VC-dimension

A. Moschitti, R. Basili

Dipartimento di Informatica Sistemi e produzione
Università di Roma “Tor Vergata”
Email: basili@info.uniroma2.it

Sommario

- Computational Learning theory
  - PAC learning
  - VC-dimension

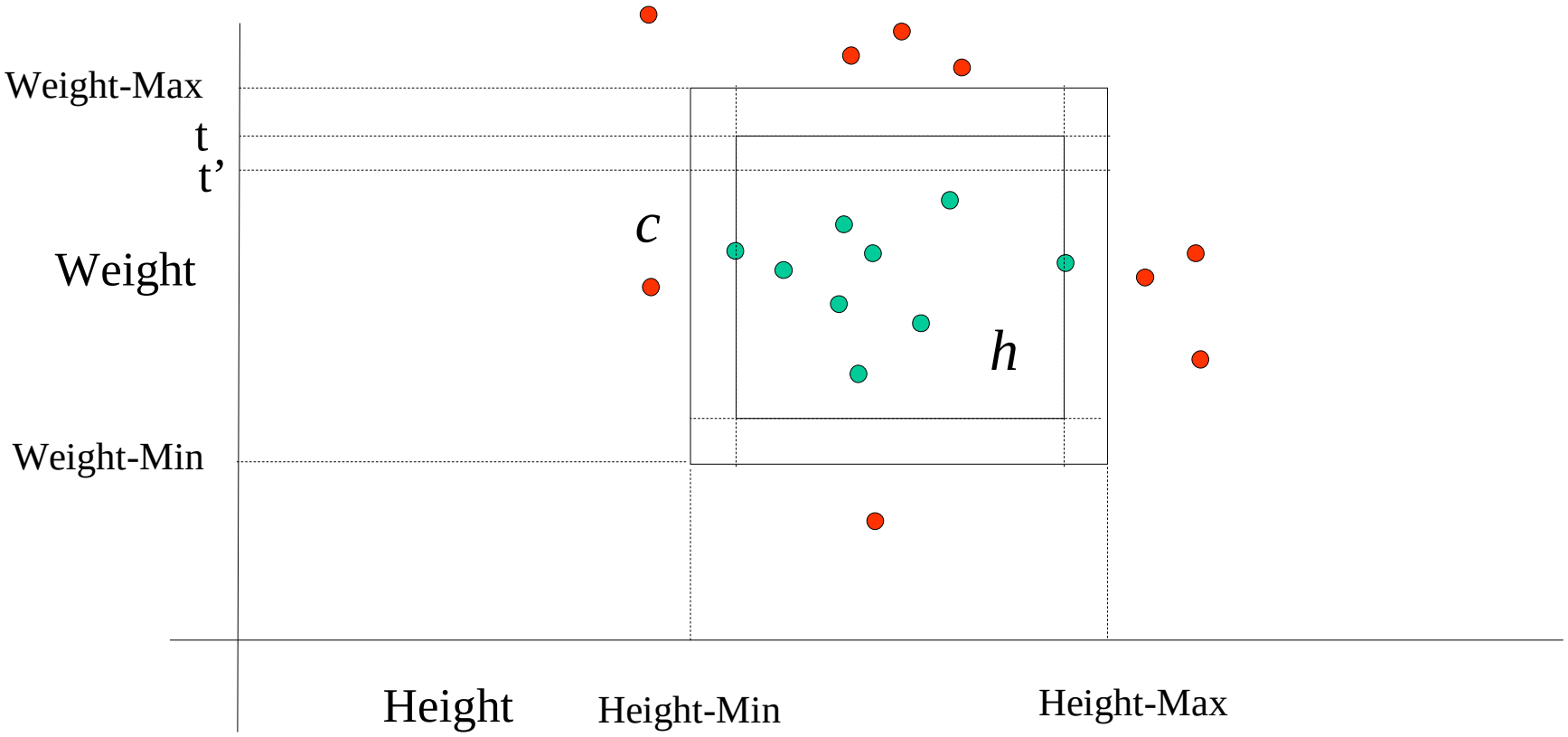
PAC-Learning

- 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*) < > *h*(*x*)]
- *F e' una classe di funzioni PAC apprendibile, se esiste un* ***algoritmo di learning*** *che per ogni f, per tutte le distribuzioni D su X e per ogni* $0<\varepsilon, \delta<1$, ***produce h :*** $P(error(h)> \varepsilon)< \delta$

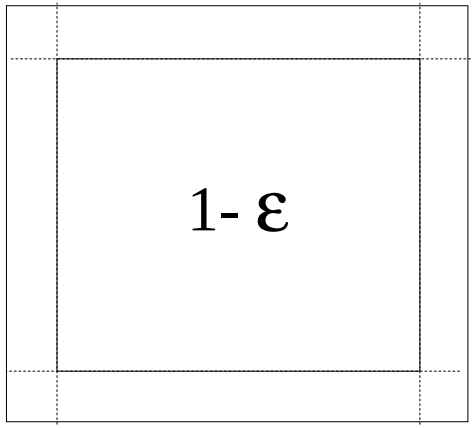
Bound per le ipotesi “rettagolari” al concetto corporatura media

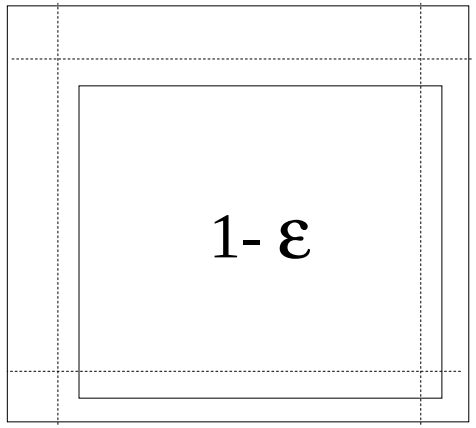
- Fissiamo un errore ε e δ vogliamo sapere quanti esempi di training *m* sono necessari per apprendere il *concetto*.
- Dobbiamo mettere un limite δ (bound) alla probabilità di apprendere una funzione *h* che ha un errore > ε.
- Per fare questo calcoliamo la probabilità di scegliere un’ipotesi *h* che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

Figura dell'esempio

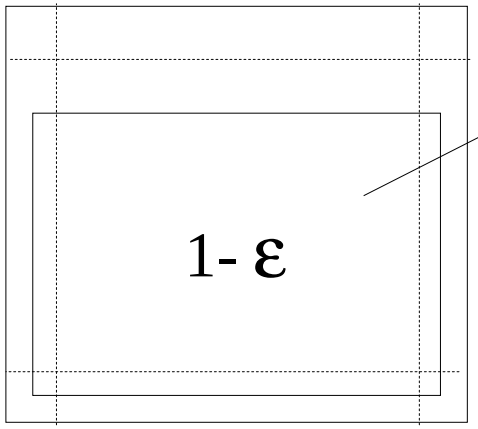


h cattiva non interseca più di tre strisce alla volta





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

Dimostrazione

- 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) striscia *h* può essere cattiva.

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

Dimostrazione (cont.)

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

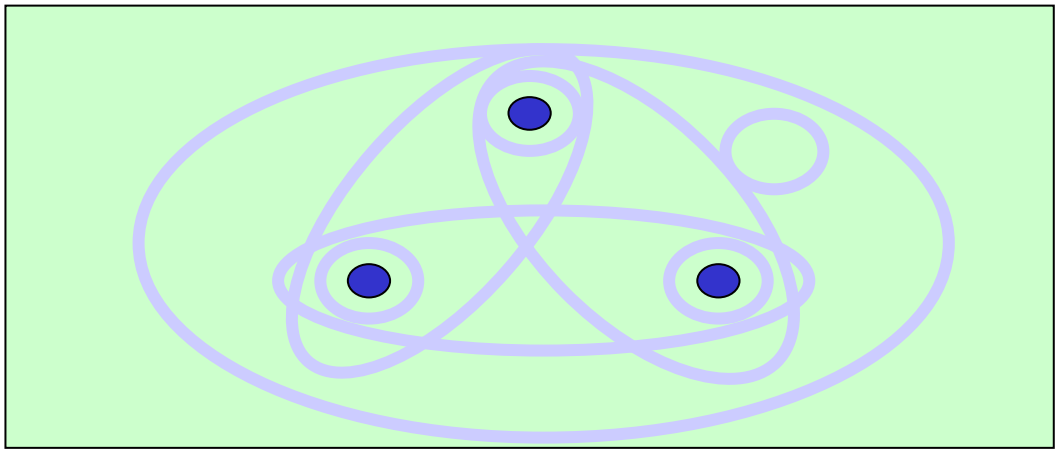
$\Rightarrow P(\text{errore}(h) > \varepsilon) < 4 \cdot (1 - \varepsilon/4)^m$

Computational Learning Theory

- Abbiamo risolto solo un caso particolare di learning:
  - Corporatura media
  - Non è stata derivata nessuna regola generale
- Esiste un modo sistematico per decidere se una funzione è PAC apprendibile e determinare il bound?
- La risposta è affermativa ed è basata sul concetto di Vapnik-Chervonenkis dimension (VC-dimension, [Vapnik 95])

Definizione della VC-Dimension (1)

- Def.1: (**frantumazione di un insieme**): Un sotto-insieme S di istanze di uno spazio X è frantumato da una famiglia di funzioni F se $\forall S' \subseteq S$ esiste una funzione $f \in F$ tale che: $f(x) = \begin{cases} 1 & x \in S' \\ 0 & x \in S - S' \end{cases}$



Definizione della VC-Dimension (2)

- Def. 2: (**VC-dimension**): La VC-dimension di un insieme di funzioni *F* (*VC-dim*(*F*)) è la cardinalità dell'insieme più grande frantumabile da *F*.
- *Oss:* la tipologia delle *f* usate per frantumare l'insieme determina la *VC-dim*

VC-Dim di superfici lineari (iperpiani)

- Nel piano:
  - VC(H) è almeno 3
  - VC(H) < 4 perché non esiste nessun insieme di 4 punti che può essere frantumato da una retta.

$\Rightarrow$ VC(H)=3, in generale per uno spazio a k dimensioni VC(H)=k+1

- NB: Se prendo punti linearmente dipendenti non posso sperare di frantumarli, quindi in genere non li considero

Connessione tra VC-dimension e l'errore (1)

- *Teorema 1*: Siano H e F classi di funzioni tale che $F \subseteq H$. Sia A un algoritmo che, dati m esempi di training restituisce una ipotesi $h \in H$ consistente con il training. Allora esiste una costante c_0 tale che per ogni funzione target $f \in F$, per ogni distribuzione sottostante D e per ogni $0 < \varepsilon$, $\delta < 1$, ***se***

$$m > \frac{c_0}{\varepsilon} \cdot \left(vc - dim(H) \cdot \log \frac{1}{\varepsilon} + \log \frac{1}{\delta} \right)$$

- ***allora*** con probabilità $(1-\delta)$ $\quad\quad$ $\mathrm{err}_{D,f}(h) \leq \varepsilon$

Connessione tra VC-dimension e l'errore (2)

- *Teorema 2*: Ogni algoritmo di apprendimento PAC per una classe di concetti *F* tale che *d=VC-dim(H)* richiede $m = O(1/\varepsilon (\log(1/\delta) + d)$ esempi.

Esempio 1: I rettangoli hanno VC-dim > 4

- Dobbiamo scegliere un insieme di 4 punti e far vedere che può essere frantumato in tutti i modi possibili
- Scelti 4 punti, gli etichettiamo in tutti i modi possibili.
- Per ogni etichettamento deve esistere un rettangolo che induce tale assegnamento

Esempio 1: I rettangoli hanno VC-dim > 4

- Scegliamo 4 punti non allineati.

a) Se assegniamo tutti “+” prendi il rettangolo definito dai 4 punti

b) Se ass. tutti “-” prendi il rettangolo vuoto

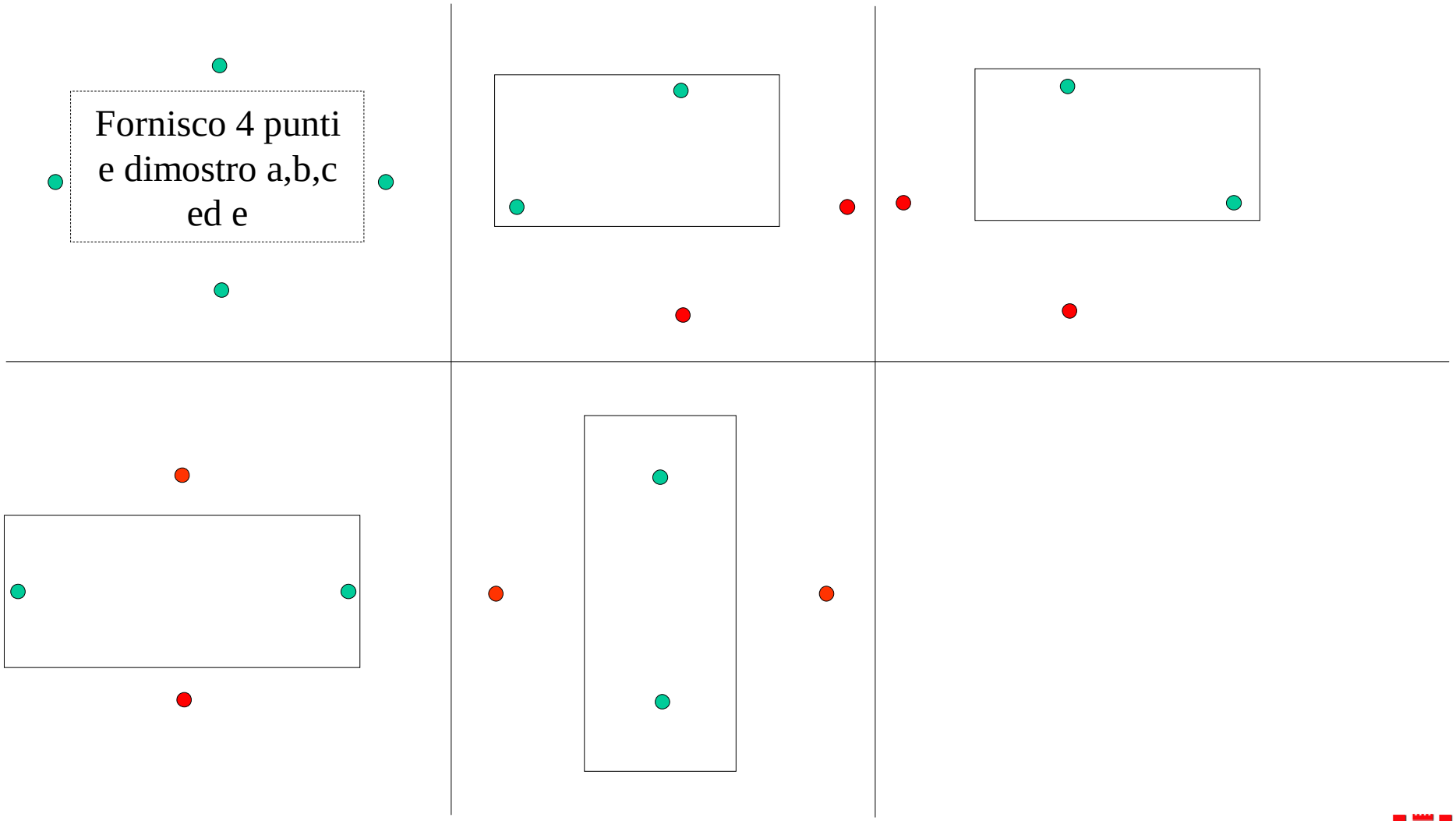
c) 3 punti “-” ed uno “+” prendi il rettangolo centrato solo sul punto positivo

d) 3 punti “+” ed uno “–” si può definire un rettangolo che esclude un punto

e) 2 punti “+” e 2 “–” si può definire un rettangolo che include 2 punti e ne esclude 2.

- Alfine di dimostrare d) and e) è necessario elencare tutte le possibilità

Esempio: dimostrazione di e)



La *VC-dim* non può essere 5

- Per qualsiasi insieme di 5 punti posso sempre definire un rettangolo che ha come vertici i punti più esterni
- Se assegno a tali vertici esempi positivi e al punto interno il valore negativo … non riuscirò a trovare nessun rettangolo in grado di separarmi questo assegnamento

Compariamo i bound

- $m > (4/\varepsilon) \cdot ln(4/\delta)$ (ad hoc bound)
- $m > (1/\varepsilon) \cdot ln(1/\delta) + 4/\varepsilon =$ (basato sullaVC-dim)

$(4/\varepsilon) \cdot ln(4/\delta) > (1/\varepsilon) \cdot ln(1/\delta) + 4/\varepsilon$

$4 \cdot ln(4/\delta) > ln(1/\delta) + 4$

$ln(4/\delta) > ln((1/\delta)^{1/4}) + 1$

$4/\delta > (1/\delta)^{1/4} \cdot e$

$4 > \delta^{3/4} \cdot e$

$4 > (<1) \cdot (<3)$ **verificata**

Riferimenti

- *A tutorial on Support Vector Machines for Pattern Recognition*
  - **Articolo scaricabile dalla rete**
- *The Vapnik-Chervonenkis Dimension and the Learning Capability of Neural Nets*
  - **Presentazione scaricabile dalla rete**
- Conputational Learning Theory

  (Sally A Goldman Washington University St. Louis Missouri)
  - **Scaricabile dalla rete**
- *AN INTRODUCTION TO SUPPORT VECTOR MACHINES*

  (and other kernel-based learning methods)

  N. Cristianini and J. Shawe-Taylor Cambridge University Press
  - **Esaustivo ma deve essere acquistato**