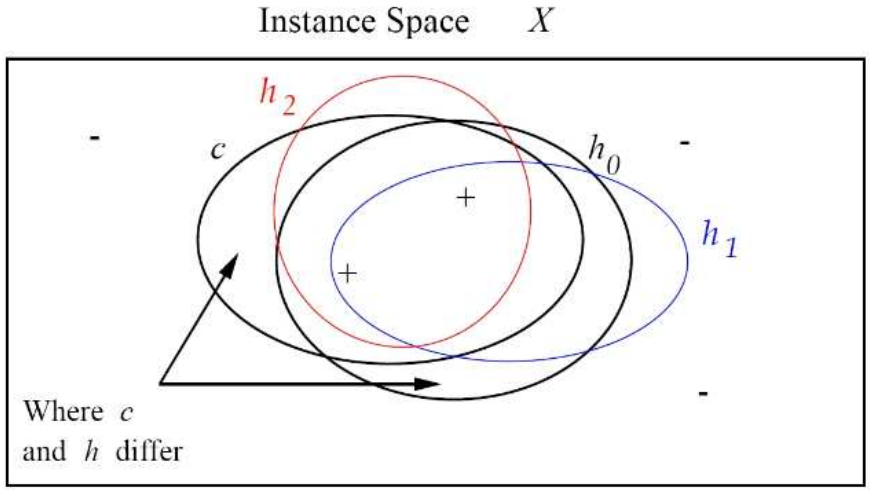
Errore Empirico ed Errore Ideale



Errore Ideale:
probabilità che h classifichi erroneamente un input selezionato dallo spazio delle istanze

(secondo la distribuzione di probabilità di occorrenza dell'input)

Errore Empirico:
(frazione del) numero di esempi classificati erroneamente da h

Overfitting

errore empirico(h_1) < errore empirico(h_2)

ma

errore ideale(h_1) > errore ideale(h_2)

Problema dell'overfitting

pochi dati

tante ipotesi con stesso errore empirico ...
... ma errore ideale maggiore o minore

Quale ipotesi scegliere ?

Problema dell'underfitting

poche ipotesi

↓

nessuna ipotesi "spiega" bene i dati

↓

errore empirico alto!

Come rimediare ?

Soluzione

utilizzare uno spazio delle ipotesi che non sia

- né troppo semplice (underfitting)
- né troppo complesso (overfitting)

Occorre "misurare" la complessità dello spazio delle ipotesi

non è facile!!!

Occorre **misurare** la complessità dello spazio delle ipotesi

Bound sull'Errore Ideale

VC dimension
misura di complessità di H

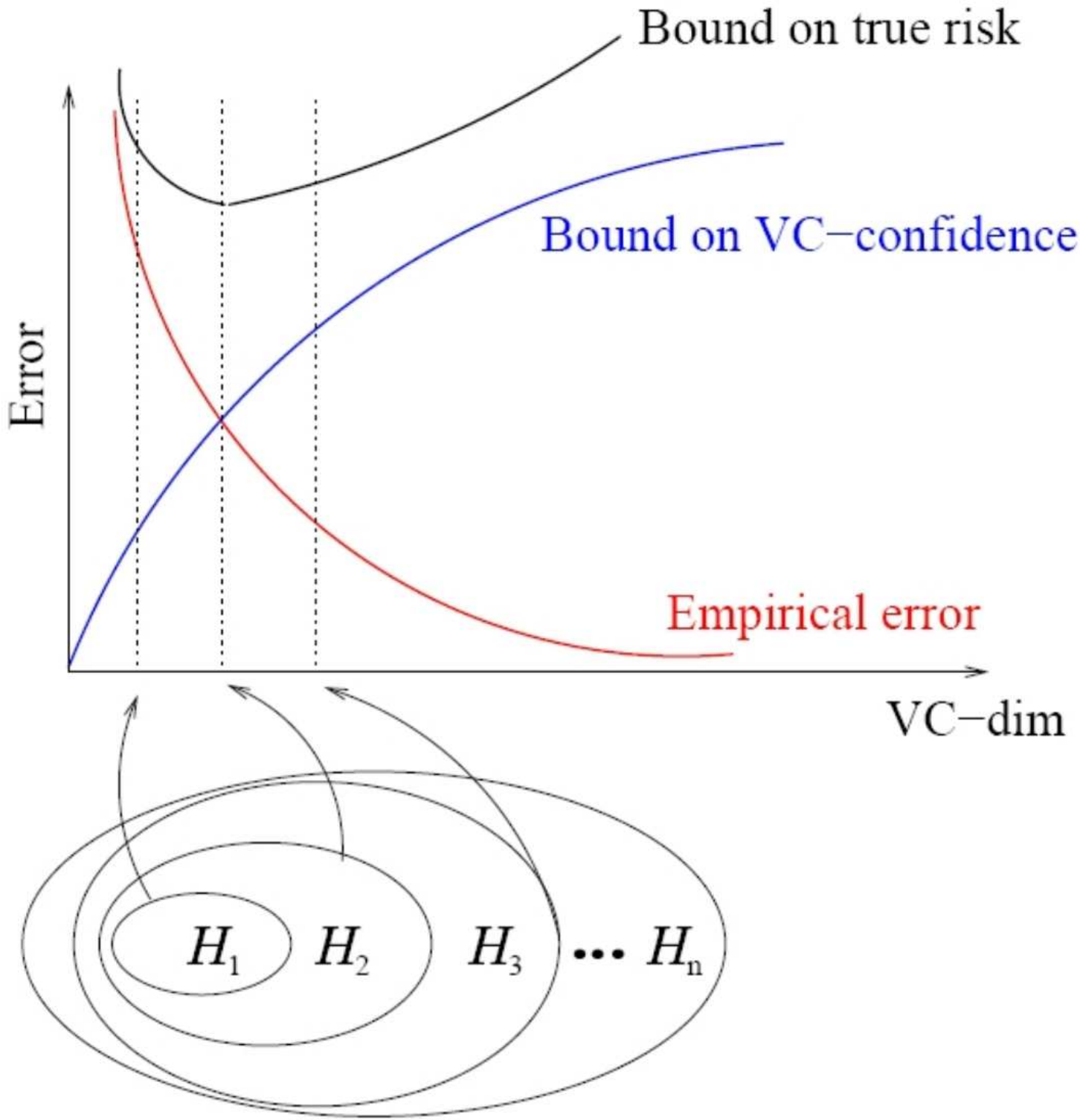
errore ideale(h) < errore empirico(h) + e(N, VC(H), d)

numero di esempi di apprendimento

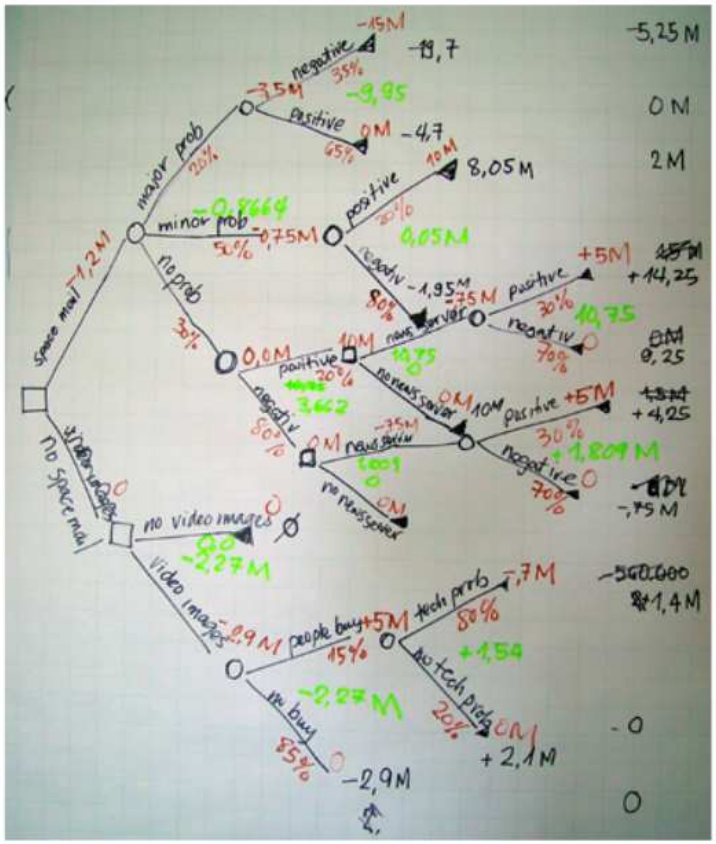
il bound vale con probabilità 1-d

e(N, VC(H), d) è

- inversamente proporzionale a N
- direttamente proporzionale a VC(H)


Bound on true risk
Bound on VC−confidence
Empirical error
Error
VC−dim
H_1
H_2
H_3
H_n

Decision Trees



- istanze rappresentate da coppie attributo-valore
- classificazione multiclasse
- esempi di apprendimento possono contenere errori e/o valori mancanti

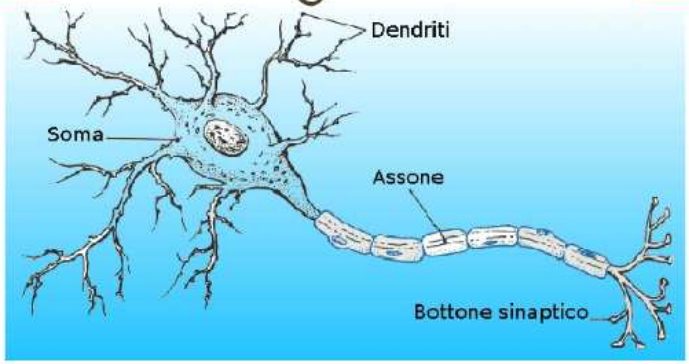
- nodo: test su attributo
- ramo: corrisponde ad un possibile valore dell'attributo
- foglia: classificazione

Algoritmo di apprendimento

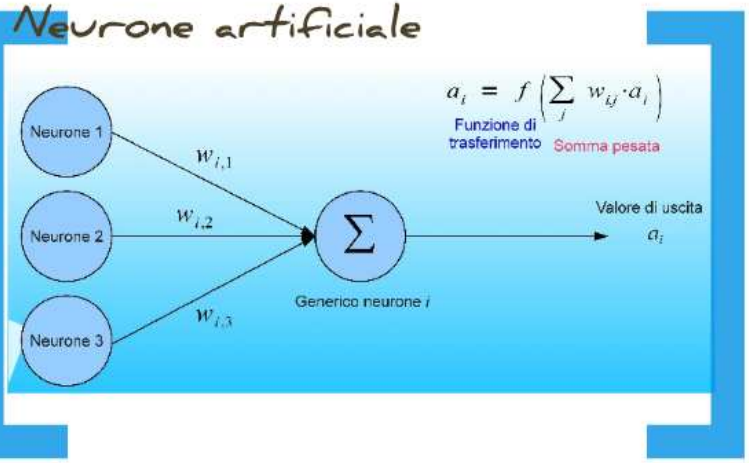

guadagno entropico

Neural Networks

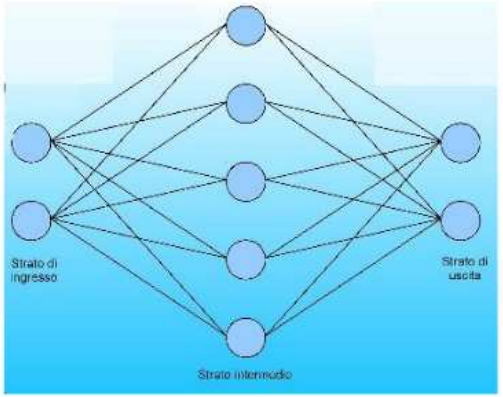
Neurone biologico



Neurone artificiale

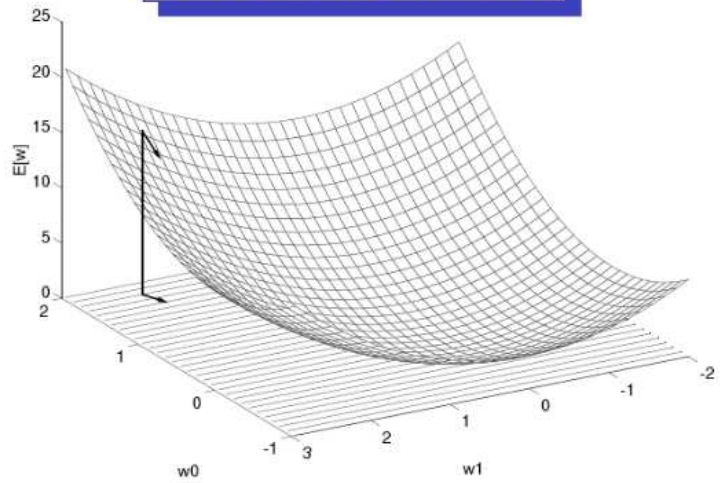


- si ispirano al cervello umano
- sia classificazione che regressione
- sia supervised che unsupervised
- adatte ad approssimare funzioni reali continue



Apprendimento

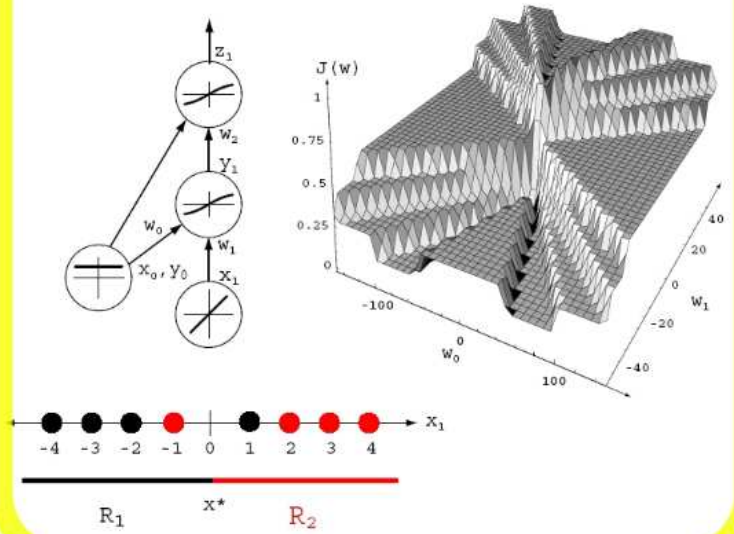
Discesa di Gradiente



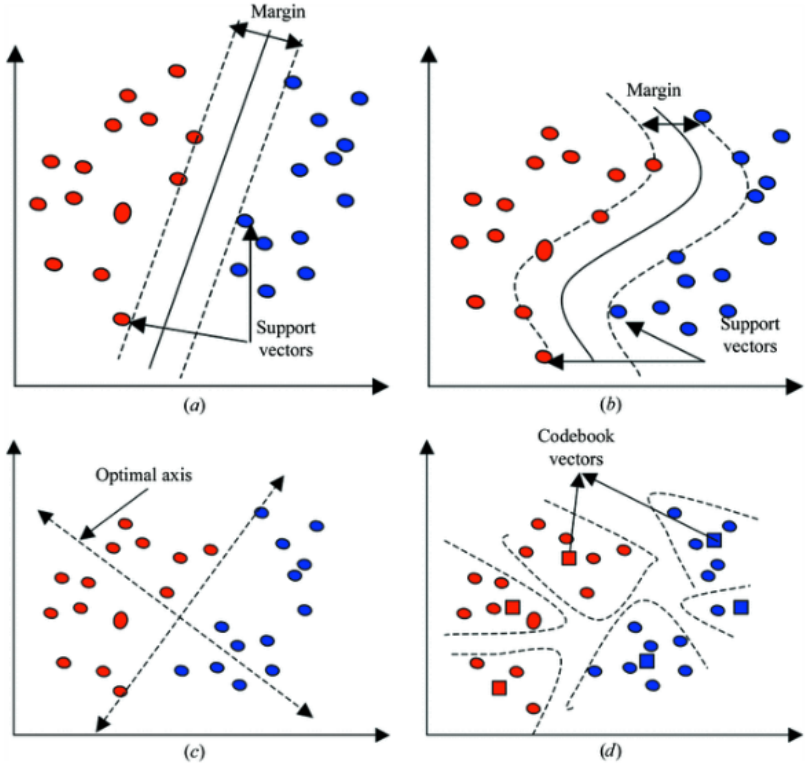
Idea base: partire da un $\vec{w}$ random e modifi carlo nella direzione contraria al gradiente (che indica la direzione di crescita di $E[\vec{w}]$)

$$\underbrace{\nabla E[\vec{w}]}_{\text{gradiente}} \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n}\right], \quad \Delta\vec{w} = -\eta\nabla E[\vec{w}], \quad \Delta w_i = -\eta\frac{\partial E}{\partial w_i}$$

Esempio di Funzione Errore



Kernel Methods



- sia supervised che unsupervised
- sfruttano le funzioni kernel
- possono trattare direttamente dati strutturati
- fra le migliori tecniche di apprendimento automatico

Apprendimento di Support Vector Machines

l'apprendimento consiste nel risolvere un problema di ottimizzazione vincolata

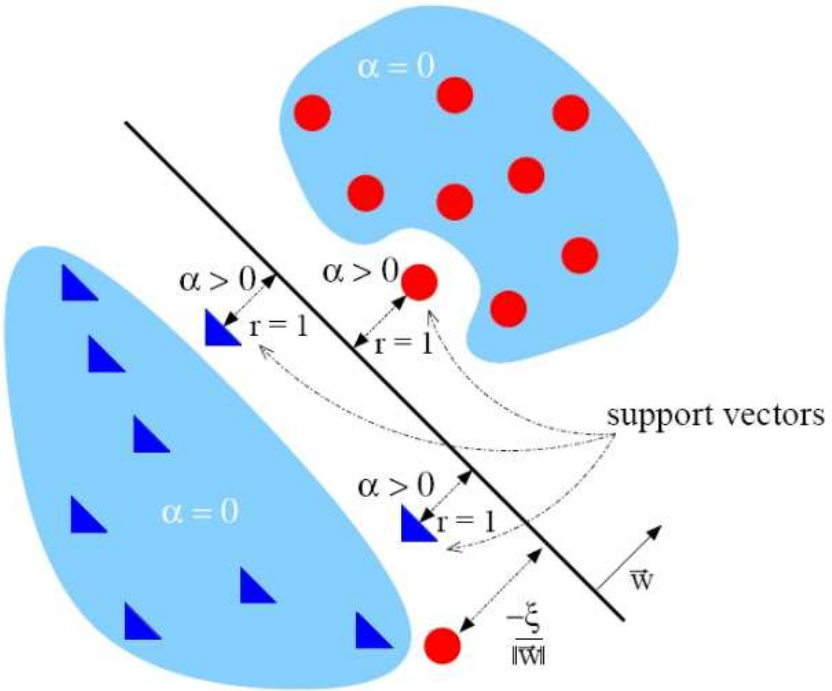
formulazione duale

$\max_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j)$

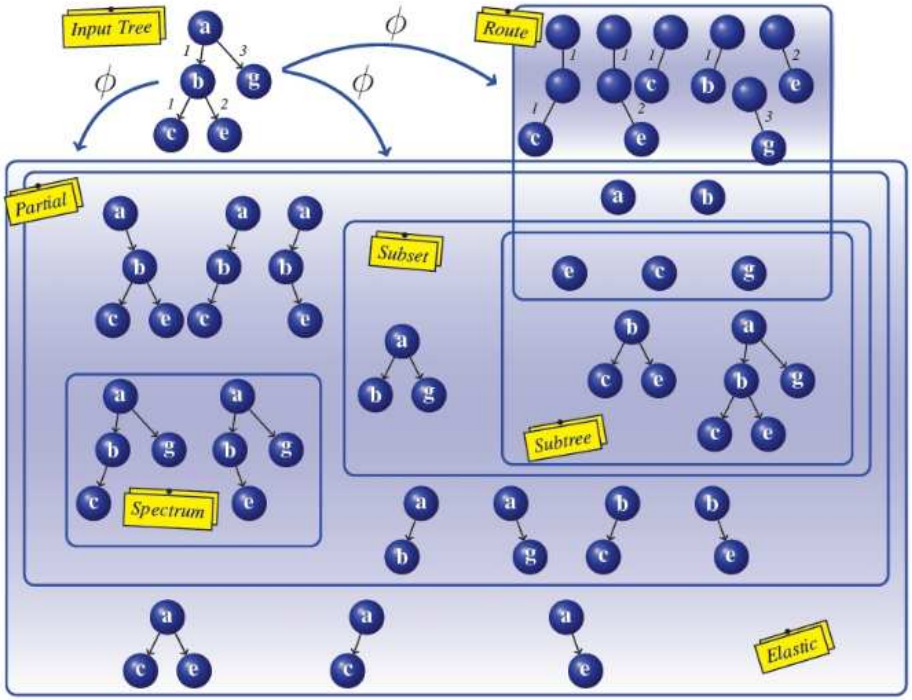
soggetto a: $\forall i \in \{1, \ldots, n\} : 0 \leq \alpha_i \leq C$ e $\sum_{i=1}^{n} y_i \alpha_i = 0.$

K(x,y) funzione kernel:

- calcola il grado di "similarita" fra x e y
- si può definire anche su input strutturati



$$K(albero_1, albero_2) = Phi(albero_1) \cdot Phi(albero_2)$$



Vedremo anche...

- PAC Learning
- Boosting
- Apprendimento Probabilistico (bayesiano)
- Algoritmi "semplici" di clustering

e non finisce qui...