UNIVERSITÀ DI PISA
Dipartimento di Informatica

MACHINE LEARNING

Appunti dalle lezioni del Prof. Alessio Micheli

Pasquale Miglionico
Guido Narduzzi
Enrico Negri
Filippo Quattrocchi
Francesco Zigliotto

ANNO ACCADEMICO 2019–2020

INDICE

1 INTRODUZIONE

1.1 LEZIONE DI GIOVEDÌ 26 SETTEMBRE

L'obiettivo è insegnare ad un sistema un compito preciso, costruendo un modello utilizzabile per predire il corretto output, dopo aver esaminato un gran numero di esempi. Tale metodo di apprendimento è detto *per generalizzazione*.

È utile per esempio quando l'approccio teorico a un determinato problema è difficilmente praticabile, o quando i dati in input sono poco accurati, affetti da errore o incompleti.

1.1.1 Terminologia

Definizione 1.1. (Machine Learning). Il *Machine Learning* studia e propone metodi per inferire funzioni o correlazioni che, a partire da dati osservati, producano il corretto output sui *samples* forniti e li generalizzino con ragionevole accuratezza.

Un *machine learning system* si compone di *dati*, *tasks*, *modelli*, *algoritmi di apprendimento* e *validazione*.

Definizione 1.2. (Dati). I *dati* rappresentano le *esperienze disponibili*. Possono essere organizzati in un certo numero l di istanze x_p (*samples*, *instances*), ciascuno contenente n attributi (*features*). Con $x_{p,j}$ indicheremo l'attributo j-esimo della p-esima istanza.

Risulta conveniente indicare i valori dei vari attributi come vettori di dimensione k (dove k è il numero dei valori possibili) con componenti tutte nulle a parte una, in modo che valori diversi siano ortogonali.

Esempio 1.3. Se i valori possibili sono i tre colori *rosso* (R), *verde* (G), *blu* (B), si pone

$$R = (1,0,0), \quad G = (0,1,0), \quad B = (0,0,1). \tag{1.1}$$

Definizione 1.4. (Rumore, outliers). Il *rumore* è l'aggiunta di fattori esterni dovuta al processo di misura (e non alla legge soggiacente). Gli *outliers* sono dati che si collocano molto al di fuori, rispetto agli altri.

I *tasks* sono in genere di due tipologie (ma ce ne sono altre):

Definizione 1.5. (Supervised learning). Nel *supervised learning* sono dati dei *samples* di una funzione f ignota, nella forma

$$\langle input, output \rangle$$

si tratta di trovare una buona approssimazione di f. Gli input sono detti anche *variabili indipendenti*, gli ouput *variabili dipendenti* o *risposte*. Se f è a valori discreti, il problema si dice di *classificazione*. Se l'output è costituito da valori reali, allora si parla di *regressione*.

Definizione 1.6. (Unsupervised learning). Nell'*unsupervised learning* non si dispone di input e output nelle istanze, ma solo di dati non etichettati. Un problema tipico è quello di raggruppare tali dati secondo determinati criteri.

Noi ci occuperemo soprattutto di supervised learning.

Definizione 1.7. (Modello, ipotesi). Il modello cerca di descrivere la relazione tra i dati con un *linguaggio*, legato alla rappresentazione dei dati. Le *ipotesi* sono le funzioni h_w proposte dal modello per approssimare la "vera" funzione f. Le ipotesi sono indicizzate da parametri (w) e formano uno *spazio delle ipotesi* H.

In generale non esiste un modello *ottimo*: se un modello di apprendimento è il migliore in qualche problema, sarà peggiore di altri in altri problemi. Questo concetto è noto come *No Free Lunch Theorem*, ovvero "non c'è un pranzo gratis" (mah, sarà qualche detto inglese, ndr). In ogni caso, questo non significa che tutti i modelli siano equivalenti.

Definizione 1.8. (Algoritmo di apprendimento). Un *algoritmo* di apprendimento si occupa di cercare nello spazio delle ipotesi H (di un modello fissato) la migliore approssimazione della funzione f.

Come definiamo *buona approssimazione*? Si utilizza una *loss function* $L(h(x), d)$, che misura la distanza tra $h(x)$ e d, dove d è il valore osservato.

Definizione 1.9. (Errore). L'errore è definito da

$$E = \frac{1}{l}\sum_{i=1}^{l} L\big(h(x_i), d_i\big). \tag{1.2}$$

Esempio 1.10. Nei problemi di regressione spesso si usa

$$L(h(x), d) = \big(d - h(x)\big)^2$$

e in tal caso l'errore si dice *errore quadratico medio* (MSE).

Se siamo di fronte a un problema di classificazione, allora è più conveniente usare la *loss function* che vale 1 se i suoi due argomenti sono uguali (e dunque la classificazione è corretta) e 0 altrimenti.

Osservazione 1.11. In Machine Learning, quando si parla di *performance*, si fa riferimento all'accuratezza predittiva, non all'efficienza computazionale.

2 ESEMPI

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

2.1 LEZIONE DI DOMENICA 22 SETTEMBRE

2.1.1 Alcune indicazioni

Un paio di cose:

- Per creare paragrafi numerati, non va usato `\section`, ma `\subsection`;
- Non usate il grassetto, ma il *corsivo*;
- Usate `equation` per le equazioni, in modo che vengano tutte numerate;
- usare gli ambienti `definition`, `theorem`, `example`.

Definizione 2.1. (Tensore). Un *tensore* è ciò che ruota come un tensore.

$$e^z = \sum_{n=0}^{+\infty} \frac{z^n}{n!}. \tag{2.1}$$

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

```c
#include <stdio.h>
int main()
{
  // printf() displays the string inside quotation
  printf("Hello, World!");
```

```
  return 0;
}
```