

# STATISTICAL METHODS FOR MACHINE LEARNING

giosumarin

March 2021

## Contents

<b>1</b>	<b>Lezione 1</b>	<b>2</b>
1.1	Label set . . . . .	2
1.2	Loss function . . . . .	3
<b>2</b>	<b>Lezione 2</b>	<b>3</b>
2.1	Data Points . . . . .	3
2.2	Predictor . . . . .	3
2.3	Supervised learning . . . . .	4
2.3.1	Training Set . . . . .	4
2.3.2	Test Set . . . . .	4
2.3.3	Completo . . . . .	4
2.4	Empirical Risk Minimizer . . . . .	4
2.4.1	Esempio . . . . .	4
<b>3</b>	<b>Lezione 3</b>	<b>5</b>

## 1 Lezione 1

- clustering: raggruppare punti in accordo alla loro similarità (raggruppare clienti per soldi spesi);
- classification: predire label semantiche associate ai data points (classificare documenti per argomento);
- planning: vogliamo decidere una sequenza di azioni che devono essere fatte per raggiungere un goal (robot che va da qualche parte con ostacoli sul percorso o guida autonoma).
- supervised learning: abbiamo label per degli esempi e imparo a classificare d questi
- unsupervised learning: clustering (label "attaccata" ai data points)

## 1.1 Label set

- $Y$  label set
- news classification:  $Y = \{\text{sport, politica, business, ...}\}$
- predizione stock price:  $Y \in \mathbb{R}$
- classification/categorization:  $Y$  insieme finito di simboli,  $\hat{y} \stackrel{?}{=} y$ , con  $\hat{y}$  predizione e  $y$  valore reale;
- regression:  $Y \in \mathbb{R}$ ,  $|\hat{y} - y|$ .

## 1.2 Loss function

$$l(y, \hat{y}) = \begin{cases} 0 & \text{se } y = \hat{y} \\ 1 & \text{altrimenti} \end{cases}$$

$Y = \{\text{spam (positivo), nonspam (negativo)}\}$ , binary classification problem

$$l(y, \hat{y}) = \begin{cases} 2 & \text{se } y = \text{nonspam e } \hat{y} = \text{spam} \leftarrow \text{falso positivo} \\ 1 & \text{se } y = \text{spam e } \hat{y} = \text{nonspam} \leftarrow \text{falso positivo} \\ 0 & \text{altrimenti} \end{cases}$$

absolute loss (per regressione):  $l(y, \hat{y}) = |\hat{y} - y|$

square loss (per regressione):  $l(y, \hat{y}) = (\hat{y} - y)^2$

[ESEMPIO] previsioni meteo:  $Y = \{\text{pioggia, asciutto}\}$

$\hat{y}$  = probabilità assegnata a pioggia; prediction set:  $Z = \{0, 1\}$

$$l(y, \hat{y}) = |\hat{y} - y|$$

$$l(y, \hat{y}) = \begin{cases} \ln \frac{1}{\hat{y}} & \text{se } y = 1 \\ \ln \frac{1}{1-\hat{y}} & \text{se } y = 0 \end{cases}$$

La loss logaritmica ha le seguenti proprietà:

- $\lim_{\hat{y} \rightarrow 0^+} l(1, \hat{y}) = \infty$
- $\lim_{\hat{y} \rightarrow 1^-} l(0, \hat{y}) = \infty$

## 2 Lezione 2

### 2.1 Data Points

$X$  dominio dati,  $x$  spesso è codificato convenientemente come vettore di numeri attraverso per esempio la one-hot encoding.

$$X = \begin{cases} \mathbb{R}^d & \text{attributi numerici} \\ X_1, \dots, X_d & \text{attributi categorici} \end{cases}$$

Possiamo avere anche un mix di diversi attributi.

## 2.2 Predictor

Un predittore è una funzione che mappa data points in label

$$f : X \rightarrow Y, f : X \rightarrow \overline{Z}, \overline{Z} \neq Y$$

Dato un punto  $x$  abbiamo quindi

$$\hat{y} = f(x).$$

Quello che vogliamo è avere una loss piccola per molti  $x \in X$ .

## 2.3 Supervised learning

Abbiamo le coppie  $(x, y)$  con  $x$  singolo data point e  $y$  la sua rispettiva label. Le label possono essere soggettive (annotazioni umane) o oggettive (misurazioni di strumenti).

### 2.3.1 Training Set

Insieme di esempi su cui effettuiamo l'addestramento; abbiamo quindi un training set in input a un algoritmo di apprendimento (con la sua loss) e che in output genera un predittore.

### 2.3.2 Test Set

Insieme di esempi ( $\neq$  training set) su cui viene valutata la capacità di generalizzazione di un predittore addestrato sul training set.

### 2.3.3 Completo

Abbiamo il predittore  $f$  uscente dall'algoritmo di apprendimento  $A$  usando la funzione di loss  $l$ . Abbiamo il test set  $(x'_1, y'_1), \dots, (x'_n, y'_n)$ , calcoliamo il nostro test error come

$$\frac{1}{n} \sum_t^n l(y'_t, f(x'_t)).$$

Il nostro goal è quello di sviluppare una teoria per guidare nel design di  $A$  che ci genera predittori con un piccolo test error w.r.t. una loss function.

## 2.4 Empirical Risk Minimizer

Fisso un insieme  $F$  di predittori e una loss function  $f$ . Entra quindi il training set  $(S)$  in questo ERM (che ha  $F$  e  $l$ ) e abbiamo in output

$$\hat{f} \in \arg \min_{f \in F} \hat{l}_S(f).$$

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$f^*$	-1	1	1	$f^*(x_4)$	$f^*(x_5)$
$f^1$	-1	1	1	1	1
$f^2$	-1	1	1	-1	1
$f^3$	-1	1	1	1	-1
$f^4$	-1	1	1	-1	-1

L'idea è di minimizzare il training error in una classe  $F$  di predittori. Se  $\min_{f \in F} \frac{1}{n} \sum_{t=1}^n l(y'_t, f(x'_t))$  è grande siamo in un caso di underfitting.

### 2.4.1 Esempio

Prendiamo  $F$  grande e vediamo cosa succede.

$$X = \{x_1, \dots, x_5\}, Y = \{-1, 1\}, F \text{ contiene tutti i classificatori binari}$$

$$|F| = 2^5 = 32, \exists f^* \text{ t.c. } y_t = f^*(x_t) \text{ con } t = \{1, \dots, 5\}$$

Se il training set è formato dai primi 3 data point tutti e 4 i predittori hanno lo stesso training error uguale a 0. In questo caso non possiamo decidere quale predittore usare. Chiamo questo caso overfitting.

Possiamo estrapolare la seguente regola da questo esempio (quando  $F$  è finito):

$$m \geq \log_2 |F|$$

## 3 Lezione 3