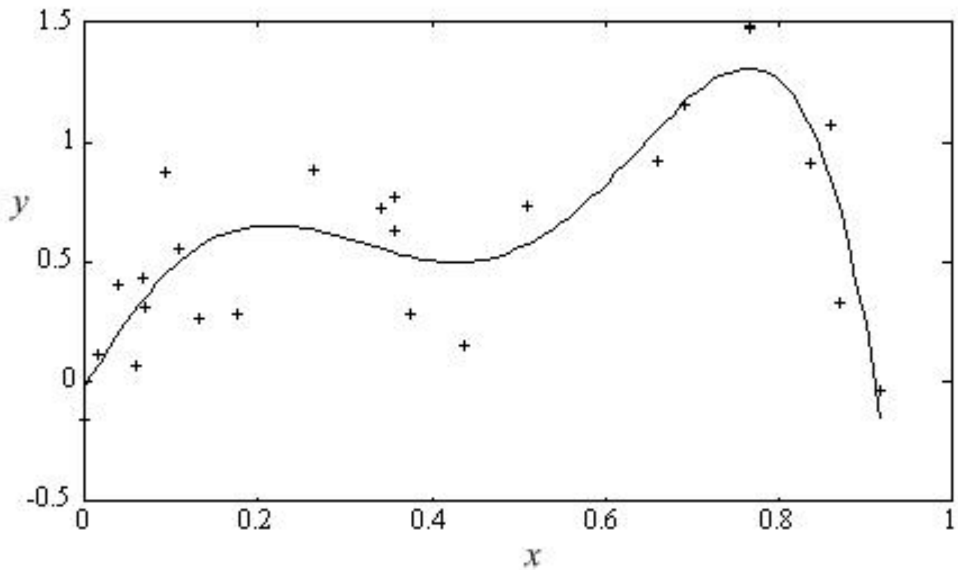
Ill posed: violate one or more of these conditions, therefore, difficult to solve.

https://www.collimator.ai/reference-guides/what-is-an-ill-posed-problem

Example: imitation of system's output (regression problem)

- Common setting ~ function estimation





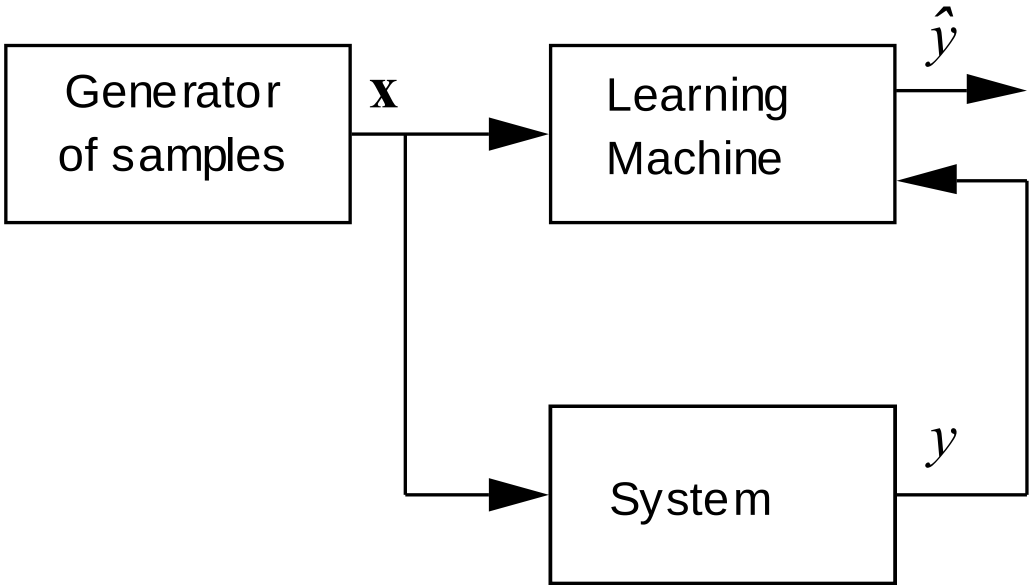
Some terminologies

Learning is the process of estimating an unknown (input, output) dependency or structure of a System using a limited number of observations.

- Past observations ~ data points
- Explanation (model) ~ function
- Learning ~ function estimation (from data)
- Prediction ~using the model to predict new inputs

General learning scenario

- Three components



- Unknown joint distribution $P(\mathbf{x},y)$
- *Set of functions* (possible models) $\boldsymbol{f}(\mathbf{x},\omega)$
- **Pre-specified** *Loss function* $L(y,\boldsymbol{f}(\mathbf{x},\omega))$

  (by convention, non-negative *L*)

Set of functions (possible models)

- Parametric regression (fixed-degree polynomial)

EX: the set of functions is specified as a polynomial of fixed degree and the training data have a single predictor variable.

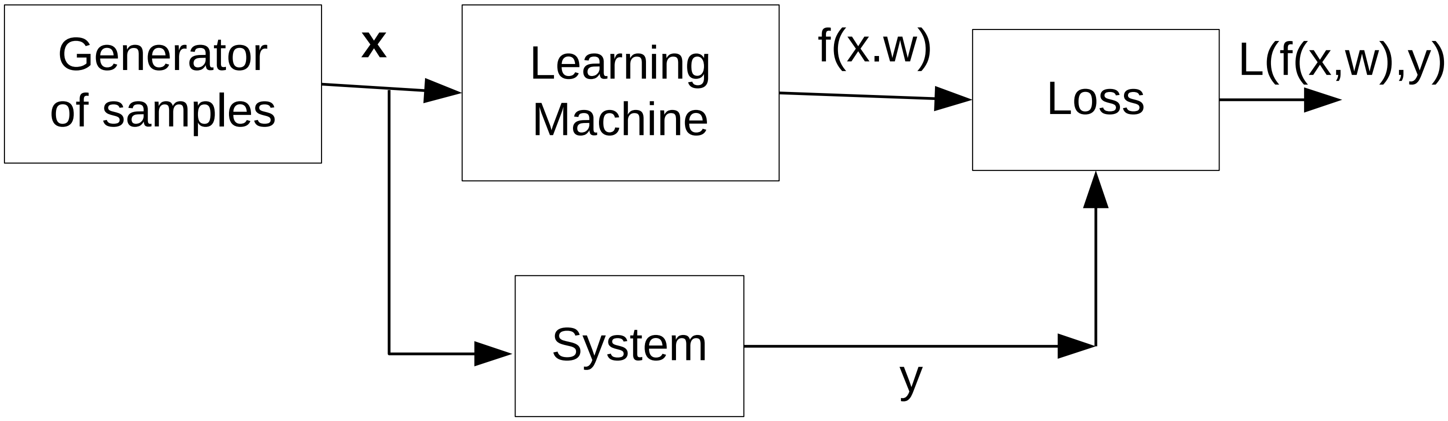
$$f(x, \mathbf{w}) = \sum_{i=0}^{M-1} w_i x^i$$

- Different *w* vectors result in different functions.

Inductive Learning Setting

- The *learning machine* observes samples (**x** ,*y*), and returns an estimated response $\hat{y} = f(\mathbf{x}, w)$
- Recall 'first-principle' vs 'empirical' knowledge

  → Two modes of inference: identification vs imitation
- Risk $\int Loss(y, f(\mathbf{x},w))\, dP(\mathbf{x},y) \rightarrow min$



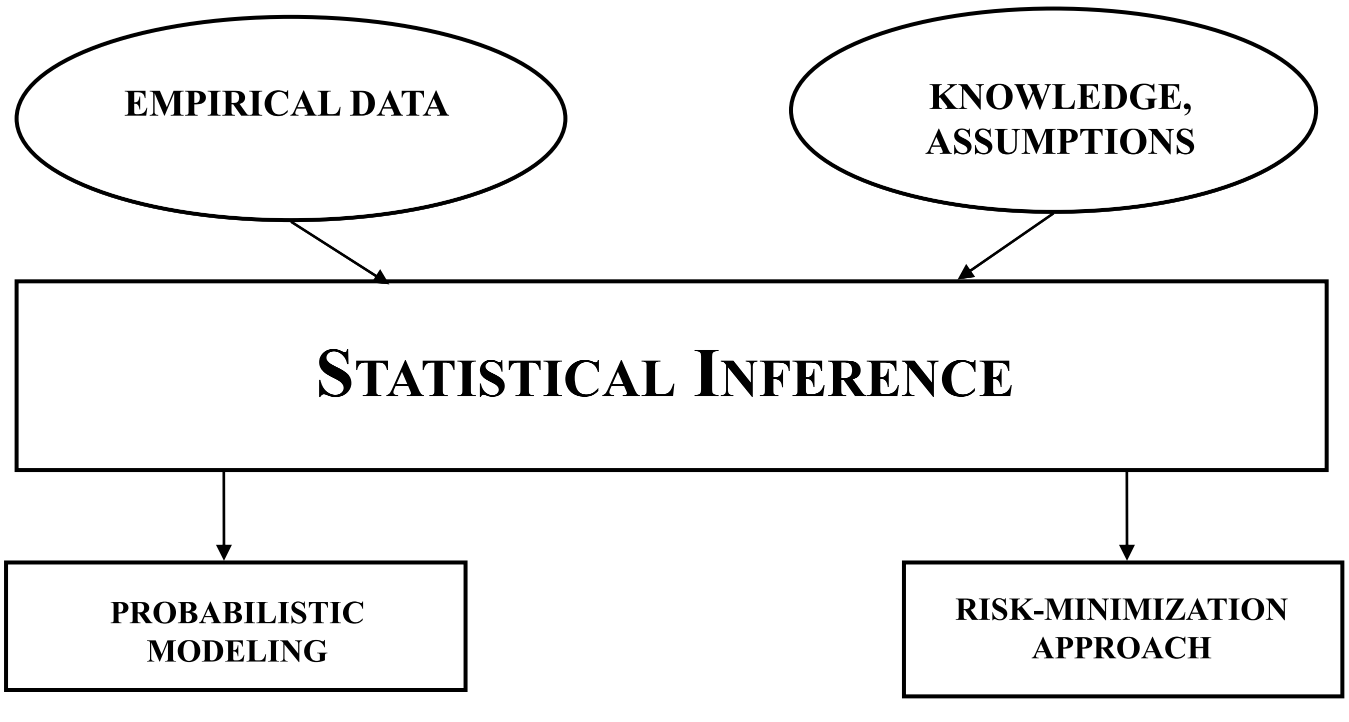
How should a Learning Machine use training data?

- **Inductive principle**: a general prescription for obtaining an estimate the "true dependency" in the class of approximating functions from the available (finite) training data.

- A **learning method** is a constructive implementation of an inductive principle for selecting an estimate $f(\mathbf{x}, \omega^*)$ from a particular set of functions $f(\mathbf{x}, \omega)$.

Two Views of Empirical Inference

- Two approaches to empirical or statistical inference



- These two approaches are different both technically/mathematically and philosophically

Statistical Dependency vs Causality

- Statistical dependency *does not imply* causality (~understanding)

  Examples: male violence

  married people live longer
- Causality is not necessary for prediction
- Dangerous to infer causality *from data alone* (as common in social studies, politics etc.)
- Causality can be demonstrated by arguments outside the data, or by carefully designed experimental setting

Classical Approaches to Inductive Inference

Generic problem: *finite data → Model*

Classical Statistics M1 ~ hypothesis testing

experimental data is generated by a given model (*single function* ~ scientific theory)

Classical Statistics M2 ~ max likelihood

~ data generated by a parametric model for density.

Note: loss fct ~ likelihood *(not application-specific)*

→ The same methodology for all learning problems

R. Fisher: *"uncertain inferences" from finite data*

see: R. Fisher (1935), The Logic of Inductive Inference, *J. Royal Statistical Society*, available at http://www.dcscience.net/fisher-1935.pdf

Summary and Discussion

- Math formulation useful for quantifying
  - explanation ~ fitting error (training data)
  - generalization ~ prediction error
- Natural assumptions
  - future similar to past: *stationary* $P(\mathbf{x},y)$, i.i.d.data
  - discrepancy measure or loss function, i.e. MSE

OUTLINE

2.0 Objectives

2.1 Terminology and Learning Problems

- supervised/ unsupervised
- classification
- regression etc.

2.2 Basic Learning Methods and Complexity Control

2.3 Inductive Principles

2.4 Alternative Learning Formulations

2.5 Summary

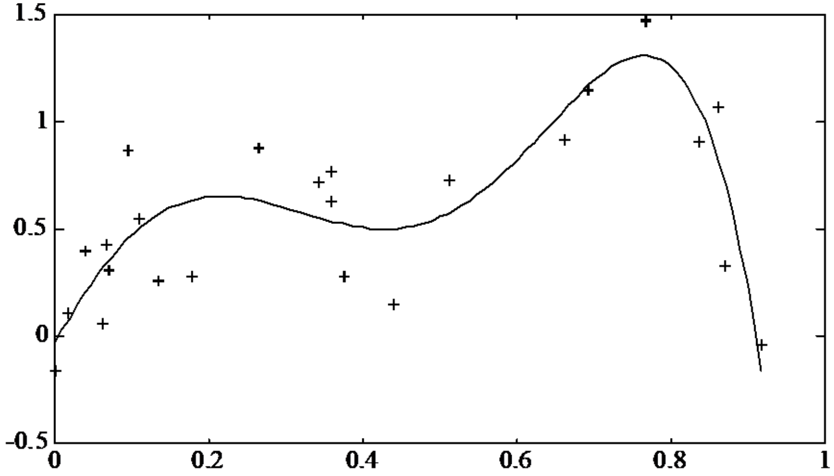
Supervised Learning: Regression

- Data in the form ($\mathbf{x}$,y), where

  - **x** is multivariate input (i.e. vector)

  - y is univariate output ('response')

- Regression: y is real-valued $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$



→ Estimation of real-valued function

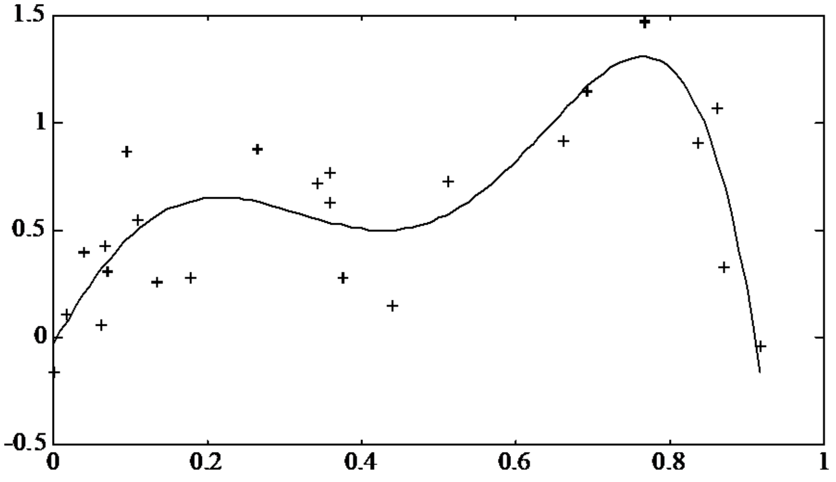
Regression Estimation Problem

Given: training data $(\mathbf{x}_i, y_i), i = 1,2,...n$

Find a function $f(\mathbf{x}, w^*)$ that minimizes squared error for a large number (*N*) of future samples:

$$\sum_{k=1}^{N} [(y_k - f(\mathbf{x}_k, w)]^2 \to min$$

$$\int (y - f(\mathbf{x},w))^2 \, dP(\mathbf{x},y) \to min$$



BUT future data is unknown ~ *P(**x**,y)* unknown

→ All estimation problems are ill-posed

Supervised Learning: Classification

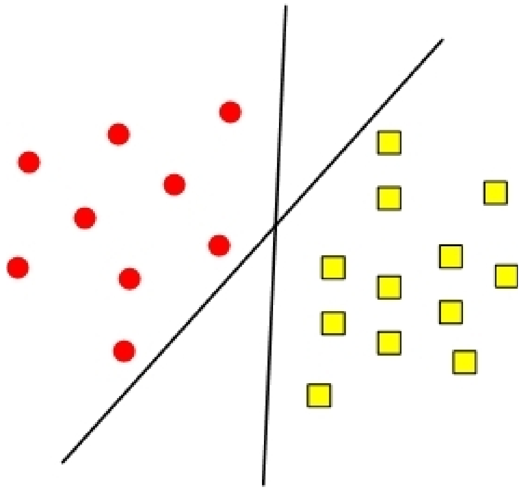
- Data in the form ($\mathbf{x}$,y), where

  - $\mathbf{x}$ is multivariate input (i.e. vector)

  - y is univariate output ('response')

- Classification: y is categorical (class label)

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



→ Estimation of indicator function

Density Estimation

- Data in the form (**x**), where

  \- **x** is multivariate input (feature vector)
- Parametric form of density is given: $\boldsymbol{f}(\mathbf{x}, \omega)$
- The loss function is likelihood or, more common, the negative log-likelihood

$$L(\boldsymbol{f}(\mathbf{x}, \omega)) = -\boldsymbol{ln}\, \boldsymbol{f}(\mathbf{x}, \omega)$$

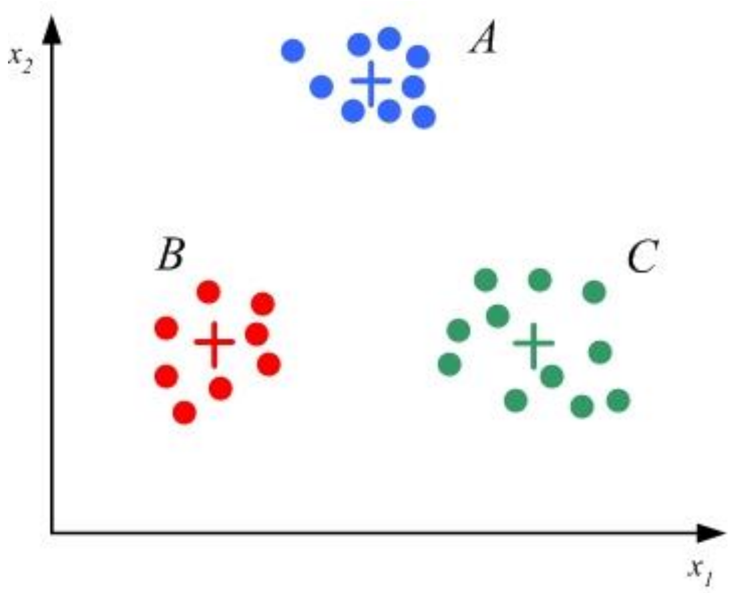
- The goal of learning is minimization of

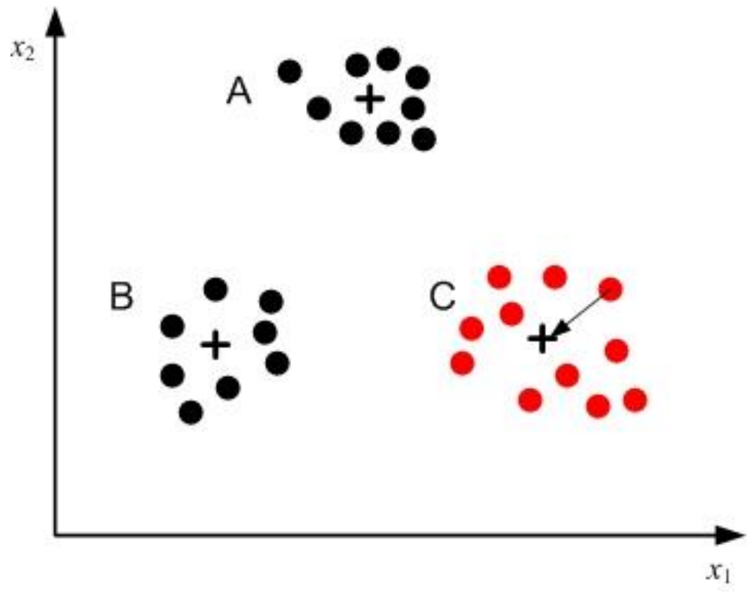
$$R(\omega) = \int -\boldsymbol{ln} \boldsymbol{f}(\mathbf{x}, \omega) \boldsymbol{p}(\mathbf{x}) d\mathbf{x}$$

from finite training data, yielding $\boldsymbol{f}(\mathbf{x}, \omega_0)$

Unsupervised Learning 1

- Data in the form (**x**), where
  - **x** is multivariate input (i.e. feature vector)
- Goal: data reduction or clustering



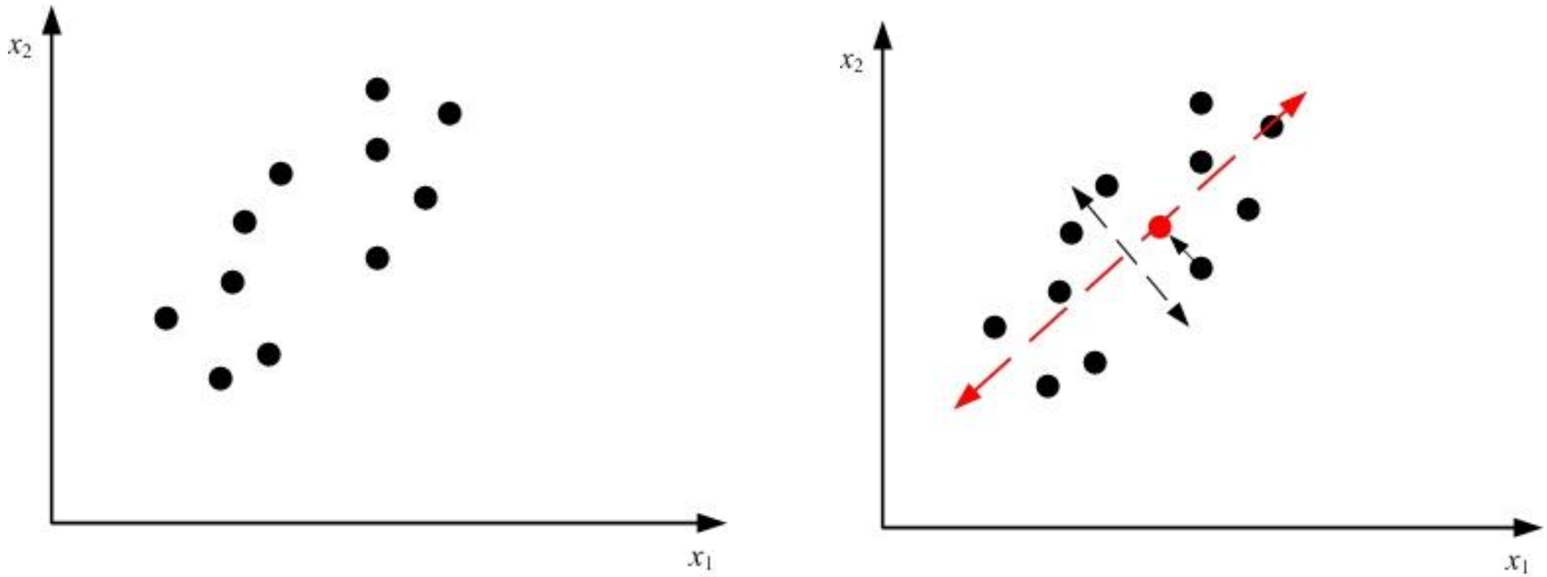


→ Clustering = estimation of mapping **X**→ **C**,
where $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$ and $L(\mathbf{x}, f(\mathbf{x})) = \|\mathbf{x} - f(\mathbf{x})\|^2$

Unsupervised Learning 2

- Data in the form (**x**), where

  \- **x** is multivariate input (i.e. vector)
- Goal: dimensionality reduction



→ Mapping $f(\mathbf{x})$ is projection of the data onto low-dimensional subspace, maximizing the variance (e.g., PCA).

OUTLINE

2.0 Objectives

2.1 Terminology and Learning Problems

2.2 Basic Learning Methods and Complexity Control

- Parametric modeling
- Non-parametric modeling
- Data reduction
- Complexity control

2.3 Inductive Principles

2.4 Alternative Learning Formulations

2.5 Summary

Basic learning methods

General idea

- Specify a **wide set** of possible models $f(\mathbf{x}, \omega)$ where ω is an abstract set of 'parameters'
- Estimate model parameters ω^* by minimizing *some loss function* for training data

Learning methods differ in

- Chosen parameterization
- Loss function used
- Optimization method used for parameter estimation

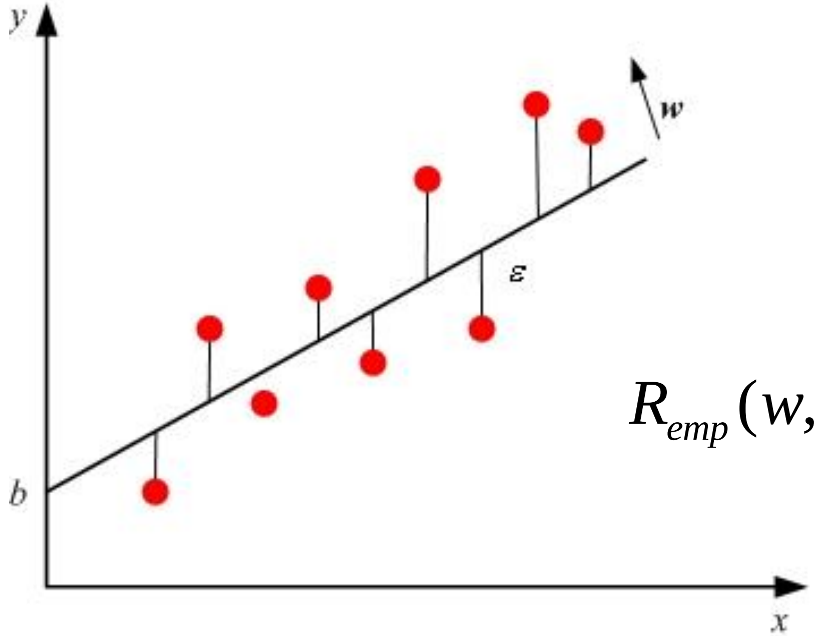
Parametric Modeling (~ERM)

Given training data $(\mathbf{x}_i, y_i), i = 1,2,\ldots n$

(1) Specify parametric model

(2) Estimate its parameters (via fitting to data)

- Example: Linear regression F(**x**)= (**w x**) + b



$$R_{emp}(w,b) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - (w \cdot x_i) - b\right]^2 \rightarrow min$$

Parametric Modeling: classification

Given training data $(\mathbf{x}_i, y_i), i = 1,2,\ldots n$

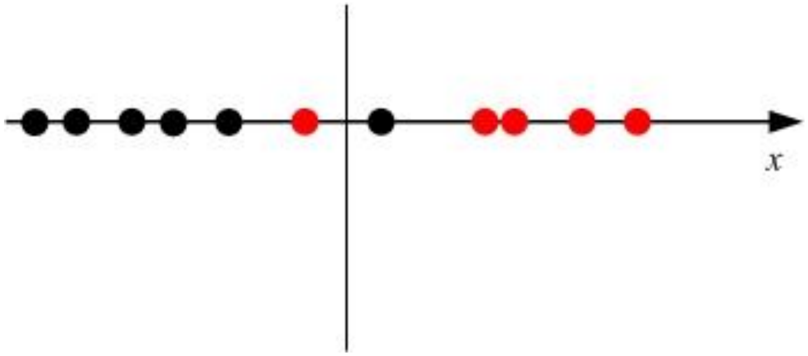
(1) Specify parametric model

(2) Estimate its parameters (via fitting to data)

Example: univariate classification data set

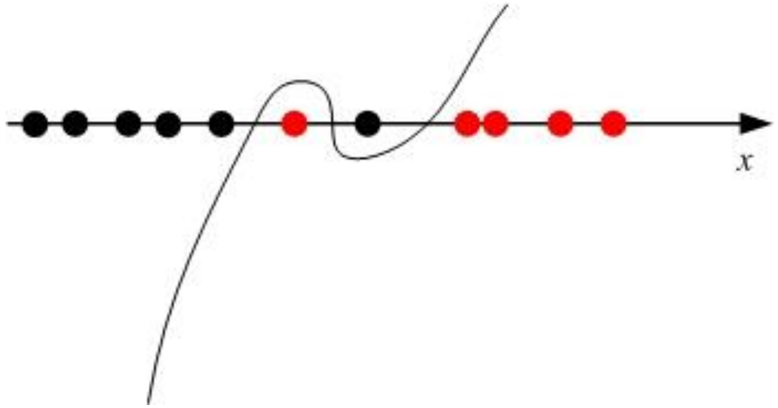
(a) Linear decision boundary

$$f(x) = sign(x - b)$$



(b) third-order polynomial

$$f(x) = sign(x^2 + wx + b)$$



Method of Maximum Likelihood

- The idea behind the method of maximum likelihood is to **estimate a parameter** with the value that makes the observed data **most likely**.
- When a **probability mass function** or **probability density function** is considered to be a function of the parameters, it is called a **likelihood function**.
- The maximum likelihood estimate is the value of the estimators that when substituted in for the parameters **maximizes the likelihood function**.

Navidi, William Cyrus. *Statistics for engineers and scientists*. Vol. 2. New York: McGraw-Hill, 2006.

Parametric modeling in Statistics

- Goal of learning : density estimation
- **Maximum Likelihood** principle:

 choose **w*** maximizing

$$P\left[\text{data}\middle|\text{model}\right] = P\left(\mathbf{X}\middle|\mathbf{w}\right) = \prod_{i=1}^{n} p\left(\mathbf{x}_i\,;\mathbf{w}\right)$$

equivalently, minimize negative log-likelihood

Maximum Likelihood illustration

let $X \sim Bin(20, p)$, where p is unknown. Suppose we observe the value $X = 7$. The pmf is

$$f(7; p) = \frac{20!}{7!\,13!} p^7 (1-p)^{13} \Rightarrow \underbrace{\text{a function of } p}_{\text{likelihood function}}$$

The MLE is the value $\hat{p}$ which, when substituted for p, maximize the likelihood function $f(7; p)$.

In principle, we can maximize the function by $\frac{d f(7;p)}{dp} = 0$.

However, it is easier to maximize $\ln f(7; p)$ instead!

Maximum Likelihood illustration (conti.)

($\because \ln x$ is an increasing function, therefore, the p maximizing $\ln f(7;p)$ also maximize $f(7;p)$)

(when we have n data points $x_1, x_2, \cdots x_n$, from $f(x;\theta)$ with unknown θ.
MLE of θ can be obtained by solving the following problems:

$$\max_{\theta} \prod_{i=1}^{n} f(x_i;\theta) \longrightarrow \text{(difficult)}$$

$$\max_{\theta} \sum_{i=1}^{n} \log(f(x_i;\theta)) \longrightarrow \text{(easy)}$$)

$$\ln f(7;p) = \ln 20! - \ln 7! - \ln 13! + 7\ln p + 13\ln(1-p)$$

$$\frac{d}{dp}\ln f(7;p) = \frac{7}{p} - \frac{13}{1-p} = 0 \Rightarrow \frac{7-7p-13p}{p(1-p)} = 0 \Rightarrow \frac{7-20p}{p(1-p)} = 0 \Rightarrow \hat{p} = \frac{7}{20}$$

Non-Parametric Modeling

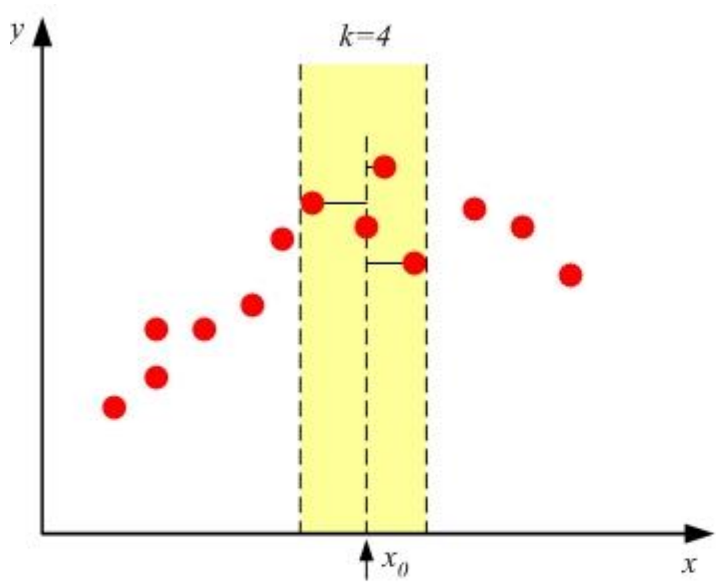
Given training data $(\mathbf{x}_i, y_i), i = 1,2,\ldots n$

Estimate the model (for given $\mathbf{x}_0$) as

'local average' of the data.

Note: need to define 'local', 'average'

- Example: k-nearest neighbors regression



$$f(\mathbf{x}_0) = \frac{\sum_{j=1}^{k} y_j}{k}$$

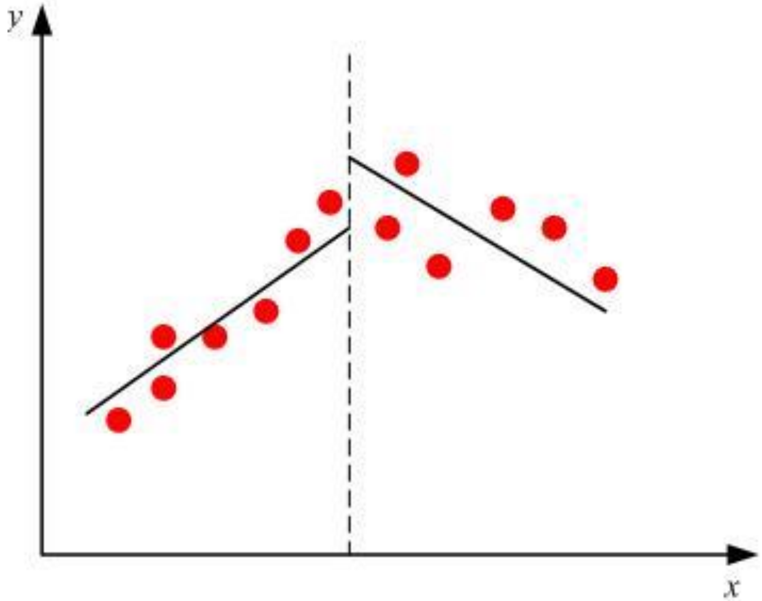
Data Reduction Approach

Given training data, estimate the model as 'compact encoding' of the data.

Note: 'compact' ~ # of bits to encode the model

or # of bits to encode the data (MDL)

- Example: piece-wise linear regression



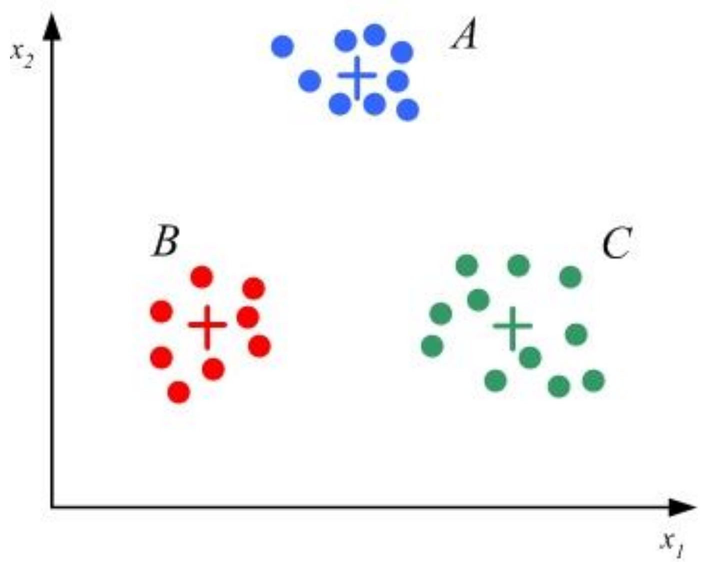
How many parameters needed for two-linear-component model?

Data Reduction Approach (cont'd)

Data Reduction approaches are commonly used for unsupervised learning tasks.

- Example: clustering.

Training data encoded by 3 points (cluster centers)



Issues:

- How to find centers?
- How to select the number of clusters?

Diverse terminology (of learning methods)

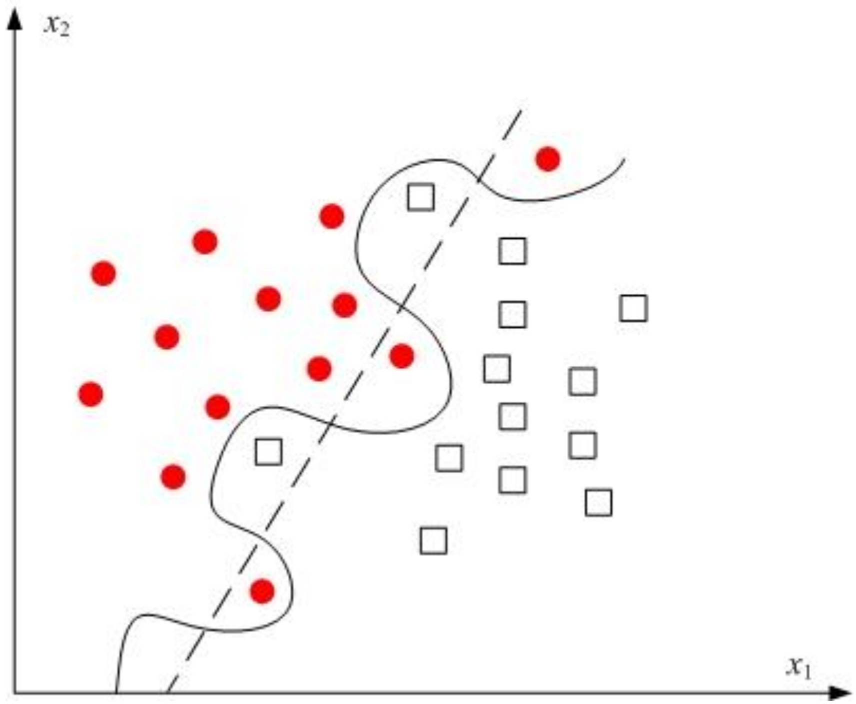
- Many methods differ in parameterization of admissible models or approximating functions $\hat{y} = f(\mathbf{x}, w)$
  - neural networks
  - decision trees
  - signal processing (~ wavelets)

- How training samples are used:
  Batch methods
  On-line or flow-through methods

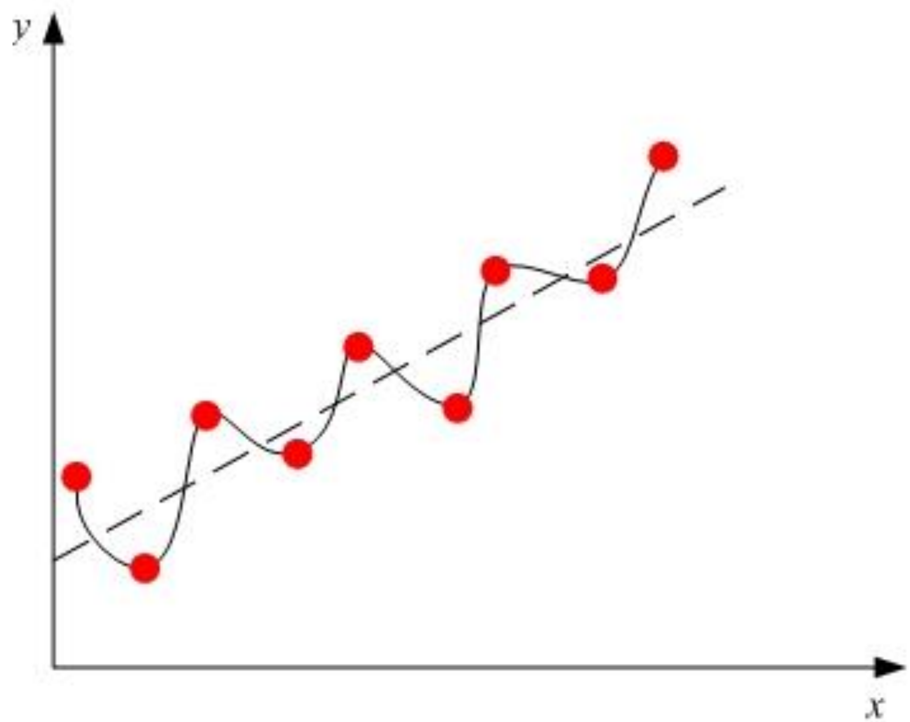
Explanation vs Prediction

→ Importance of complexity control

(a) Classification



(b) Regression



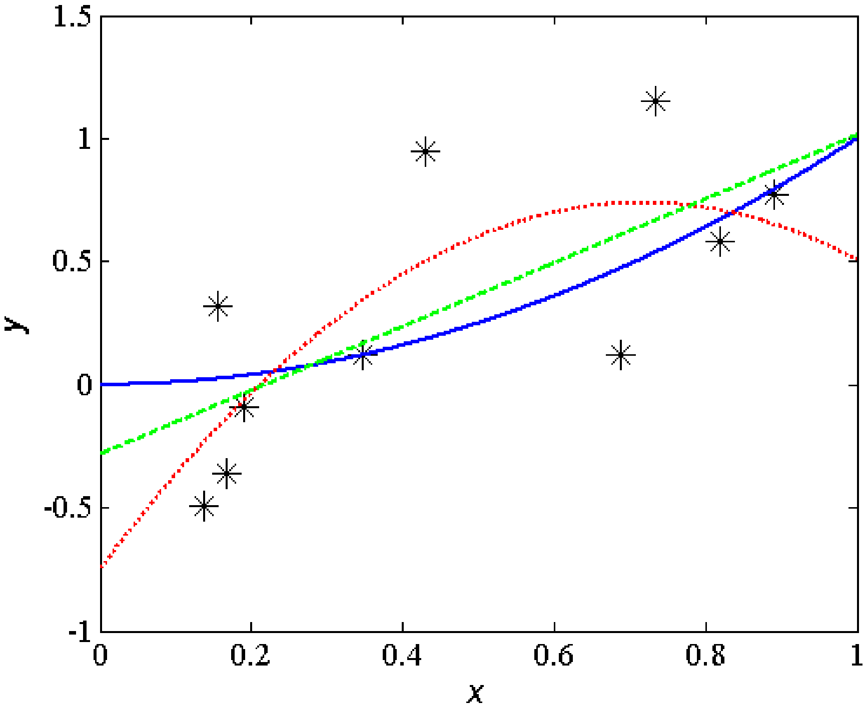
Complexity Control: parametric modeling

Consider regression estimation

- Ten training samples

$$y = x^2 + N(0, \sigma^2), where \sigma^2 = 0.25$$

- Fitting linear and 2-nd order polynomial:



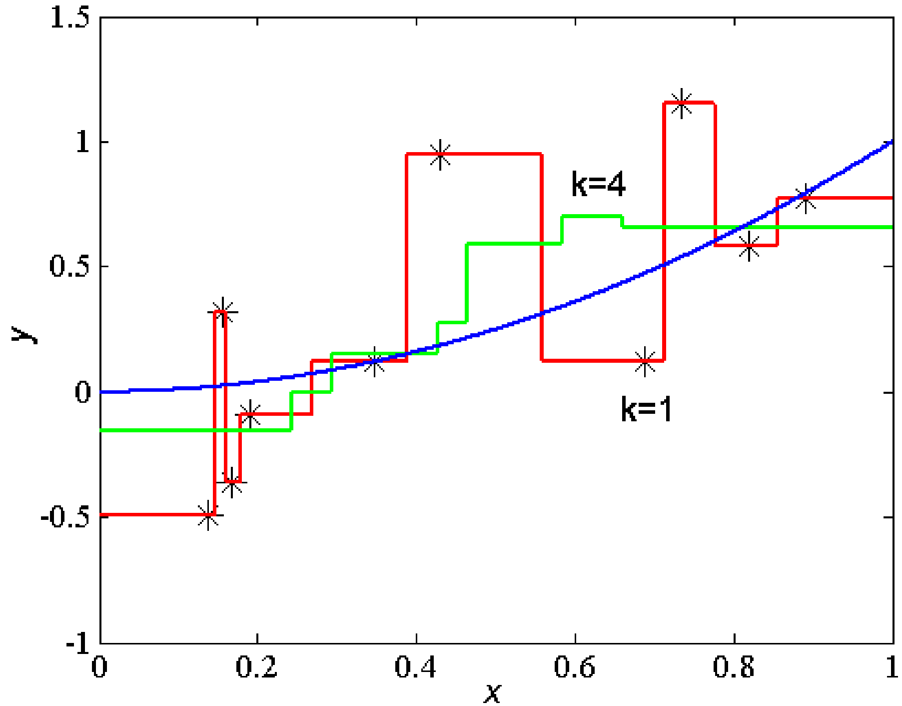
Complexity Control: local estimation

Consider regression estimation

- Ten training samples from

$$y = x^2 + N(0,\sigma^2), where \sigma^2 = 0.25$$

- Using k-nn regression with k=1 and k=4:



Complexity Control (summary)

- Complexity (of admissible models) affects generalization (for future data)
- Specific complexity indices for
  - Parametric models: ~ # of parameters
  - Local modeling: *size of local region*
  - Data reduction: # of clusters
- Complexity control = choosing optimal complexity (~ good generalization) for given (training) data set
- not well-understood in classical statistics

OUTLINE

2.0 Objectives

2.1 Terminology and Learning Problems

2.2 Basic Learning Methods and Complexity Control

2.3 Inductive Principles

- Motivation
- Inductive Principles: Penalization, SRM, Bayesian Inference

2.4 Alternative Learning Formulations

2.5 Summary

Motivation (cont'd)

- Generalization from finite data requires:

  a priori knowledge = *any info* outside the data, e.g. ???

  inductive principle = how to combine a priori knowledge with training data

  learning method = constructive implementation of inductive principle

- **Example: Empirical Risk Minimization** ~ parametric modeling approach

Motivation (cont'd)

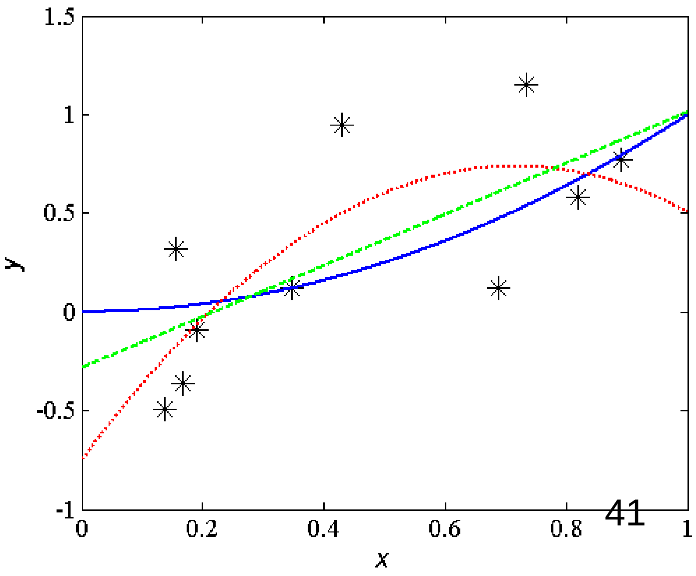
- **Example: Empirical Risk Minimization** ~ parametric modeling approach
- Prior knowledge:

$$y = x^2 + N(0,\sigma^2), where \sigma^2 = 0.25$$

- Inductive principle:

  e.g., the order of poly.

- Given the prior, we might not choose high order poly. functions.



Motivation (cont'd)

- Need for flexible (adaptive) methods $f(\mathbf{x}, w)$

  - **wide (~ flexible) parameterization**

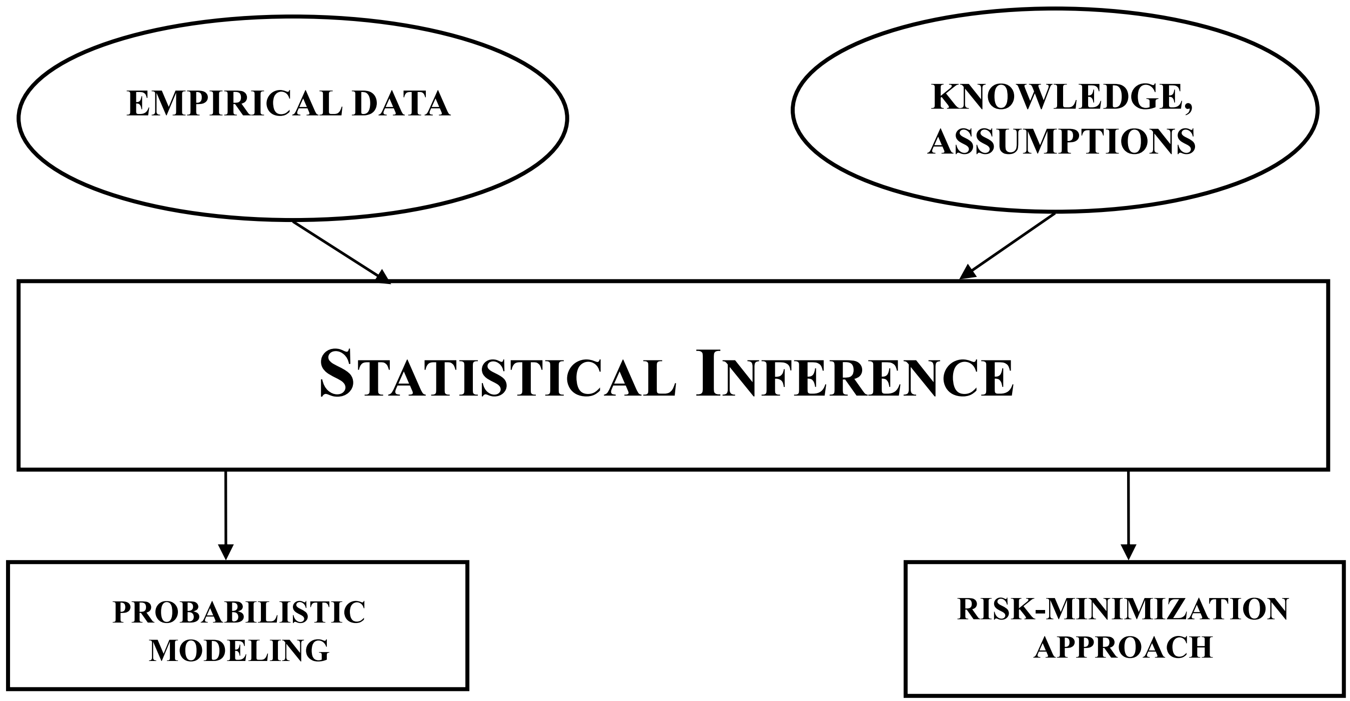
  → ill-posed estimation problems

  - need provisions for **complexity control**

- **Inductive Principles** originate from statistics, applied math, info theory, learning theory – and they adopt distinctly different terminology & concepts

Empirical Inference

- Two approaches to empirical or statistical inference



- General strategies for obtaining good models from data ~ known as **inductive principles** in learning theory

Inductive Principles

The main issue here is choosing the candidate model of the right complexity to describe the training data.

- **Inductive Principles differ in terms of**

  - representation (encoding) of a priori knowledge

  - **mechanism** for combining a priori knowledge with data

  - **applicability** when the true model does not belong to admissible models

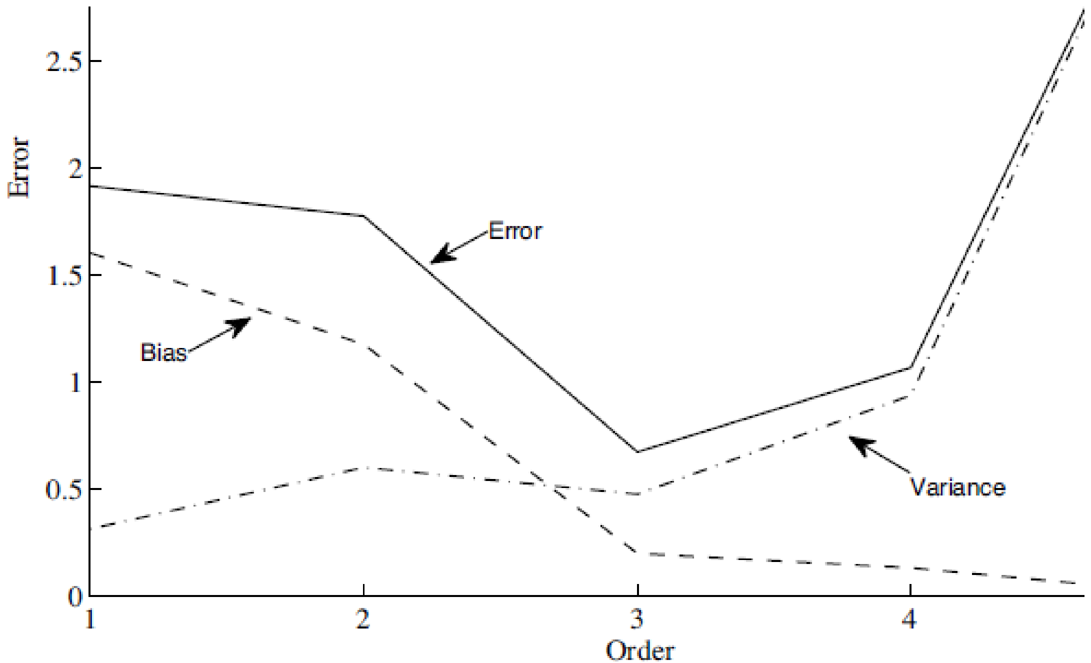
  - **availability** of constructive procedures (learning methods/ algorithms)

Model Selection Procedures

- Cross-validation (CV)
- Regularization or Penalization
- Structural Risk Minimization
- Bayesian model selection

Cross-validation

- Given a dataset, we divide it into two parts as training and validation sets.
- Train candidate models of different complexities, and test their error on the validation set left out during training.



Alpaydin, Ethem. *Introduction to machine learning*. MIT press, 2020.

Penalization

- Overcomes the limitations of ERM
- Penalized empirical risk functional

$$R_{pen}(\omega) = R_{emp}(\omega) + \lambda\phi[\mathrm{f}(\mathbf{x},\omega)]$$

$\phi[\mathrm{f}(\mathbf{x},\omega)]$ is non-negative penalty functional specified *a priori* (independent of the data); its larger values penalize complex functions.

λ is regularization parameter (non-negative number) tuned to training data

Example: LASSO

Least absolute shrinkage and selection operator (LASSO)

- A type of linear regression with penalization.
- The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters).
- Lower model complexity results in better generalization.
- The goal of the algorithm is to minimize:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

https://www.statisticshowto.com/lasso-regression/

Structural Risk Minimization

- Complexity ordering on a set of admissible models, as a nested structure

$$S_0 \subset S_1 \subset S_2 \subset \ldots S_k$$

Examples: a set of polynomial models, polynomials of degree *m* are a subset of polynomials of degree *(m+1)*.

The complexity is given by the number of free parameters.

Structural Risk Minimization (conti.)

- The optimal choice of model complexity provides the minimum of the expected risk.
- Statistical learning theory (Vapnik 1995) provides analytic upper-bound estimates for expected risk.

Bayesian Inference

- Probabilistic approach to inference
- Explicitly defines a priori knowledge as **prior probability** (distribution) on a set of model parameters
- Bayes formula for updating prior probability using the evidence given by the data:

$$P[\mathrm{model}|\mathrm{data}] = \frac{P[\mathrm{data}|\mathrm{model}]P[\mathrm{model}]}{P[\mathrm{data}]}$$

$P[\mathrm{model}|\mathrm{data}]$ ~ posterior probability

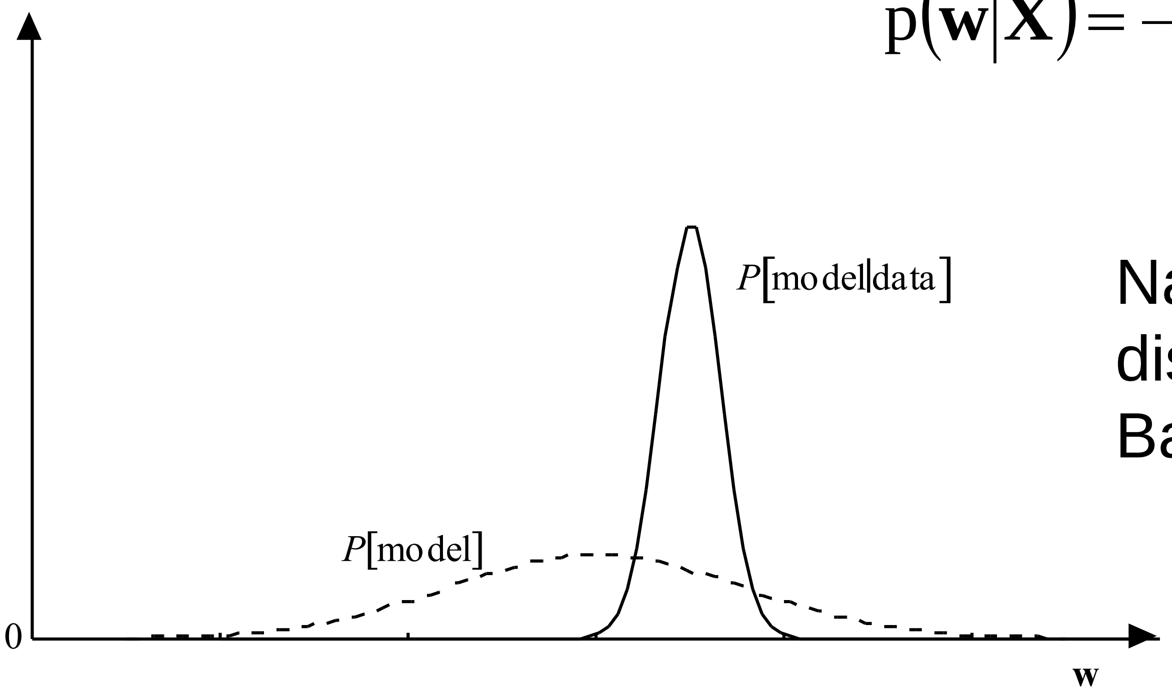
$P[\mathrm{data}|\mathrm{model}]$ ~ likelihood (probability that the data are generated by a model)

Bayesian Density Estimation

- Consider parametric density estimation where prior probability distribution $P[\text{model}] = \boldsymbol{p}(\mathbf{w})$

  Then posterior probability distribution is updated

$$\mathrm{p}(\mathbf{w}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{w})\mathrm{p}(\mathbf{w})}{P(\mathbf{X})}$$



Narrow posterior distribution using Bayes rule

Implementation of Bayesian Inference

EX: Classification problem

- The posterior probability of class C_i can be calculated as

$$P(C_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_i)P(C_i)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|C_i)P(C_i)}{\sum_{k=1}^{K} p(\boldsymbol{x}|C_k)P(C_k)}$$

- For minimum error, chooses the class with the highest posterior probability; that is, we

$$\text{choose } C_i \text{ if } P(C_i|\boldsymbol{x}) = \max_k P(C_k|\boldsymbol{x})$$

Alpaydin, Ethem. *Introduction to machine learning*. MIT press, 2020.

MAP (Maximum A Posteriori)

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

$$\propto P(X|\theta)P(\theta)$$

$$\begin{aligned}
\theta_{MAP} &= \arg\max_{\theta} P(X|\theta)P(\theta) \\
&= \arg\max_{\theta} \log P(X|\theta) + \log P(\theta) \\
&= \arg\max_{\theta} \log \prod_{i} P(x_i|\theta) + \log P(\theta) \\
&= \arg\max_{\theta} \sum_{i} \log P(x_i|\theta) + \log P(\theta)
\end{aligned}$$

http://www.sefidian.com/2022/06/21/difference-between-maximum-likelihood-estimation-mle-and-maximum-a-posteriori-map/

MLE (Maximum Likelihood Estimation)

$$\theta_{MLE} = \arg\max_{\theta} \log P(X|\theta)$$

$$= \arg\max_{\theta} \log \prod_{i} P(x_i|\theta)$$

$$= \arg\max_{\theta} \sum_{i} \log P(x_i|\theta)$$

http://www.sefidian.com/2022/06/21/difference-between-maximum-likelihood-estimation-mle-and-maximum-a-posteriori-map/

MLE vs MAP

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

Maximum Likelihood Estimate (MLE)

$$\boldsymbol{\theta}^{\text{MLE}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

Maximum *a posteriori* (MAP) estimate

$$\boldsymbol{\theta}^{\text{MAP}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})\underbrace{p(\boldsymbol{\theta})}_{\text{Prior}}$$

MLE is a special case of MAP when the prior is uniform!

http://www.sefidian.com/2022/06/21/difference-between-maximum-likelihood-estimation-mle-and-maximum-a-posteriori-map/

Comparison of Inductive Principles

- Representation of a priori knowledge/ complexity:
  penalty term, structure, prior distribution
- Formal procedure for complexity control:
  penalized risk, optimal element of a structure, posterior distribution
- Constructive implementation of complexity control:
  resampling, analytic bounds, marginalization

See Table 2.1 in [Cherkassky & Mulier, 2007]

Comparison of Inductive Principles

TABLE 2.1 Features of Inductive Principles

	Penalization	SRM	Bayes	MDL
Representation of a priori knowledge or complexity	Penalty term	Structure	Prior distribution	Codebook
Constructive procedure for complexity control	Minimum of penalized risk	Optimal element of a structure	A posteriori distribution	Not defined
Method for model selection	Resampling	Analytic bound on prediction risk	Marginalization	Minimum code length
Applicability when the true model does not belong to the set of approximating functions	Yes	Yes	No	Yes

For MDL, ***See pages 51-55 in [Cherkassky & Mulier, 2007]***

OUTLINE

2.0 Objectives

2.1 Terminology and Learning Problems

2.2 Basic Learning Methods and Complexity Control

2.3 Inductive Principles

2.4 Alternative Learning Formulations

- Vapnik's principle
- Examples of non-standard formulations
- Formalization of application domain

2.5 Summary

Keep It Direct Principle

- **Vapnik's principle**

  For estimation with finite data, do not solve a given problem by *indirectly solving a more general/harder problem* as an intermediate step

  Note 1: this is contrary to classical science

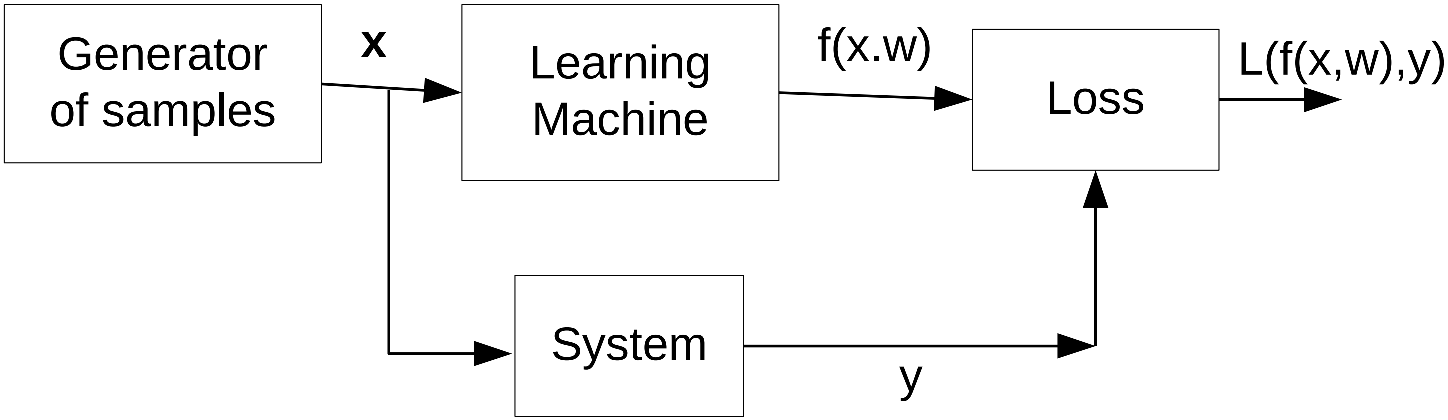
  Note 2: contradicts classical statistical approach

- **Examples**

  - Classification via density estimation etc.

  - ***Non-standard*** inductive learning settings

Assumptions for Inductive Learning



- Available (training) data **format** (**x**,y)
- Test samples (**x**-values) are **unknown**
- **Stationary distribution, i.i.d samples**
- **Single model** needs to be estimated
- **Specific loss functions** adopted for common tasks (classification, regression etc.)

Non-standard Learning Settings

- **Available Data Format**

  - x-values of test samples are known

  → Transduction, semi-supervised learning
- **Different (non-standard) Loss Function**

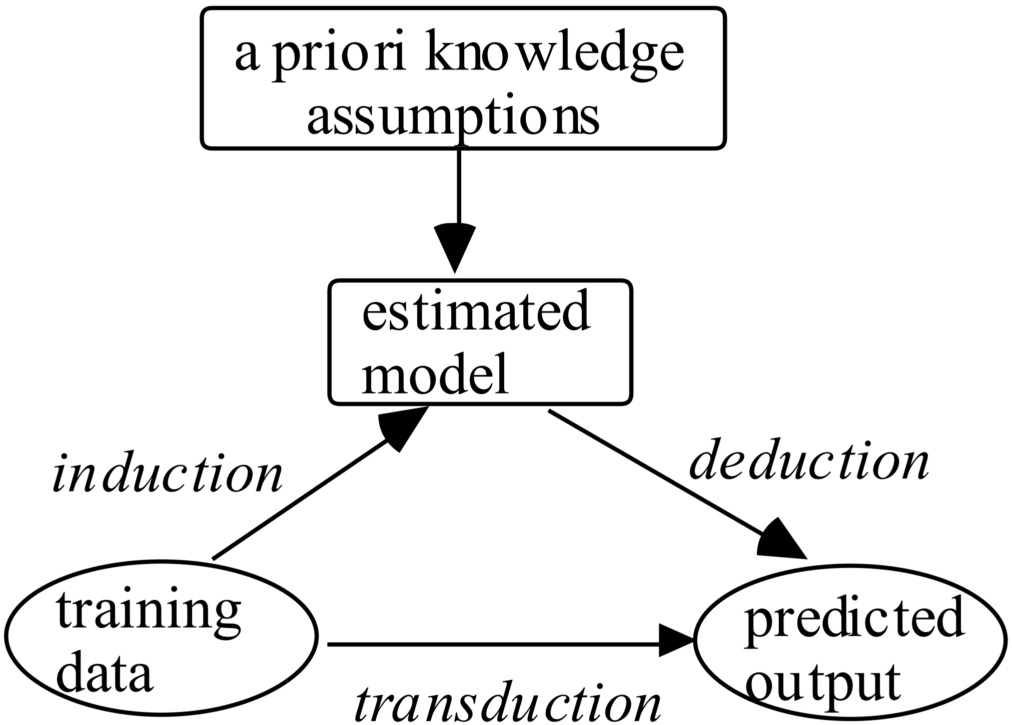
  - see example ‘learning the sign of a function’
- **Univariate Output (~ a single model)**

  - multiple models may be estimated from available/training data

Transduction

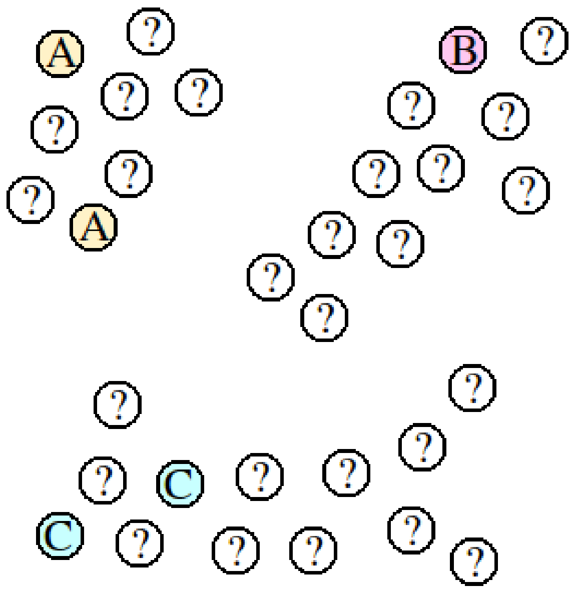
~ predicting function values at given points:

- **Given** labeled training set + x-values of test data
- **Estimate (predict)** y-values for given test inputs



Transduction

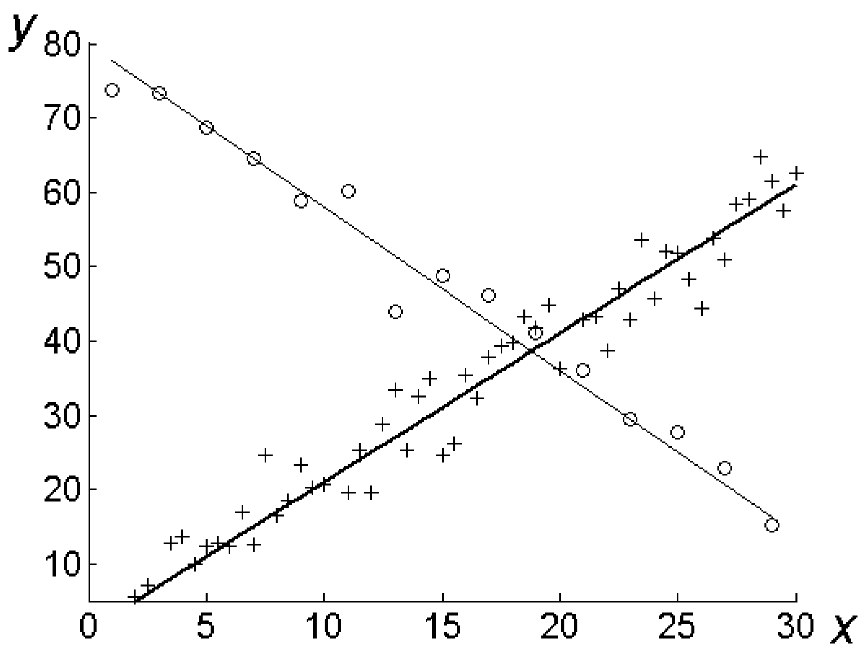
- The supervised learning algorithm will only have five labeled points for modeling.
- Transduction has the advantage of being able to consider all of the points.
- EX: Label the unlabeled points according to the local estimation.



https://en.wikipedia.org/wiki/Transduction_(machine_learning)

Multiple Model Estimation

- Training data in the form (**x**,y), where
  - **x** is multivariate input
  - y is univariate real-valued output ('response')
- Similar to standard regression, but subsets of data may be described by different models



Formalization of Application Problems

- *Problem Specification Step* (in the general experimental procedure) cannot be formalized

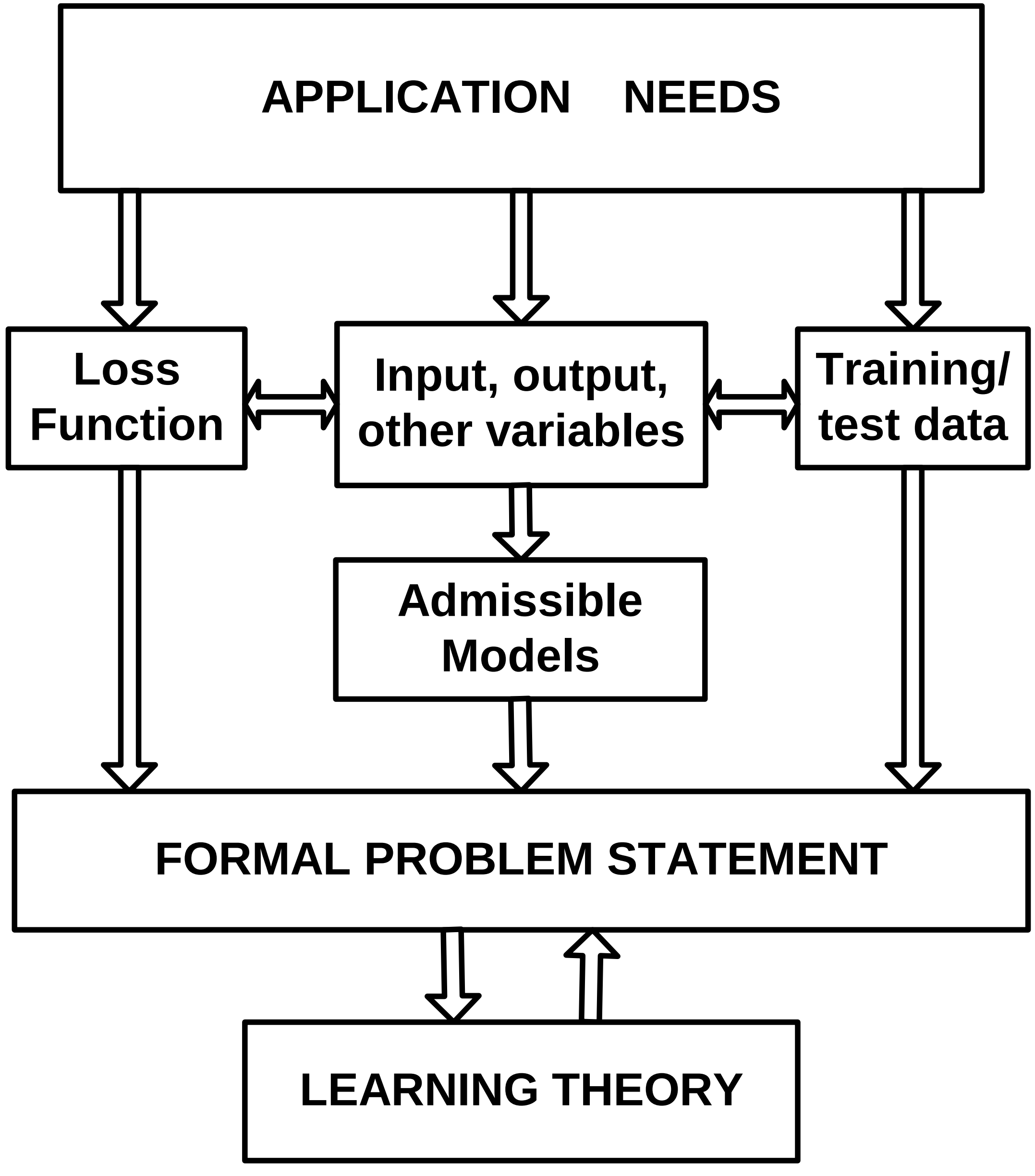
But

- **Several guidelines** can be helpful during formalization process

- Mapping process:

Application requirements → Learning formulation

- Specific components of this mapping process are shown next


APPLICATION NEEDS
Loss Function
Input, output, other variables
Training/ test data
Admissible Models
FORMAL PROBLEM STATEMENT
LEARNING THEORY

Summary

- Standard Inductive Learning ~ function estimation
- Goal of learning (empirical inference):
  to act/perform well, not system identification
- Important concepts:
  - training data, test data
  - loss function, prediction error (~ prediction risk)
  - basic learning problems
- Complexity control
- Inductive principles – which is the 'best' ?

Summary (cont'd)

- Assumptions for inductive learning
- Non-standard learning formulations

Aside: predictive modeling of
physical systems vs social systems

- For discussion think of example application that requires non-standard learning formulation

Note: (a) *do not* use examples similar to ones presented in my lectures and/or text book
(b) you can email your example(s) to instructor (maximum half-a-page)