

**MLCC 2018**  
**Statistical Learning: Basic Concepts**

Lorenzo Rosasco  
UNIGE-MIT-IIT

# Outline

Learning from Examples

Data Space and Distribution

Loss Function and Expected Risk

Stability, Overfitting and Regularization

# Learning from Examples

- ▶ Machine Learning deals with systems that are trained from data rather than being explicitly programmed
- ▶ Here we describe the framework considered in statistical learning theory.

# Supervised Learning

The goal of supervised learning is to find an underlying input-output relation

$$f(x_{\text{new}}) \sim y,$$

given data.

# Supervised Learning

The goal of supervised learning is to find an underlying input-output relation

$$f(x_{new}) \sim y,$$

given data.

The data, called *training set*, is a set of  $n$  input-output pairs (examples)

$$S = \{(x_1, y_1), \dots, (x_n, y_n)\}.$$

## We Need a Model to Learn

- ▶ We consider the approach to machine learning based on the *learning from examples* paradigm
- ▶ **Goal:** Given the training set, *learn* a corresponding I/O relation
- ▶ We have to postulate the existence of a model for the data
- ▶ The model should take into account the possible *uncertainty* in the task *and* in the data

# Outline

Learning from Examples

Data Space and Distribution

Loss Function and Expected Risk

Stability, Overfitting and Regularization

# Data Space

- ▶ The inputs belong to an input space  $X$ , we assume that  $X \subseteq \mathbb{R}^D$
- ▶ The outputs belong to an output space  $Y$ , typically a subset of  $\mathbb{R}$
- ▶ The space  $X \times Y$  is called the *data space*



## Examples of Data Space

We consider several possible situations:

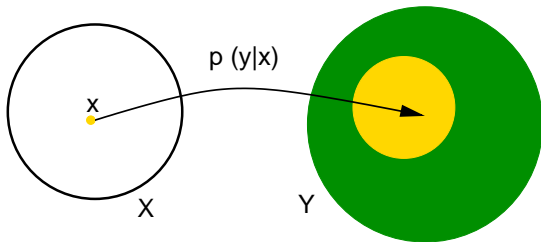
- ▶ Regression:  $Y \subseteq \mathbb{R}$
- ▶ Binary classification  $Y = \{-1, 1\}$
- ▶ Multi-category (multiclass) classification  $Y = \{1, 2, \dots, T\}$ .
- ▶ ...

## Modeling Uncertainty in the Data Space

- ▶ **Assumption:**  $\exists$  a fixed unknown distribution  $p(x, y)$  according to which the data are **identically and independently sampled**
- ▶ The distribution  $p$  models different sources of **uncertainty**
- ▶ **Assumption:**  $p$  factorizes as  $p(x, y) = p_X(x)p(y|x)$

## Marginal and Conditional

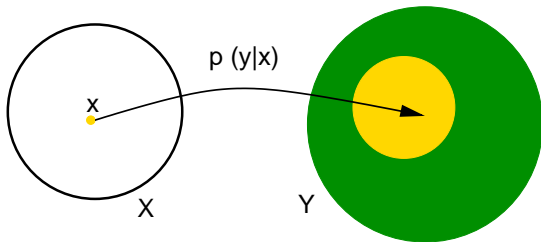
$p(y|x)$  can be seen as a form of *noise* in the output



**Figure:** For each input  $x$  there is a distribution of possible outputs  $p(y|x)$ .

## Marginal and Conditional

$p(y|x)$  can be seen as a form of *noise* in the output



**Figure:** For each input  $x$  there is a distribution of possible outputs  $p(y|x)$ .

The marginal distribution  $p_X(x)$  models uncertainty in the sampling of the input points.

## Data Models

- ▶ In **regression**, the following model is often considered:

$$y = f^*(x) + \epsilon$$

where:

- $f^*$ : fixed unknown (*regression*) function
  - $\epsilon$ : random noise, e.g. standard Gaussian  $\mathcal{N}(0, \sigma I)$ ,  $\sigma \in [0, \infty)$
- ▶ In **classification**,

$$p(1|x) = 1 - p(-1|x), \forall x$$

Noiseless classification,  $p(1|x) = \{1, 0\}, \forall x \in X$

# Outline

Learning from Examples

Data Space and Distribution

Loss Function and Expected Risk

Stability, Overfitting and Regularization

# Loss Function

**Goal of learning:** Estimate “best” I/O relation (not the whole  $p(x, y)$ )

- ▶ We need to fix a *loss function*

$$\ell : Y \times Y \rightarrow [0, \infty)$$

$\ell(y, f(x))$  is a point-wise error measure. It is the cost of when predicting  $f(x)$  in place of  $y$

## Expected Risk and Target Function

The *expected loss* (or *expected risk*)

$$\mathcal{E}(f) = \mathbb{E}[\ell(y, f(x))] = \int p(x, y)\ell(y, f(x))dx dy$$

can be seen as a measure of the error on past as well as future data.



## Expected Risk and Target Function

The *expected loss* (or *expected risk*)

$$\mathcal{E}(f) = \mathbb{E}[\ell(y, f(x))] = \int p(x, y)\ell(y, f(x))dxdy$$

can be seen as a measure of the error on past as well as future data.

Given  $\ell$  and a distribution, the "best" I/O relation is the *target function*

$$f^* : X \rightarrow Y$$

that minimizes the expected risk

## Learning from Data

- ▶ The target function  $f^*$  cannot be computed, since  $p$  is **unknown**

# Learning from Data

- ▶ The target function  $f^*$  cannot be computed, since  $p$  is **unknown**
- ▶ The goal of learning is to find an *estimator* of the target function from data

# Outline

Learning from Examples

Data Space and Distribution

Loss Function and Expected Risk

Stability, Overfitting and Regularization

## Learning Algorithms and Generalization

- ▶ A **learning algorithm** is a procedure that given a training set  $S$  computes an estimator  $f_S$

## Learning Algorithms and Generalization

- ▶ A **learning algorithm** is a procedure that given a training set  $S$  computes an estimator  $f_S$
- ▶ An estimator should mimic the target function, in which case we say that it **generalizes**

## Learning Algorithms and Generalization

- ▶ A **learning algorithm** is a procedure that given a training set  $S$  computes an estimator  $f_S$
- ▶ An estimator should mimic the target function, in which case we say that it **generalizes**
- ▶ More formally we are interested in an estimator such that the **excess expected risk**

$$\mathcal{E}(f_S) - \mathcal{E}(f^*),$$

is small

## Learning Algorithms and Generalization

- ▶ A **learning algorithm** is a procedure that given a training set  $S$  computes an estimator  $f_S$
- ▶ An estimator should mimic the target function, in which case we say that it **generalizes**
- ▶ More formally we are interested in an estimator such that the **excess expected risk**

$$\mathcal{E}(f_S) - \mathcal{E}(f^*),$$

is small

The latter requirement needs some care since  $f_S$  depends on the training set and hence is **random**



## Generalization and Consistency

A natural approach is to consider the expectation of the excess expected risk

$$\mathbb{E}_S[\mathcal{E}(f_S) - \mathcal{E}(f^*)]$$

## Generalization and Consistency

A natural approach is to consider the expectation of the excess expected risk

$$\mathbb{E}_S[\mathcal{E}(f_S) - \mathcal{E}(f^*)]$$

- ▶ A basic requirement is **consistency**

$$\lim_{n \rightarrow \infty} \mathbb{E}_S[\mathcal{E}(f_S) - \mathcal{E}(f^*)] = 0$$

## Generalization and Consistency

A natural approach is to consider the expectation of the excess expected risk

$$\mathbb{E}_S[\mathcal{E}(f_S) - \mathcal{E}(f^*)]$$

- ▶ A basic requirement is **consistency**

$$\lim_{n \rightarrow \infty} \mathbb{E}_S[\mathcal{E}(f_S) - \mathcal{E}(f^*)] = 0$$

- ▶ **Learning rates** provide finite sample information, for all  $\epsilon > 0$  if  $n \geq n(\epsilon)$ , then

$$\mathbb{E}_S[\mathcal{E}(f_S) - \mathcal{E}(f^*)] \leq \epsilon,$$

- ▶  $n(\epsilon)$  is called **sample complexity**

## Generalization: Fitting and Stability

How to design a good algorithm?

## Generalization: Fitting and Stability

How to design a good algorithm?

Two concepts are key:

## Generalization: Fitting and Stability

How to design a good algorithm?

Two concepts are key:

- ▶ **Fitting**: an estimator should *fit* data well

## Generalization: Fitting and Stability

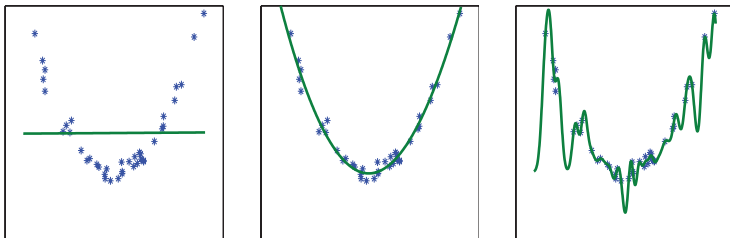
How to design a good algorithm?

Two concepts are key:

- ▶ **Fitting:** an estimator should *fit* data well
- ▶ **Stability:** an estimator should be stable, it should not change much if data change slightly

## Generalization: Fitting and Stability

How to design a good algorithm?



We say that an algorithm **overfits**, if it fits the data while being unstable

We say that an algorithm **oversmooths**, if it is stable while disregarding the data



## Regularization as a Fitting-Stability Trade-off

- ▶ Most learning algorithms depend on one (or more) **regularization parameter**, that controls the trade-off between *data-fitting* and *stability*
- ▶ We broadly refer to this class of approaches as **regularization algorithms**, our main topic of discussion

## Wrapping up

In this class, we introduced the basic definitions in statistical learning theory, including the key concepts of overfitting, stability and generalization.

## Next Class

We will introduce the a first basic class of learning methods, namely local methods, and study more formally the fundamental trade-off between overfitting and stability.