MIT 9.520/6.860, Fall 2019
Statistical Learning Theory and Applications

Class 02: Statistical Learning Setting

Lorenzo Rosasco

Learning from examples

- Machine Learning deals with systems that are trained from data rather than being explicitly programmed.

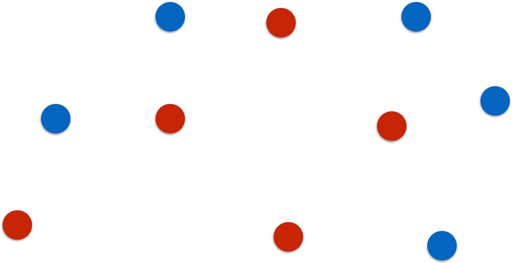
- Here we describe the framework considered in statistical learning theory.

All starts with DATA

- **Supervised**: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$.

- Unsupervised: $\{x_1, \ldots, x_m\}$.

- Semi-supervised: $\{(x_1, y_1), \ldots, (x_n, y_n)\} \cup \{x_1, \ldots, x_m\}$.



The supervised learning problem

- $X \times \mathbb{R}$ probability space, with measure P.
- $\ell : Y \times Y \to [0, \infty)$, measurable *loss function*.

Define **expected risk**:

$$L(f) = \; L(f) = \mathbb{E}_{(x,y)\sim P}[\ell(y, f(x))]$$

Problem: Solve

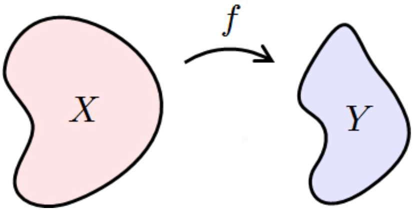
$$\min_{f:X\to Y} L(f),$$

given only

$$S_n = (x_1, y_1), \ldots, (x_n, y_n) \sim P^n,$$

i.e. n i.i.d. samples w.r.t. P fixed, but unknown.

Data space



$$\underbrace{X}_{\text{input space}} \qquad \underbrace{Y}_{\text{output space}}$$

Input space

X input space:

- Linear spaces, e. g.
  - vectors,
  - functions,
  - matrices/operators.

- "Structured" spaces, e. g.
  - strings,
  - probability distributions,
  - graphs.

Output space

Y output space:

- linear spaces, e. g.
  - $Y = \mathbb{R}$, regression,
  - $Y = \mathbb{R}^T$, multitask regression,
  - Y Hilbert space, functional regression.

- "Structured" spaces, e. g.
  - $Y = \{-1, 1\}$, classification,
  - $Y = \{1, \dots, T\}$, multicategory classification,
  - strings,
  - probability distributions,
  - graphs.

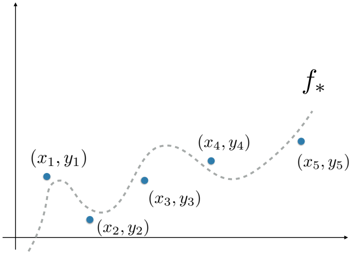
Probability distribution

Reflects *uncertainty* and *stochasticity* of the learning problem,

$$P(x, y) = P_X(x)P(y|x),$$

- P_X marginal distribution on X,
- $P(y|x)$ conditional distribution on Y given $x \in X$.

Conditional distribution and noise



Regression

$$y_i = f_*(x_i) + \epsilon_i.$$

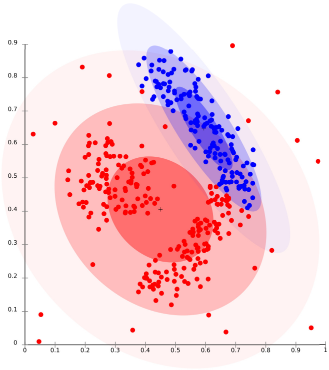
- Let $f_* : X \to Y$, fixed function,
- $\epsilon_1, \dots, \epsilon_n$ zero mean random variables, $\epsilon_i \sim N(0, \sigma)$,
- $x_1, \dots, x_n$ random,

$$P(y|x) = N(f^*(x), \sigma).$$

Conditional distribution and misclassification

Classification

$$P(y|x) = \{P(1|x), P(-1|x)\}.$$

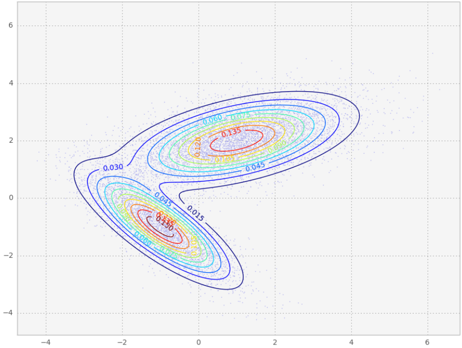


Noise in classification: overlap between the classes,

$$\Delta_\delta = \left\{ x \in X \;\middle|\; \left| P(1|x) - 1/2 \right| \leq \delta \right\}.$$

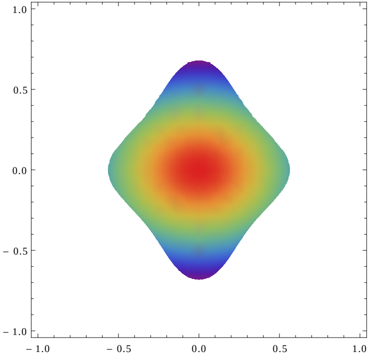
Marginal distribution and *sampling*

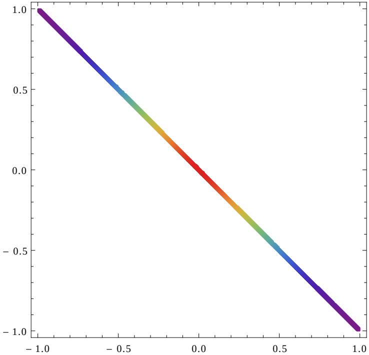
P_X takes into account uneven sampling of the input space.



Marginal distribution, densities and manifolds

$$p(x) = \frac{dP_X(x)}{dx} \Rightarrow p(x) = \frac{dP_X(x)}{d\,\mathrm{vol}(x)}$$





Loss functions

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

- Cost of predicting $f(x)$ in place of y.
- Measures the *pointwise error* $\ell(y, f(x))$.
- Part of the problem definition since $L(f) = \int_{X\times Y} \ell(y, f(x)) dP(x, y)$.

Note: sometimes it is useful to consider loss of the form

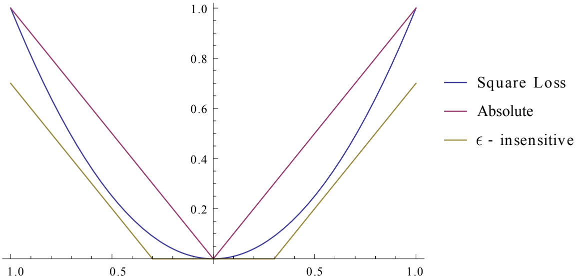
$$\ell : Y \times \mathcal{G} \to [0, \infty)$$

for some space $\mathcal{G}$, e.g. $\mathcal{G} = \mathbb{R}$.

Loss for regression

$$\ell\ell(y, y') = V(y - y'), \quad V : \mathbb{R} \to [0, \infty).$$

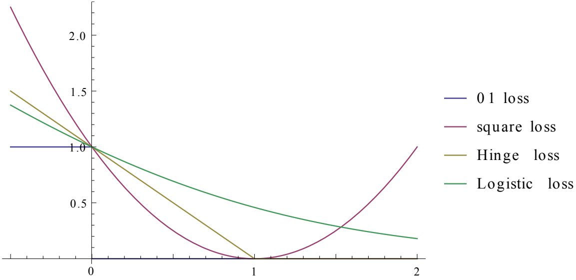
- Square loss $\ell(y, y') = (y - y')^2$.
- Absolute loss $\ell(y, y') = |y - y'|$.
- ϵ-insensitive $\ell(y, y') = \max(|y - y'| - \epsilon, 0)$.



Loss for classification

$$\ell(y, y') = V(-yy'), \quad V : \mathbb{R} \to [0, \infty).$$

- 0-1 loss $\ell(y, y') = \Theta(-yy'), \quad \Theta(a) = 1$, if $a \geq 0$ and 0 otherwise.
- Square loss $\ell(y, y') = (1 - yy')^2$.
- Hinge-loss $\ell(y, y') = \max(1 - yy', 0)$.
- Logistic loss $\ell(y, y') = \log(1 + \exp(-yy'))$.



Loss function for structured prediction

Loss specific for each learning task, e.g.

- Multiclass: square loss, weighted square loss, logistic loss, ...
- Multitask: weighted square loss, absolute, ...
- ...

Expected risk

$$L(f) = \mathbb{E}_{(x,y)\sim P}[\ell(y, f(x))] = \int_{X\times Y} \ell(y, f(x))dP(x, y),$$

with

$$f \in \mathcal{F}, \quad \mathcal{F} = \{f : X \to Y \mid f \text{ measurable}\}.$$

Example

$$Y = \{-1, +1\}, \quad \ell(y, f(x)) = \Theta(-yf(x))^{\ 1}$$

$$L(f) = \mathbb{P}(\{(x, y) \in X \times Y \mid f(x) \neq y\}).$$

---

[1] $\Theta(a) = 1$, if $a \geq 0$ and 0 otherwise.

Target function

$$f_P \quad = \quad \arg\min_{f \in \mathcal{F}} \; L(f),$$

can be derived for many loss functions.

$$L(f) = \int dP(x,y)\ell(y, f(x)) = \int dPX(x) \underbrace{\int \ell(y, f(x))dP(y|x)}_{L_x(f(x))},$$

It is possible to show that:

- $\inf_{f \in \mathcal{F}} L(f) = \int dP_X(x) \inf_{a \in \mathbb{R}} L_x(a)$.
- Minimizers of $L(f)$ can be derived "pointwise" from the inner risk $L_x(f(x))$.
- Measurability of this pointwise definition of f_P can be ensured.

Target functions in regression

$$f_P(x) = \arg\min_{a \in \mathbb{R}} L_x(a).$$

square loss

$$f_P(x) = \int_Y y dP(y|x).$$

absolute loss

$$f_P(x) = \textbf{median}(P(y|x)),$$

$$\textbf{median}(p(\cdot)) = y \;\; \text{s.t.} \;\; \int_{-\infty}^{y} t dp(t) = \int_{y}^{+\infty} t dp(t).$$

Target functions in classification

misclassification loss

$$f_P(x) = \mathbf{sign}(P(1|x) - P(-1|x)).$$

square loss

$$f_P(x) = P(1|x) - P(-1|x).$$

logistic loss

$$f_P(x) = \log \frac{P(1|x)}{P(-1|x)}.$$

hinge-loss

$$f_P(x) = \mathbf{sign}(P(1|x) - P(-1|x)).$$

Different loss, different target

- Each loss functions defines a different optimal target function. Learning enters the picture when the latter is impossible or hard to compute (as in simulations).

- As we see in the following, loss functions also differ in terms of induced computations.

Learning algorithms

Solve

$$\min_{f \in \mathcal{F}} L(f),$$

given only

$$S_n = (x_1, y_1), \ldots, (x_n, y_n) \sim P^n.$$

Learning algorithm

$$S_n \to \widehat{f}_n = \widehat{f}_{S_n}.$$

f_n estimates f_P given the observed examples S_n.

How to measure the error of an estimate?

Excess risk

Excess risk:

$$L(\widehat{f}) - \min_{f \in \mathcal{F}} L(f).$$

Consistency: For any $\epsilon > 0$,

$$\lim_{n \to \infty} \mathbb{P}\left(L(\widehat{f}) - \min_{f \in \mathcal{F}} L(f) > \epsilon \right) = 0.$$

Other forms of consistency

Consistency in Expectation: For any $\epsilon > 0$,

$$\lim_{n\to\infty} \mathbb{E}[L(\widehat{f}) - \min_{f\in\mathcal{F}} L(f)] = 0.$$

Consistency almost surely: For any $\epsilon > 0$,

$$\mathbb{P}\left(\lim_{n\to\infty} L(\widehat{f}) - \min_{f\in\mathcal{F}} L(f) = 0\right) = 1.$$

Note: different notions of consistency correspond to different notions of convergence for random variables: weak, in expectation and almost sure.

Sample complexity, tail bounds and error bounds

- Sample complexity: For any $\epsilon > 0, \delta \in (0,1]$, when $n \geq n_{P,\mathcal{F}}(\epsilon, \delta)$,

$$\mathbb{P}\left(L(\widehat{f}) - \min_{f \in \mathcal{F}} L(f) \geq \epsilon\right) \leq \delta.$$

- *Tail bounds*: For any $\epsilon > 0, n \in \mathbb{N}$,

$$\mathbb{P}\left(L(\widehat{f}) - \min_{f \in \mathcal{F}} L(f) \geq \epsilon\right) \leq \delta_{P,\mathcal{F}}(n, \epsilon).$$

- *Error bounds*: For any $\delta \in (0,1]$, $n \in \mathbb{N}$,

$$\mathbb{P}\left(L(\widehat{f}) - \min_{f \in \mathcal{F}} L(f) \leq \epsilon_{P,\mathcal{F}}(n, \delta)\right) \geq 1 - \delta.$$

No free-lunch theorem

A good algorithm should have small sample complexity for many distributions P.

No free-lunch

Is it possible to have an algorithm with small (finite) sample complexity for **all** problems?

The no free lunch theorem provides a negative answer.

In other words given an algorithm there exists a problem for which the learning performance are arbitrarily bad.

Algorithm design: complexity and regularization

The design of most algorithms proceed as follows:

- Pick a (possibly large) class of function $\mathcal{H}$, ideally

$$\min_{f \in \mathcal{H}} L(f) = \min_{f \in \mathcal{F}} L(f)$$

- Define a procedure $A_\gamma(S_n) = \hat{f}_\gamma \in \mathcal{H}$ to *explore* the space $\mathcal{H}$

Bias and variance

Let f_γ be the solution obtained with an infinite number of examples.

Key error decomposition

$$L(\hat{f}_\gamma) - \min_{f \in \mathcal{H}} L(f) = \underbrace{L(\hat{f}_\gamma) - L(f_\gamma)}_{Variance/Estimation} + \underbrace{L(f_\gamma) - \min_{f \in \mathcal{H}} L(f)}_{Bias/Approximation}$$

Small Bias lead to good data fit, high variance to possible instability.

ERM and structural risk minimization

A classical example.

Consider $(\mathcal{H}_\gamma)_\gamma$ such that

$$\mathcal{H}_1 \subset \mathcal{H}_2, \dots \mathcal{H}_\gamma \subset \dots \mathcal{H}$$

Then, let

$$\hat{f}_\gamma = \min_{f \in \mathcal{H}_\gamma} \widehat{L}(f), \qquad \widehat{L}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$$

Example
$\mathcal{H}_\gamma$ are functions $f(x) = w^\top x$ (or $f(x) = w^\top \Phi(x)$), s.t.$\|w\| \leq \gamma$

Beyond constrained ERM

In this course we will see other algorithm design principles:

- Penalization
- Stochastic gradient descent
- Implicit regularization
- Regularization by projection

Beyond supervised learning

- Z probability space, with measure P.
- $\mathcal{H}$ a set.
- $\ell : Z \times \mathcal{H} \to [0, \infty)$, measurable *loss function*.

Problem: Solve

$$\min_{h \in \mathcal{H}} \mathbb{E}_{z \sim P}[\ell(z, h)],$$

given only

$$S_n = z_1, \dots, z_n \sim P^n,$$

i.e. n i.i.d. samples w.r.t. P fixed, but unknown.

- $\mathcal{H}$ is part of the definition of the problem
- The above setting covers for example many unsupervised learning problems as well as decision theory problem (aka, general learning setting).