

CONCETTI CHIAVE DI MACHINE
LEARNING

Gemma Martini

7 giugno 2017

Indice

1	Introduzione	3
2	Modelli supervisionati	3
2.1	Definizioni preliminari	3
2.2	Concept learning	4
2.2.1	Algoritmi per il concept learning	4
2.3	Regressione	5
2.3.1	Algoritmi per la regressione	6
2.4	Concept learning	6
2.5	Concept learning	7
3	Modelli non supervisionati	7
3.1	Definizioni	7

Questi appunti sono frutto dello studio personale della materia e non rappresentano un sostituto del materiale fornito dai docenti, bensì uno strumento di ripasso, opportunamente filtrato dal personale senso critico.

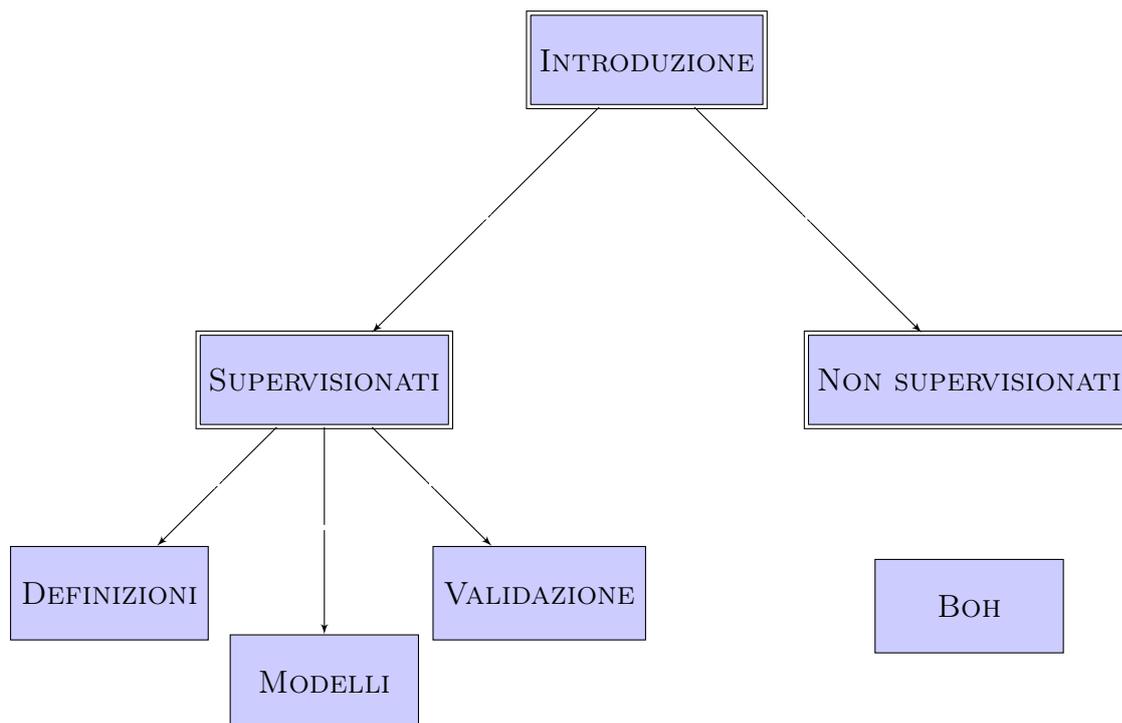


Figura 1: Organigramma delle dispense sul ML

1 Introduzione

Lo scopo del *machine learning* è quello di indovinare funzioni (in una o più variabili) a valori discreti (si parla in questo caso di **classificazione**) o continui (problema di **regressione**). Oltre alla suddivisione sopracitata, vi è un altro modo di classificare i problemi di *machine learning*: supervisionati e non supervisionati. I problemi **supervisionati** sono quelli che dispongono di un *training set*, ossia un insieme sul quale il **modello** (ossia ciò che definisce l'insieme delle funzioni candidate) viene specializzato. I modelli non supervisionati, invece, si occupano di trovare caratteristiche del mondo senza avere degli esempi di soluzione. In figura 1 un diagramma della struttura delle parti che seguono.

2 Modelli supervisionati

Dato un insieme di esempi $\langle input, output \rangle = \langle x, d \rangle$ si vuole ricavare la funzione f , detta funzione **target** o funzione **obiettivo**.

Per ricavare tale funzione si definisce **ipotesi** la funzione h , mediante la quale si cerca di approssimare f .

2.1 Definizioni preliminari

Definizione 2.1 (Modello)

Un **modello** o **spazio delle ipotesi** definisce una classe di funzioni all'interno della quale si muove la ricerca di un'ipotesi.

Definizione 2.2 (Esempi di training)

Una coppia della forma $(x, f(x))$, dove x è un vettore di attributi del modello e $f(x)$ è l'immagine di tali attributi si definisce **training set** o **insieme di training**.

Definizione 2.3 (Algoritmo di apprendimento)

Un **algoritmo di apprendimento** o **learning algorithm** è la ricerca all'interno dello spazio delle ipotesi H dell'ipotesi migliore, ossia la funzione che più si avvicina alla funzione target.

Definizione 2.4 (Generalizzazione)

La **generalizzazione** è la capacità di predire, ossia rispondere in modo accurato per esempi che non fanno parte del *training set*. Al concetto di generalizzazione si lega il concetto di **errore di generalizzazione**, ossia quanto l'ipotesi trovata si discosta dal valore della funzione *target* sui casi non di *training*.

Definizione 2.5 (Bias)

Un **bias** è una **preferenza**, che porta a considerare soltanto alcune soluzioni. Un *bias* può essere di due tipi:

Language bias Assunzione fatta sul modello, ossia restrizione dello spazio delle ipotesi ad alcune famiglie di funzioni

Research bias Restrizione fatta dall'algoritmo di ricerca, che permette di selezionare soltanto alcune ipotesi

2.2 Concept learning



Di seguito alcune nozioni utili:

- Il **concept learning** corrisponde all'individuare una funzione booleana da variabili booleane. In questo contesto è possibile la definizione di un **concetto**: $C(x) = true$ se c è un gatto.
- Si dice che l'ipotesi **soddisfa** x se $h(x) = 1$.
- Si dice che un'ipotesi h è **consistente** con un esempio di training $\langle x, c(x) \rangle$ se $h(x) = c(x)$.
- Date h_j e h_k funzioni a valori booleani definite su X . si dice che h_j è **più generale** di h_k se $\forall x \in X (h_k(x) = 1) \Rightarrow (h_j(x) = 1)$.
- Per semplicità, lo spazio delle ipotesi è formato da funzioni in **forma congiuntiva**.

2.2.1 Algoritmi per il concept learning

Find-S

- **Algoritmo** Inizializzazione con l'ipotesi più specifica, per poi generalizzare alla prima collisione.
- **Pro** Un'ipotesi così ottenuta è garantita essere **consistente**.

- **Contro** Non ha tolleranza del rumore, non può dire se la funzione trovata è la migliore del modello

Version spaces Un **version space** è un sottoinsieme dello spazio delle ipotesi consistente con tutti gli esempi di training.

- **Algoritmo** Partendo dai due estremi (massima generalità e massima specificità) ci si muove verso un punto di incontro tra i due estremi.
- **Pro**
- **Contro**

Algoritmi per i modelli lineari Un **version space** è un sottoinsieme dello spazio delle ipotesi consistente con tutti gli esempi di training.

- **Algoritmo**
- **Pro**
- **Contro**

Decision tree Un **version space** è un sottoinsieme dello spazio delle ipotesi consistente con tutti gli esempi di training.

- **Algoritmo**
- **Pro**
- **Contro**

2.3 Regressione



Di seguito alcune nozioni utili:

- Un **modello lineare** per un problema di regressione corrisponde all'assunzione di restringersi ad una funzione lineare che minimizzi la distanza dai dati del *training set*.

- Si definisce **empirical loss** o **errore** $L(h_w) = \sum_{p=1}^l (y_p - h_w(x_p))^2 = \sum_{p=1}^l (y_p - \sum_{i=0}^n w_i x_{pi})^2 = \|y - Xw\|^2$, con $X \in \mathcal{M}(l, n, \mathbb{R})$.

- Un **modello polinomiale** dato un problema di regressione è un modello che assume come *language bias* lo spazio delle funzioni polinomiali. In questo modello $h(x) = \sum_{k=0}^K w_k \Phi_k(x)$, dove $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$.

- Si parla di **underfitting** quando un'ipotesi non ottiene un punteggio abbastanza alto sul *training set* per la troppa semplicità/

- Si parla di **overfitting** quando un'ipotesi ottiene un punteggio alto sull'insieme di *training*, ma basso sull'insieme di *test* (se $\exists h'$ t.c. $h'(x) \leq h(x)$ sul *training set*, ma $h'(x) \leq h(x)$ sul *test set*).
- Si parla di **regolarizzazione** quando si cerca di evitare il problema dell'*overfitting* aggiungendo vincoli allo spazio delle ipotesi.

2.3.1 Algoritmi per la regressione

LMS (Least Mean Square) - modello lineare

- **Algoritmo** Sfruttare il metodo di discesa del gradiente per minimizzare la *loss*. In particolare, ad ogni passo (dato $\nabla(E(w)) = \delta \frac{E(w)}{\delta w} = \Delta w$) si sceglie w' come $w' = w + \eta \Delta w$.
- **Pro** È quello con minor complessità computazionale
- **Contro**

Metodo diretto - modello lineare Trovare i punti stazionari.

- **Algoritmo** Risolvere il sistema in cui il gradiente è imposto a 0 ed il jacobiano (matrice delle derivate parziali del secondo ordine) è maggiore di 0
- **Pro**
- **Contro**

Metodo del gradiente coniugato - modello lineare

- **Algoritmo** Si comporta in modo simile a LMS, ma al k-esimo *step* modifica il gradiente in modo che sia ortogonale alle direzioni in cui si è mosso agli step precedenti
- **Pro**
- **Contro**

Regolarizzazione di Tikhonov - modello non lineare Questo non è un algoritmo per la selezione di un'ipotesi, ma una correzione del calcolo dell'errore per penalizzare "funzioni complesse"

- **Algoritmo** $E(w) = \sum_{p=1}^l (y_p - h_w(x_p))^2 + \lambda \|w\|^2$
- **Pro** Aiuta ad evitare l'*overfitting*
- **Contro** Se λ non è scelto correttamente c'è il rischio di penalizzare le funzioni che si vogliono selezionare

2.4 Concept learning



2.5 Concept learning



3 Modelli non supervisionati

3.1 Definizioni