Appunti di machine learning

Daniele Besozzi

Anno accademico 2025/2026

Contents

1 Introduzione **2**
1.1 Vettori e matrici 2
1.2 Norme di vettori e matrici 2
1.3 Notazioni generiche 3
1.4 Rischio atteso e rischio empirico 5

Premesse

Questi sono appunti realizzati per riassumere e schematizzare tutti i concetti presentati durante il corso di machine learning tenuto presso il corso di laurea magistrale in informatica presso l'università degli studi di Milano Bicocca. Lo scopo di questo documento non è quello di sostituire le lezioni del corso o di essere l'unica fonte di studio, bensì integrare le altri fonti con un documento riassuntivo.
Mi scuso in anticipo per eventuali errori e prego i lettori di segnalarli contattandomi via mail all'indirizzo d.besozzi@campus.unimib.it.

Chapter 1

Introduzione

In questo capitolo presenterò gli aspetti matematici fondamentali per andare ad affrontare gli argomenti del corso.

1.1 Vettori e matrici

Denotiamo un vettore riga e colonna rispettivamente con (a, b, c) e $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$

dove $a, b, c \in \mathbb{R}$ sono scalari.
In generale denotiamo con le lettere maiuscole le matrici, e.g. X e i suoi elementi con X_{ij}.
$x \in \mathbb{R}^n$ è un vettore di n elementi e $X \in \mathbb{R}^{m \times n}$ è una matrice di dimensione $m \times n$.

1.2 Norme di vettori e matrici

Dato un vettore $x \in \mathbb{R}^n$ vi sono diversi tipi di norme comunemente utilizzate.

- $||x||_2 = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}}$ è il tipo più comune e viene chiamato **norma 2** di un vettore o **norma Euclidea**. Normalmente è denotato semplicemente con $||x||$.
- $||x||_1 = |x_1| + |x_2| + \cdots + |x_n|$, detta la **norma 1** o **distanza di Manhattan**
- $||x||_\infty = \max_{1 \leq i \leq n} |x_i|$, detta la **norma** ∞.

Analogamente, per una matrice $X \in \mathbb{R}^{m \times n}$, si possono definire diverse norme:

- **Norma di Frobenius**: $||X||_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |X_{ij}|^2 \right)^{\frac{1}{2}}$
- **Norma spettrale (o norma 2)**: $||X||_2$
- **Norma 1**: $||X||_1 = \sum_{i,j} |X_{ij}|$

1.3 Notazioni generiche

Siano:

- u: variabile indipendente (input), non necessariamente un vettore o uno scalare
- v: variabile dipendente (output), come sopra

allora abbiamo che:

- $x = \Phi(u)$, dove $x \in \mathbb{R}^d$ è il vettore di features e Φ è la funzione di mapping o embedding.
- $y = \Psi(v)$, dove $y \in \mathbb{R}^m$ è il vettore target (o di output) e Ψ è la funzione di mapping di feature in output.

Siano $x^1, \dots x^n$ e $y^1, \dots y^n$ due dataset di n esempi, dove x^i e y^i formano la i-esima coppia di dati. Dunque n è il numero di campioni, allora posso associarvi le due matrici dei dati

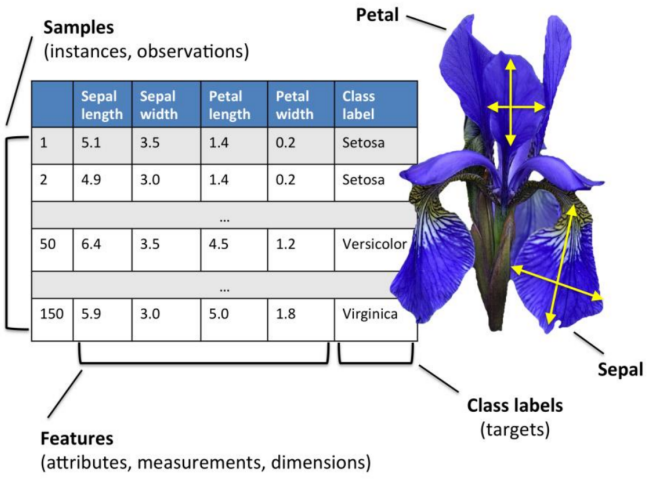
$$X = \begin{bmatrix} (x^1)^T \\ \vdots \\ (x^n)^T \end{bmatrix} \in \mathbb{R}^{n \times d}, \quad Y = \begin{bmatrix} (y^1)^T \\ \vdots \\ (y^n)^T \end{bmatrix} \in \mathbb{R}^{n \times m}$$

Le cui righe sono i vettori feature e i vettori target rispettivamente, trasposti.
Definiamo allora:

- $g_\theta : \mathbb{R}^d \to \mathbb{R}^m$ è un predittore.
- $\hat{y} = g_\theta(x)$ è la predizione di y, dato x.
- $\Theta \in \mathbb{R}^p$ è il vettore di parametri del predittore.

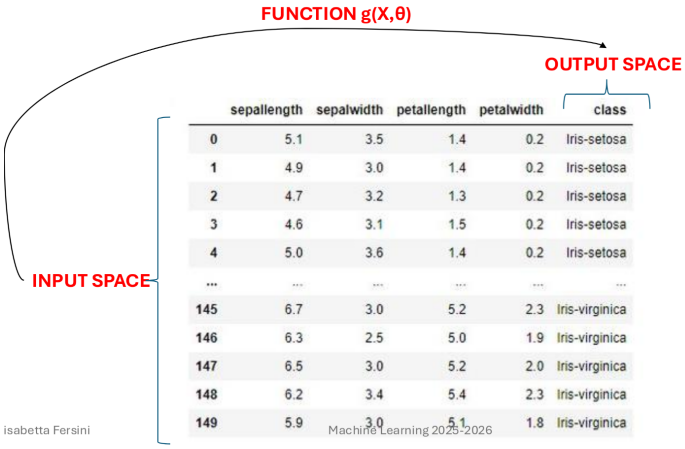
La scelta dei parametri Θ a seconda dei dati viene chiamato *training* o *fitting* del predittore.
Separando le definizioni in base al dominio di riferimento, abbiamo le seguenti suddivisioni:

- Spazio di input:
  - Istanza (sample, oggetto, record): un esempio descritto da un certo numero di attributi.
  - Attributo (campo, caratteristica, variabile): misura di un aspetto di una istanza.
- Spazio di output:
  - Classe (label, target): categoria a cui appartiene una istanza.
- Predittore o modello:
  - Una funzione g con parametri Θ che dato uno spazio di input X produce uno spazio di output Y.produce una predizione nello spazio di output Y.



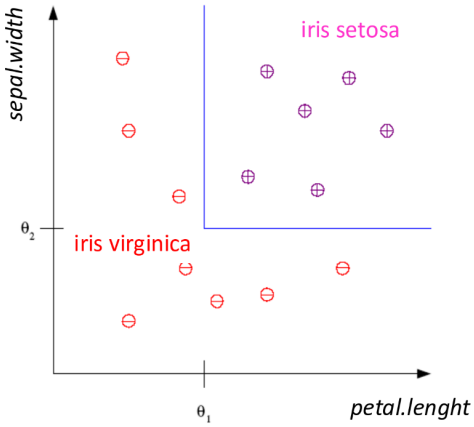
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Figure 1.1: Esempi pratici.



	sepallength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

Figure 1.2: Esempi pratici 2.



MODEL $f(X,\theta)$: IF *petal.lenght* > θ_1 AND *sepal.width* > θ_2

Figure 1.3: Esempi pratici 3.

1.4 Rischio atteso e rischio empirico

Quando un modello di machine learning g viene addestrato, vogliamo che abbia buone prestazioni non solo per i dati utilizzati per il training, ma anche per dati sconosciuti (*generalizzazione*). Dovremo stimare il **rischio atteso**, ovvero la loss media che dovrebbe presentarsi sulla reale distribuzione dei dati $P(x, y)$.
Supponiamo di avere un dataset di n coppie (x^i, y^i), dove x^i è il vettore di feature e y^i il vettore target associato alla i-esima istanza. Definiamo la **loss function** $L(g_\Theta(x), y)$, che misura l'errore commesso dal modello g_Θ.
Dunque il rischio atteso è definito come:

$$R(g_\Theta) = E_{(x,y)\sim P}[L(g_\Theta(x), y)]$$

Da notare però che la reale distribuzione P non è nota, dunque non possiamo stimare il rischio atteso. Le cause sono:

1. Distribuzione non nota
   Osserviamo solo un campione finito di campioni estratti dal mondo reale. La distribuzione completa che genera quei dati non è accessibile.

2. Impossibilità pratica
   Anche se conoscessimo il processo di generazione in teoria, calcolare esattamente la predizione del rischio è impossibile in generale perché richiederebbe di sommare/integrare tutte i possibili esempi.

3. Dati finiti e rumorosi
   Il dataset a disposizione è finito, spesso presenta rumore e errori di misura, o bias di raccolta. Dunque possiamo solo approssimare il rischio utilizzando una **stima empirica**.

Dunque, vogliamo stimare e minimizzare il rischio empirico. Supponiamo di avere un dataset di n coppie (x^i, y^i). Definiamo la loss function $L(g_\Theta(x), y)$, che misura l'errore commesso dal modello g_Θ. Allora il rischio empirico $\hat{R}(g_\Theta)$ è definito come:

$$\hat{R}(g_\Theta) = \frac{1}{n}\sum_{i=1}^{n} L(g_\Theta(x^i), y^i)$$

La maggior parte degli algoritmi di machine learning minimizzano il rischio empirico.

$$g^* = arg\min_{g_\Theta \in G} \hat{R}(g_\Theta)$$