Apprendimento Automatico: introduzione al PAC learning

WM&IR a.a. 2008/9

A. Moschitti, R. Basili

Dipartimento di Informatica Sistemi e produzione
Università di Roma "Tor Vergata"
basili@info.uniroma2.it

Sommario

- PAC-learning
  - Motivazioni ed esempi
  - Definizione
  - Applicazioni

Introduzione

- Il classificatore Bayesiano e gli alberi di decisione possono apprendere la differenza tra categorie/classi diverse.
- Vogliamo imparare da esempi la classe "*corporatura media*" delle persone :
  - Le *features* rilevanti sono (almeno): altezza e peso.
  - Il **training-set** (esempi di apprendimento) ha taglia *m*, cioè:
    - *m* persone di cui conosciamo
      - la **corporatura**,
      - **l'altezza**
      - E (!!) la **taglia**.

Motivazioni

Vogliamo derivare il valore *m* per imparare *bene* questo *concetto*.

Cosa significa "*bene*"?

- Derivare una funzione ipotesi *h* che sia il più possibile **coerente** con il concetto "corporatura media" (*CM*)
- *CM* è una funzione
- Idea: "*bene*" significa che la probabilità di errore è piccola, cioe'

$$p(h(x) \neq CM(x)) < \varepsilon.$$

Definizione di PAC Learning (1)

- Sia c la funzione che vogliamo imparare (il ***concetto***).
- Sia h il *concetto* appreso e x un individuo
- $error(h) = Prob\ (c(x) \neq h(x))$
- Vorremmo che:

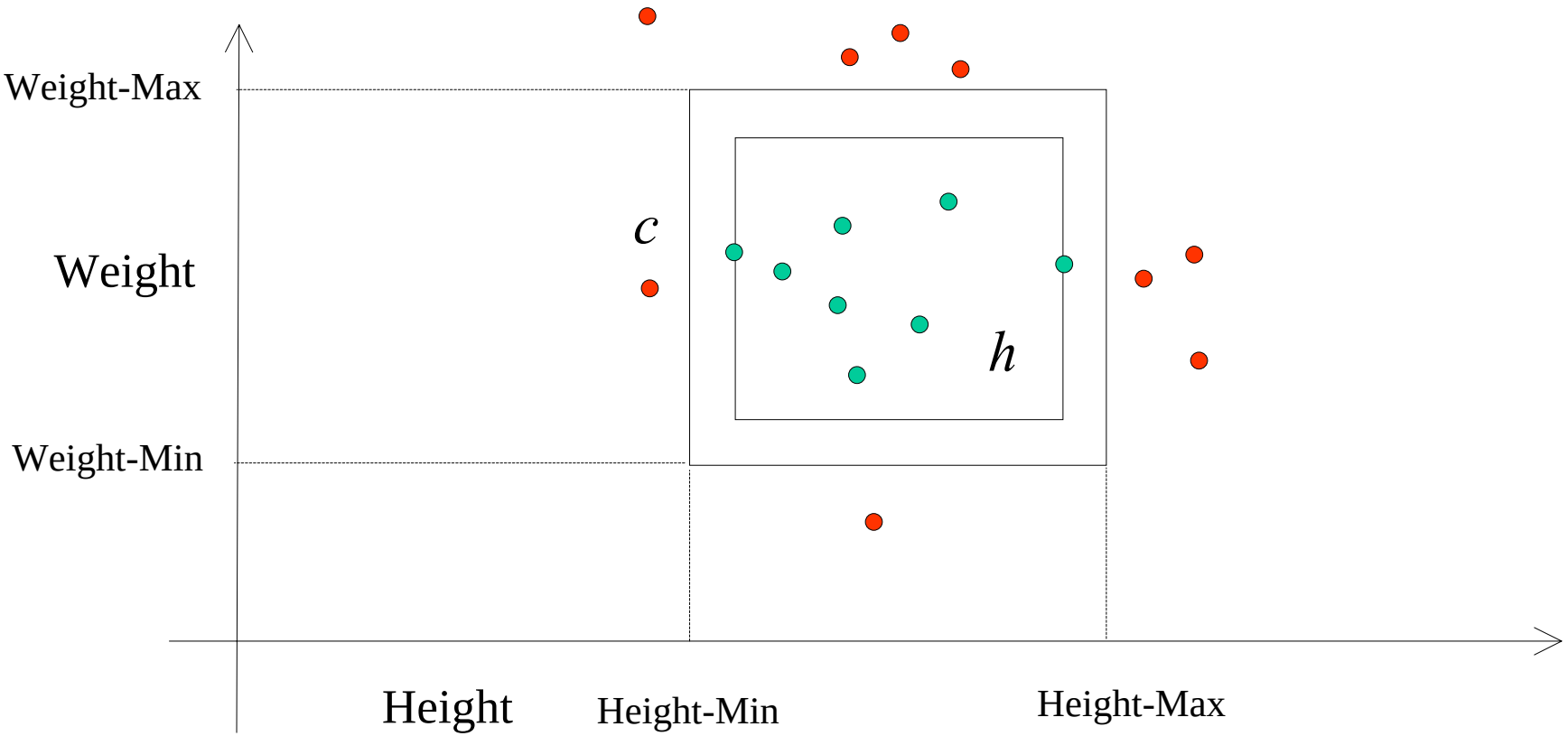
$$Pr(error(h) > \varepsilon\ ) < \delta$$

- L'apprendimento è *buono* se
  - dato l'obiettivo di mantenere l'errore minore di ε,
  - la probabilità di superarlo è *inferiore* a δ

Definizione di PAC Learning (2)

- Questo paradigma si definisce: *Probably Approximately Correct (PAC) Learning*
- L' apprendimento è tanto più buono quanto più ε e δ possono essere mantenuti piccoli
- Problema:
  - Fissati ε e δ, determinare la taglia m del training-set.
  - La proprietà imposta dal PAC potrebbe non dipendere dall'algoritmo di *learning* ma solo dallo spazio degli esempi

Esempio

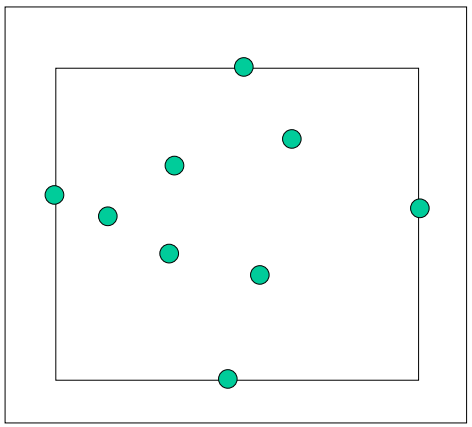

Weight-Max
Weight
Weight-Min
c
h
Height
Height-Min
Height-Max

La Classe di funzioni di apprendimento

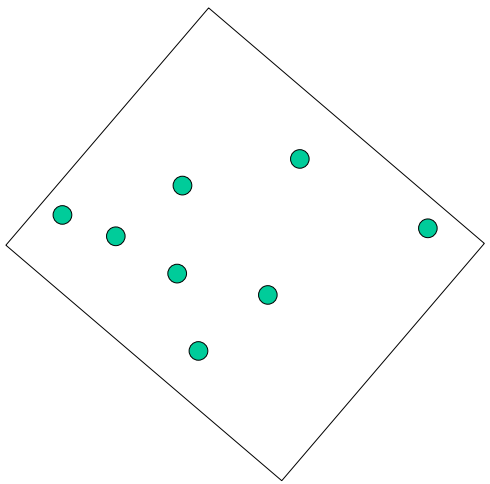
- Se non ci sono esempi di training positivi

$\Rightarrow$ Il concetto *C* appreso è nullo (cioè nessun esempio costituisce una istanza di *C*)

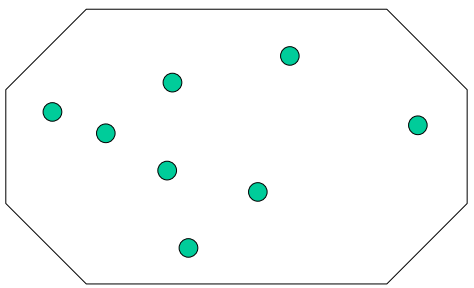
- Altrimenti, il concetto è il più piccolo rettangolo (parallelo agli assi) che contiene gli esempi postivi



Escludiamo altre possibili ipotesi



Escludiamo altre possibili ipotesi



Quanto è buona la nostra scelta?

- Un esempio x è classificato in modo sbagliato da $h(.)$ se cade tra i due rettangoli.
- Sia l' area compresa tra i due rettangoli pari a ε

$\Rightarrow$ l'errore (prob. di errore) di h è ε

  - Con quale assunzione?

Idea del PAC-Learning (1)

- Fissiamo un errore ε e δ vogliamo sapere quanti esempi di training m sono necessari per apprendere il *concetto*.
- Dobbiamo mettere un limite δ (bound) alla probabilità di apprendere una funzione h che abbia un errore $> \varepsilon$.
- Per fare questo calcoliamo la probabilità di scegliere un'ipotesi h che classifichi bene gli m esempi di training MA che commetta un errore superiore a ε.
  - Questa funzione rappresenta una ***cattiva ipotesi***

Idea del PAC-Learning (2)

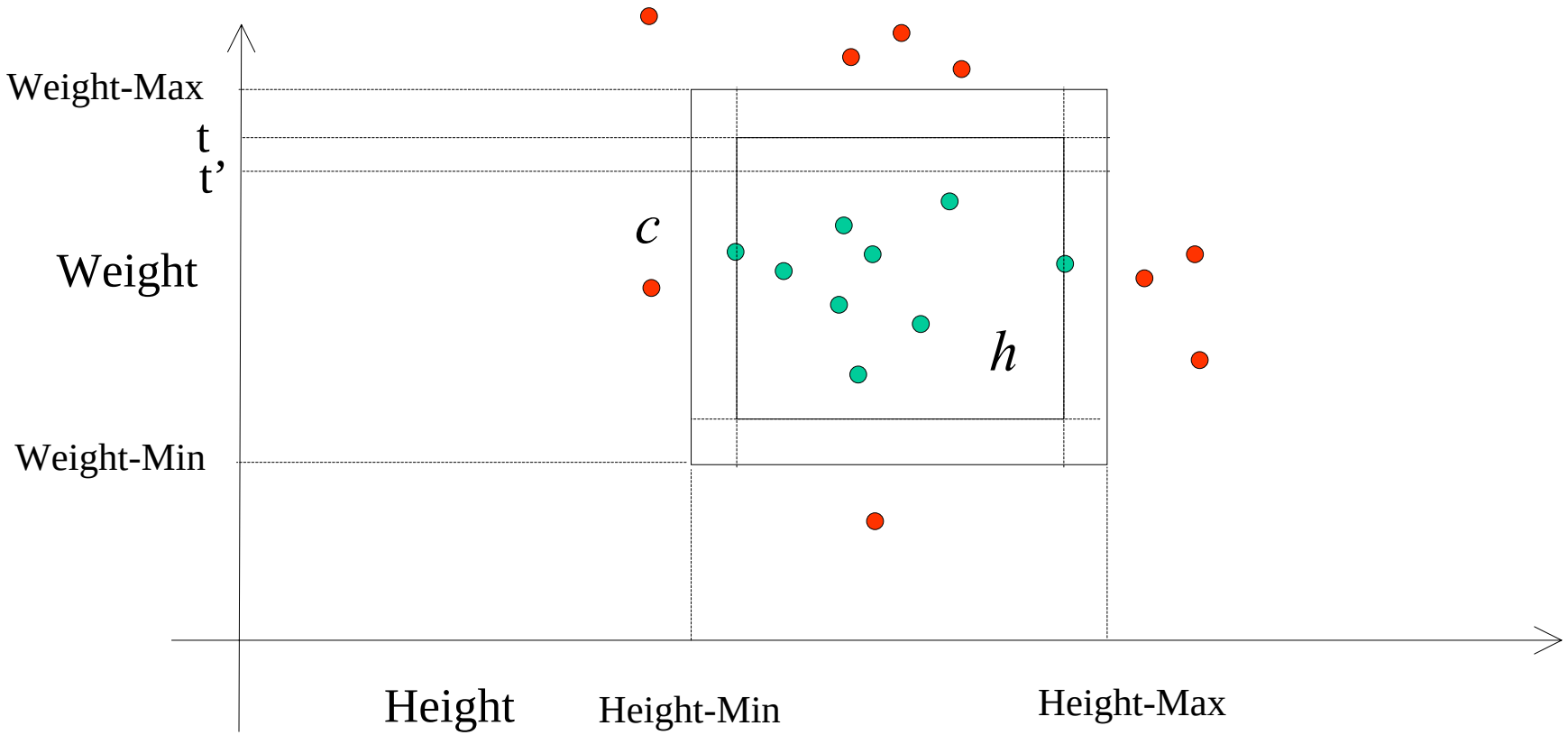
- Dato un esempio x,
  - se $P(h(x) \neq c(x)) > \varepsilon$ (cattiva ipotesi)
  - allora, $P(h(x) = c(x)) < 1-\varepsilon$
- Dato ε, m esempi di training cadono nel rettangolo h con probabilità $< (1-\varepsilon)^m$
- La probabilità di scegliere un'ipotesi h qualsiasi consistente con il training è $(1-\varepsilon)^m \cdot N$
  - Dove N è il numero di possibili ipotesi che hanno errore $> \varepsilon$.

Calcolo dell upper-bound

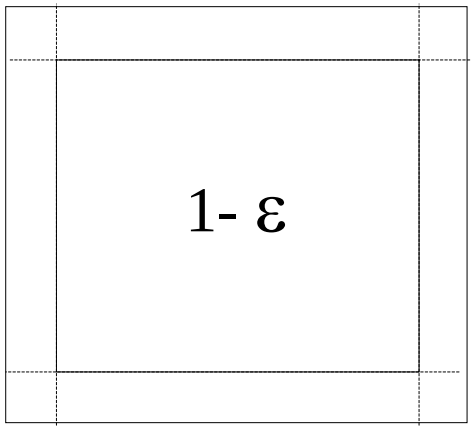
- Allora dovremmo imporre che
  $N \cdot (1-\varepsilon)^m < \delta$
  - Imponiamo quindi un limite δ alla probabilità di apprendere una h che abbia un errore $> \varepsilon$ con m esempi di training
- Problema: Non conosciamo N

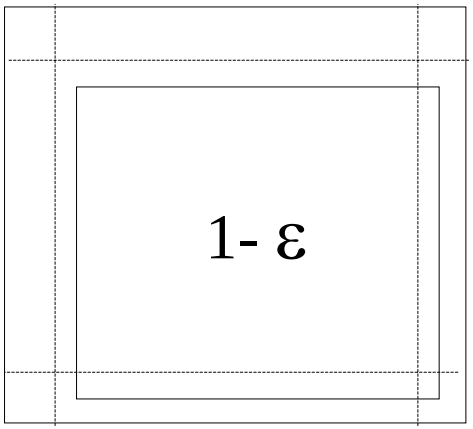
  $\Rightarrow$ dobbiamo trovare un bound che tiene conto del numero di ipotesi *cattive*.
- Dividiamo il rettangolo in quattro zone di area $\varepsilon/4$

Figura dell'esempio

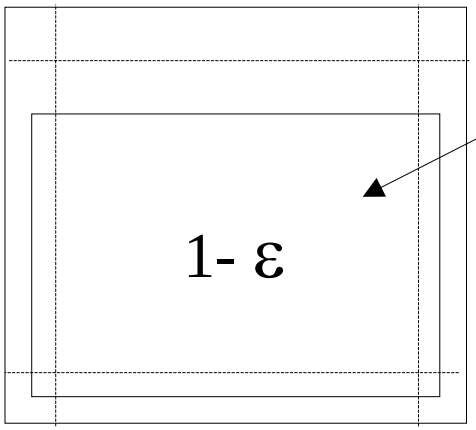


h cattiva non interseca più di tre strisce alla volta





Ipotesi cattive con errore > ε sono contenute in quelle che hanno errore = ε



Riesco a intersecare 3 lati. Ho aumentato la lunghezza di *h* ma ho dovuto diminuire l'altezza per avere sempre un'area ≤ 1- ε

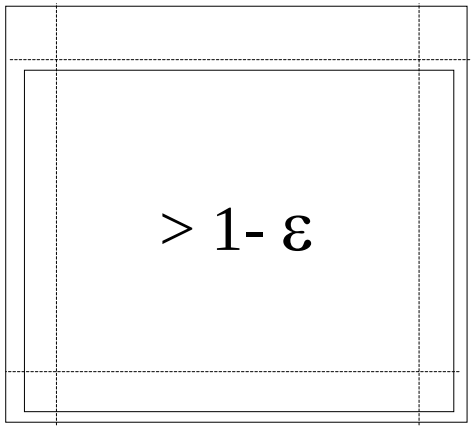
Calcolo dell upper-bound (2)

- Un'ipotesi *h* (*cattiva*) ha un errore > ε ⇒ ha un'area < 1- ε
- Un rettangolo di area < 1- ε non può intersecare le 4 strisce contemporaneamente ⇒ se *m* punti occupano tutte le strisce non possono appartenere tutti ad una *h* cattiva.
- Pertanto una condizione necessaria per avere *h* cattiva è che tutti gli *m* esempi siano fuori da almeno una delle 4 strisce.
- In altre parole, quando *m* punti sono tutti fuori di una (delle 4) strisce allora *h* può essere cattiva.

  ⇒ la probabilità di questo evento (*fuori da almeno una striscia*) è > della probabilità di avere una *cattiva h*.

Punto di vista logico

- Ipotesi cattiva $\Rightarrow$ punti fuori da almeno una delle strisce
- OSS: il viceversa non è vero
  - (fuori dal almeno una striscia $\Rightarrow$ cattiva)



$A \Rightarrow B$

P(A) < P(B)

P(*ipot. catt.*) < P(*fuori da una delle 4 strisce*)

Un bound migliore (3)

- $P(x \text{ fuori dalla striscia } t) = (1 - \varepsilon/4)$
- $P(m \text{ esempi fuori dalla striscia}) = (1 - \varepsilon/4)^m$
- $P(m \text{ esempi fuori da almeno una striscia}) <$
  $4 \cdot (1 - \varepsilon/4)^m$

$\Rightarrow P(\text{errore}(h) > \varepsilon) < 4 \cdot (1 - \varepsilon/4)^m$

Un bound migliore (3)

- L'upperbound dell'errore lo impongo minore di δ
  - $4 \cdot (1 - \varepsilon/4)^m < \delta$
  - $m > ln(\delta/4)/ln(1 - \varepsilon/4)$
- $-ln(1-y) = y + y^2/2 + y^3/3 + \ldots$

$\Rightarrow (1-y) < e^{(-y)}$

- Da $m > ln(\delta/4)/ln(1 - \varepsilon/4)$

$\Rightarrow$ $\boxed{m > (4/\varepsilon) \cdot ln(4/\delta)}$

Esempi numerici

ε	δ	m
0.1	0.1	148
0.1	0.01	240
0.1	0.001	332
0.01	0.1	1476
0.01	0.01	2397
0.01	0.001	3318
0.001	0.1	14756
0.001	0.01	23966
0.001	0.001	33176

Definizione Formale del PAC-Learning

- Sia f la funzione da apprendere, $f: X \rightarrow I$, $f \in F$
- D è la distribuzione di probabilità su X
  - *Con cui si creano il training and test test*
- $h \in H$,
  - *h è la funzione appresa e H l'insieme delle ipotesi*
- m è la taglia del training-set
- $error(h) = Prob\,[f(x) \neq h(x)]$
- *F e' una classe di funzioni PAC learnable, SSE esiste un* ***algoritmo di learning*** *che per ogni f, per tutte le distribuzioni D su X e per ogni* $\varepsilon > 0$, $\delta < 1$, ***produce h :***

$$P(error(h) > \varepsilon) < \delta$$

Lower Bound sulla taglia del training-set

- Riconsideriamo il primo bound che abbiamo trovato:
  - h è una *cattiva* funzione di learning: $error(h) > \varepsilon$
  - $P(f(x)=h(x))$ per m esempi è al massimo $(1-\varepsilon)^m$
  - Moltiplicando per il numero di ipotesi cattive si ha la probabilità di scegliere un'ipotesi cattiva e consistente
  - $P(\text{cattiva e consistente}) < N \cdot (1-\varepsilon)^m < \delta$
  - $P(\text{cattiva e consistente}) < N \cdot (e^{-\varepsilon})^m = N \cdot e^{-\varepsilon m} < \delta$

$\Rightarrow m > (1/\varepsilon)\,(ln(1/\delta)+ln(N))$ è un lower bound su m in generale

Esempio

- Supponiamo di volere apprendere una funzione booleana di n variabili
- Abbiamo al massimo 2^{2^n} funzioni differenti

$$\Rightarrow m > (1/\varepsilon)\,(ln(1/\delta)+ln(2^{2^n}))=$$

$$= (1/\varepsilon)\,(ln(1/\delta)+2^n ln(2))$$

Valori Numerici

n	epsilon	delta	m
5	0.1	0.1	245
5	0.1	0.01	268
5	0.01	0.1	2450
5	0.01	0.01	2680
10	0.1	0.1	7123
10	0.1	0.01	7146
10	0.01	0.1	71230
10	0.01	0.01	71460

Riferimenti

- PAC-learning:
  - Da pagina 552 a 555: Libro di Intelligenza Artificiale:

    (Artificial Intelligence: a modern approach)
  - http://www.cis.temple.edu/~ingargio/cis587/readings/pac.html
  - Goldman Report, http://www.learningtheory.org/articles/COLTSurveyArticle.ps
  - Machine Learning, Tom Mitchell, McGraw-Hill.