

Machine Learning

Computer Engineering

Fabio Vandin

October 6th, 2023

A Formal Model (Statistical Learning)

We have a *learner* (us, or the machine) has access to:

- ① **Domain set** \mathcal{X} : set of all possible objects to make predictions about
 - domain point $x \in \mathcal{X} = \text{instance}$, usually represented by a vector of *features*
 - \mathcal{X} is the *instance space*
- ② **Label set** \mathcal{Y} : set of possible labels.
 - often two labels, e.g. $\{-1, +1\}$ or $\{0, 1\}$
- ③ **Training data** $S = ((x_1, y_1), \dots, (x_m, y_m))$: finite sequence of labeled domain points, i.e. pairs in $\mathcal{X} \times \mathcal{Y}$
 - this is the learner's **input**
 - S : *training example* or *training set*

A Formal Model

- ④ **Learner's output** h : prediction rule $h: \mathcal{X} \rightarrow \mathcal{Y}$
 - also called *predictor*, *hypothesis*, or *classifier*
 - $A(S)$: prediction rule produced by learning algorithm A when training set S is given to it
 - sometimes \hat{f} used instead of h
- ⑤ **Data-generation model**: instances are generated by some probability distribution and labeled according to a function
 - \mathcal{D} : probability distribution over \mathcal{X} (**NOT KNOWN TO THE LEARNER!**)
 - labeling function $f: \mathcal{X} \rightarrow \mathcal{Y}$ (**NOT KNOWN TO THE LEARNER!**)
 - label y_i of instance x_i : $y_i = f(x_i)$, for all $i = 1, \dots, m$
 - each point in training set S : first sample x_i according to \mathcal{D} , then label it as $y_i = f(x_i)$
- ⑥ **Measures of success**: *error of a classifier* = probability it does not predict the correct label on a random data point generate by distribution \mathcal{D}

Loss

Given domain subset $A \subset \mathcal{X}$, $\mathcal{D}(A)$ = probability of observing a point $x \in A$.

In many cases, we refer to A as *event* and express it using a function $\pi : \mathcal{X} \rightarrow \{0, 1\}$, that is:

$$A = \{x \in \mathcal{X} : \pi(x) = 1\}$$

In this case we have $\mathbb{P}_{x \sim \mathcal{D}}[\pi(x)] = \mathcal{D}(A)$

Error of prediction rule $h : \mathcal{X} \rightarrow \mathcal{Y}$ is

$$L_{\mathcal{D}, f}(h) \stackrel{\text{def}}{=} \mathbb{P}_{x \sim \mathcal{D}}[h(x) \neq f(x)] \stackrel{\text{def}}{=} \mathcal{D}(\{x : h(x) \neq f(x)\})$$

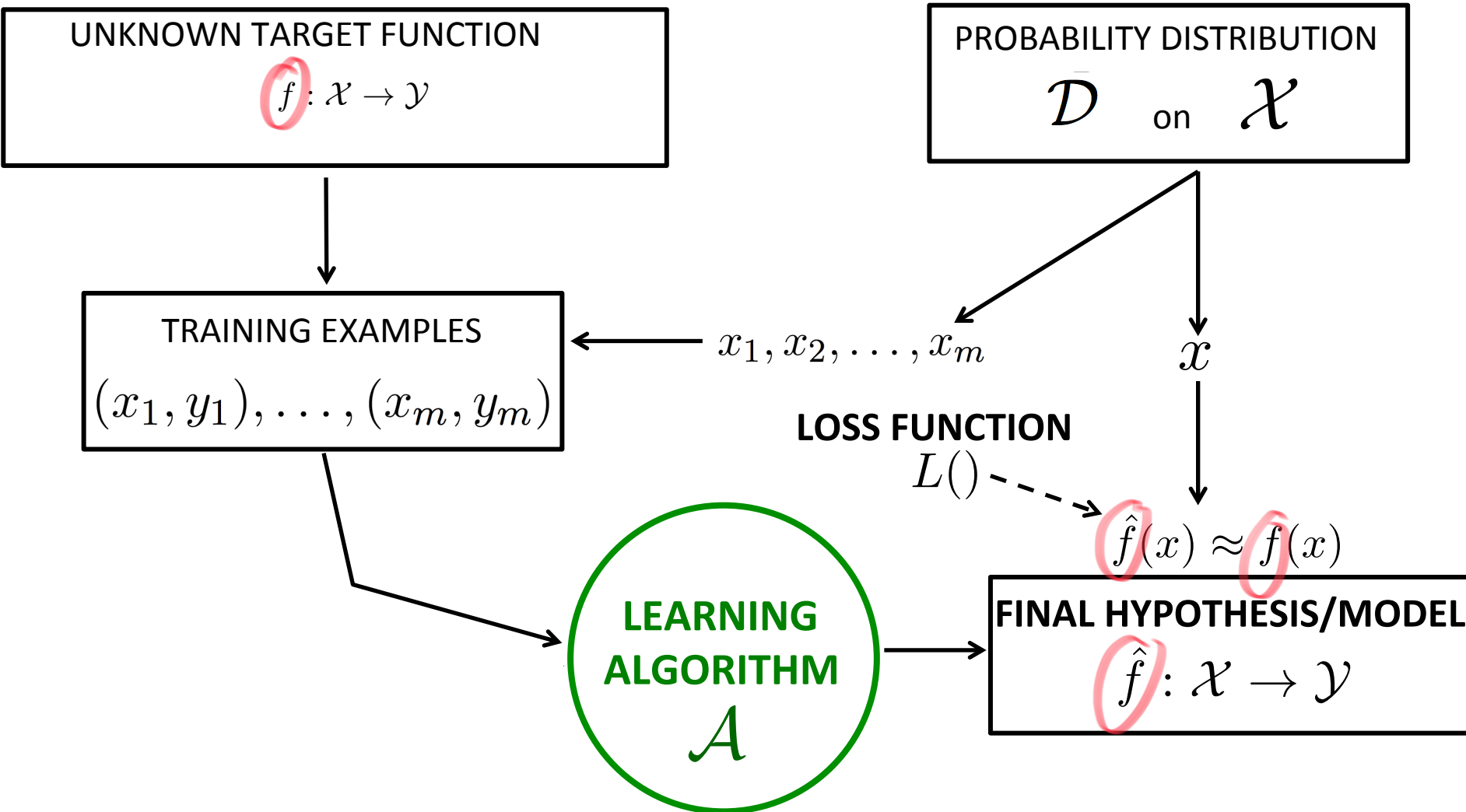
"true" label

→ label predicted by the model h

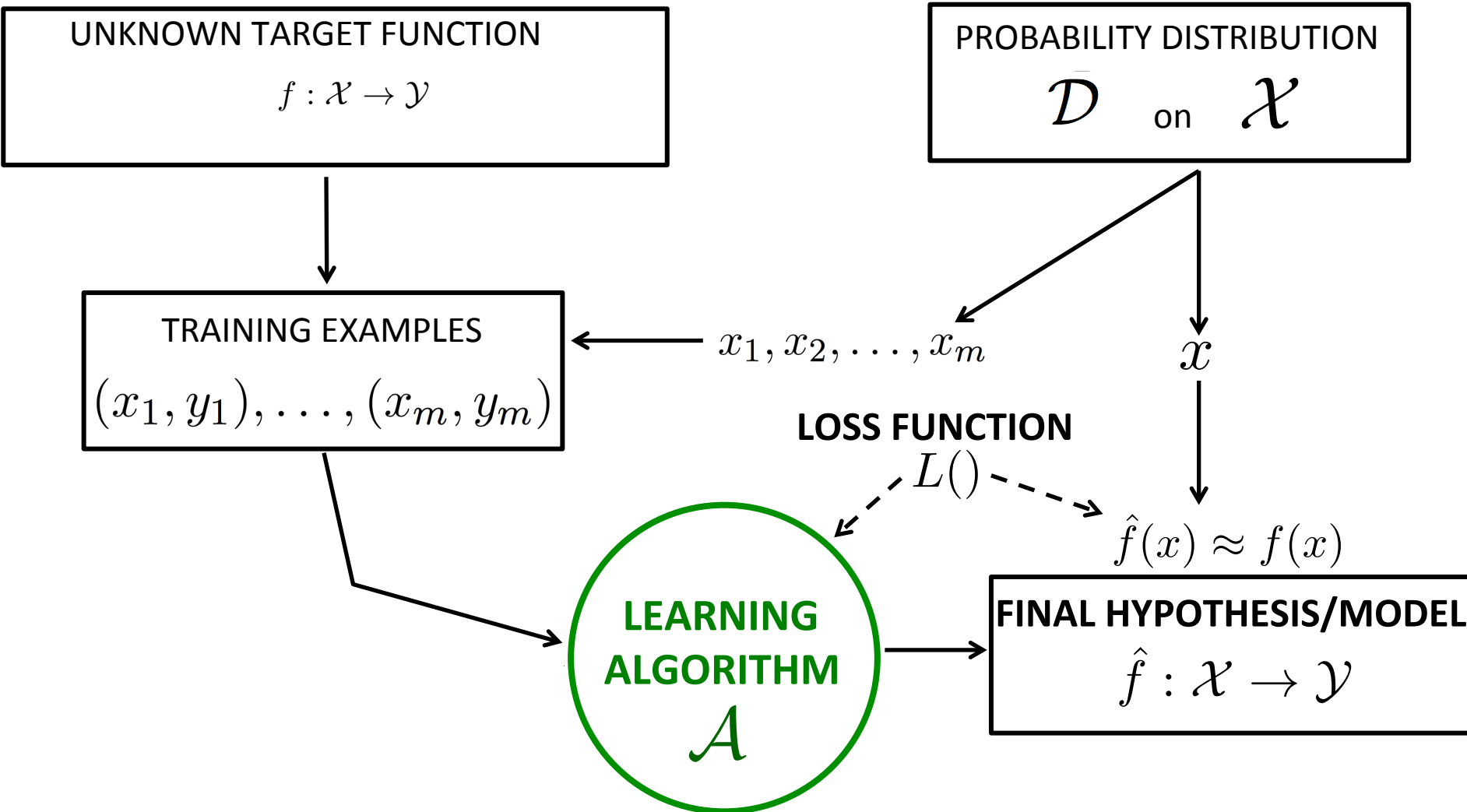
Notes:

- $L_{\mathcal{D}, f}(h)$ has many different names: **generalization error**, *true error*, *risk*, **loss**, ...
- often f is obvious, so omitted: $L_{\mathcal{D}}(h)$

Learning Process (Simplified)



Learning Process (Simplified)



Learning Process (Simplified)

UNKNOWN TARGET FUNCTION

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

PROBABILITY DISTRIBUTION

$$\bar{\mathcal{D}} \text{ on } \mathcal{X}$$

TRAINING EXAMPLES

$$(x_1, y_1), \dots, (x_m, y_m)$$

$$x_1, x_2, \dots, x_m$$

LOSS FUNCTION

$$L()$$

$$\hat{f}(x) \approx f(x)$$

LEARNING
ALGORITHM
 \mathcal{A}

FINAL HYPOTHESIS/MODEL

$$\hat{f} : \mathcal{X} \rightarrow \mathcal{Y}$$

HYPOTHESIS/MODEL SET

$$\mathcal{H}$$

Types of Learning

y_i are known: **training set** $(x_1, y_1), \dots, (x_m, y_m)$

➡ **supervised learning**

Training set contains only x_1, x_2, \dots, x_m

➡ **unsupervised learning**

There can be different types of output:

- \mathcal{Y} is **discrete**
- \mathcal{Y} is **continuous**

Notes: we will see a more general learning model soon, main ideas are the same!

Types of Learning

y_i known

y_i not available

Supervised Learning

Unsupervised Learning

\mathcal{Y} is ...
Discrete
Continuous

classification

clustering

dimensionality
reduction

regression

...

(Rough) Course Plan

PART I: Supervised Learning

Introduction

Probability Review

Learning Model: PAC Learning

Model Complexity and VC Dimension

Linear Models for Regression: least squares

Linear Models for Classification: Perceptron

Model Selection and Validation

Regularization and Feature Selection

(Rough) Course Plan

Support Vector Machines (SVM) for Classification and Regression

SVM and Kernels

Neural Networks for Classification and Regression

Deep Learning

Decision Trees and Random Forests

PART II: Unsupervised Learning

Hierarchical clustering

Cost based clustering: k-means

Objectives

Provide the **fundamentals** and **basic principles** of the **learning problem**

Introduce the **most common algorithms** for **regression** and **classification**

Analytical and **practical ability** in using these tools for the solution of basic problems

Some **hands-on** experience

Course Prerequisites!

Calculus

Programming

Linear Algebra

Probability

Calculus

- derivatives
- minimization of functions
- partial derivatives of functions of multiple variables
- integrals

Programming

- You should know at least one programming language (e.g., Java)
... learning Python will be easy!

Linear Algebra

- matrix factorization
- matrix inversion
- linear independence
- rank, column space, null space
- orthogonality, projections
- eigenvalues, eigenvectors
- symmetric positive definite matrices
- matrix differentiation

Probability

- discrete random variables (r.v.), moments, expectation
- joint, marginal, conditional distribution
- some famous distributions:
 - discrete: binomial
 - continuous: Gaussian
- Independence and conditional independence
- Bayes Theorem
- Law of large numbers

Useful but may not be required: continuous r.v.'s, probability density function (PDF), cumulative distribution function (CDF)

Useful link: *Seeing theory (visualization for probability, statistics, etc.)*

<https://students.brown.edu/seeing-theory/basic-probability/index.html>

See background material on “Useful links and other stuff”