

Richiamo di Concetti di Apprendimento Automatico ed altre nozioni aggiuntive

Libro di riferimento: T. Mitchell

Ingredienti Fondamentali Apprendimento Automatico

- Dati di Allenamento (estratti dallo Spazio delle Istanze, X)
- Spazio delle Ipotesi, \mathcal{H}
 - costituisce l'insieme delle funzioni che possono essere realizzate dal sistema di apprendimento;
 - si assume che la funzione da apprendere f possa essere rappresentata da una ipotesi $h \in \mathcal{H} \dots$ (**selezione di h attraverso i dati di apprendimento**)
 - o che almeno una ipotesi $h \in \mathcal{H}$ sia simile a f (approssimazione);
- Algoritmo di Ricerca nello Spazio delle Ipotesi, **alg. di apprendimento**

ATTENZIONE: \mathcal{H} non può coincidere con l'insieme di tutte le funzioni possibili e la ricerca essere esaustiva \rightarrow **Apprendimento è inutile!!!**

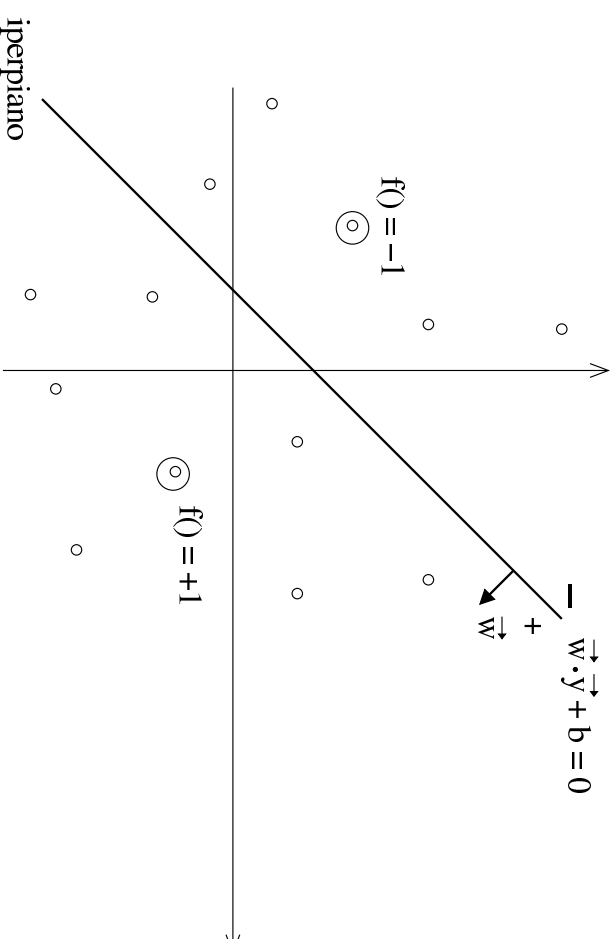
Si parla di **Bias Induttivo**: sulla rappresentazione (\mathcal{H}) e/o sulla ricerca (**alg. di apprendimento**)

Spazio delle Ipotesi: Esempio 1

Iperpiani in \mathbb{R}^2

- Spazio delle Istanze \rightarrow punti nel piano: $X = \{\vec{y} \in \mathbb{R}^2\}$
- Spazio delle Ipotesi \rightarrow dicotomie indotte da iperpiani in \mathbb{R}^2 :

$$\mathcal{H} = \{f_{(\vec{w}, b)}(\vec{y}) \mid f_{(\vec{w}, b)}(\vec{y}) = \text{sign}(\vec{w} \cdot \vec{y} + b), \vec{w} \in \mathbb{R}^2, b \in \mathbb{R}\}$$

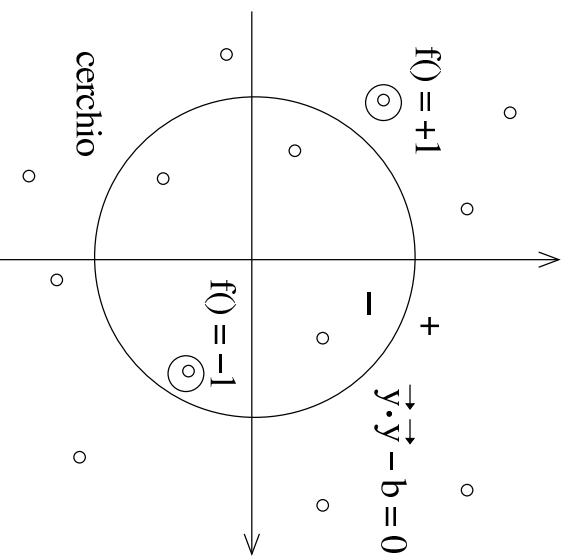


Spazio delle Ipotesi: Esempio 2

Dischi in \mathbb{R}^2

- Spazio delle Istanze \rightarrow punti nel piano: $X = \{\vec{y} \in \mathbb{R}^2\}$
- Spazio delle Ipotesi \rightarrow dicotomie indotte da dischi in \mathbb{R}^2 centrati nell'origine:

$$\mathcal{H} = \{f_b(\vec{y}) \mid f_b(\vec{y}) = \text{sign}(\vec{y} \cdot \vec{y} - b), b \in \mathbb{R}\}$$



Spazio delle Ipotesi: Esempio 3

Congiunzione di m letterali positivi

- Spazio delle Istanze \rightarrow stringhe di m bit: $X = \{s | s \in \{0, 1\}^m\}$
- Spazio delle Ipotesi \rightarrow tutte le sentenze logiche che riguardano i letterali positivi l_1, \dots, l_m (l_1 è vero se il primo bit vale 1, l_2 è vero se il secondo bit vale 1, etc.) e che contengono solo l'operatore \wedge (**and**):

$$\mathcal{H} = \{f_{\{i_1, \dots, i_j\}}(s) | f_{\{i_1, \dots, i_j\}}(s) \equiv l_{i_1} \wedge l_{i_2} \wedge \dots \wedge l_{i_j}, \{i_1, \dots, i_j\} \subseteq \{1, \dots, m\}\}$$

Es. $m = 3$, $X = \{0, 1\}^3$

Esempi di istanze $\rightarrow s_1 = 101$, $s_2 = 001$, $s_3 = 100$, $s_4 = 111$

Esempi di ipotesi $\rightarrow h_1 \equiv l_2$, $h_2 \equiv l_1 \wedge l_2$, $h_3 \equiv true$, $h_4 \equiv l_1 \wedge l_3$, $h_5 \equiv l_1 \wedge l_2 \wedge l_3$

Notare che: h_1 , h_2 , e h_5 sono false per s_1 , s_2 e s_3 e vere per s_4 ; h_3 è vera per ogni istanza; h_4 è vera per s_1 e s_4 ma falsa per s_2 e s_3

Principali Paradigmi di Apprendimento: Richiamo

Apprendimento Supervisionato:

- dato in insieme di esempi pre-classificati, $T^r = \{(x^{(i)}, f(x^{(i)}))\}$, apprendere una descrizione generale che incapsula l'informazione contenuta negli esempi (regole valide su tutto il dominio di ingresso)
- tale descrizione deve poter essere usata in modo predittivo (dato un nuovo ingresso \tilde{x} predire l'output associato $f(\tilde{x})$)
- si assume che un esperto (o maestro) ci fornisca la supervisione (cioè i valori della $f()$ per le istanze x dell'insieme di apprendimento)

Find-S è un algoritmo di apprendimento supervisionato

Dati

Consideriamo il paradigma di Apprendimento Supervisionato

Dati a nostra disposizione (**off-line**)

$$\text{Dati} = \{(x^{(1)}, f(x^{(1)})), \dots, (x^{(N)}, f(x^{(N)}))\}$$

Suddivisione tipica ($N = N_{tr} + N_{ts}$) :

- **Training Set** = $\{(x^{(1)}, f(x^{(1)})), \dots, (x^{(N_{tr})}, f(x^{(N_{tr})}))\}$

usato direttamente dall'algoritmo di apprendimento;

- **Test Set** = $\{(x^{(1)}, f(x^{(1)})), \dots, (x^{(N_{ts})}, f(x^{(N_{ts})}))\}$

usato alla fine dell'apprendimento per **stimare** la bontà della soluzione.

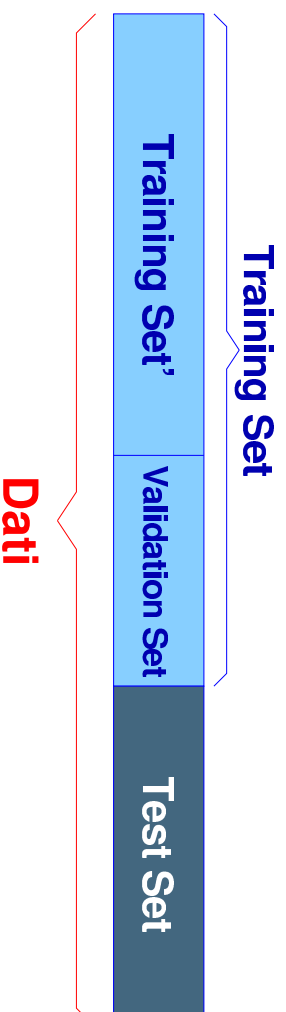


Dati (cont.)

Se N abbastanza grande il **Training Set** è ulteriormente suddiviso in due sottoinsiemi ($N_{tr} = N_{\hat{tr}} + N_{val}$):

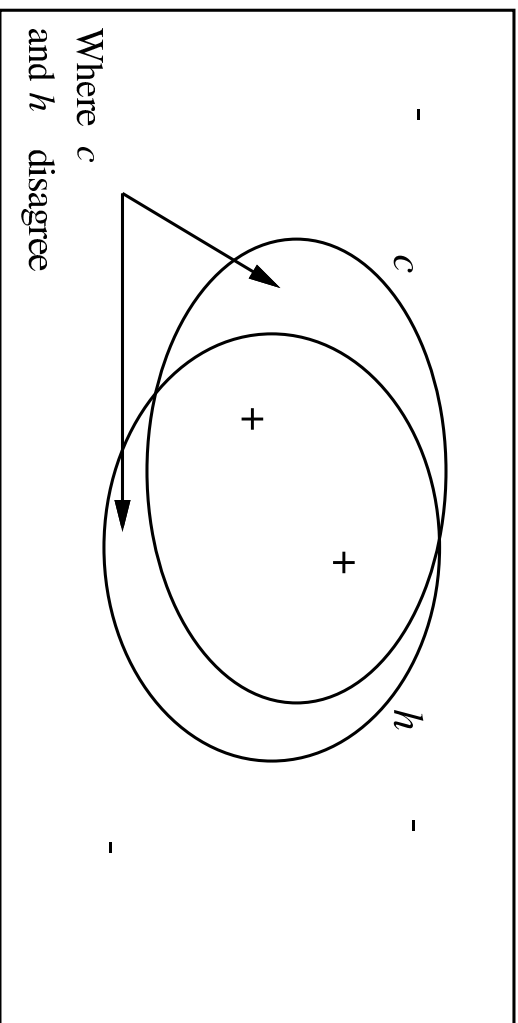
- Training Set** = $\{(x^{(1)}, f(x^{(1)})), \dots, (x^{(N_{\hat{tr}})}, f(x^{(N_{\hat{tr}})}))\}$
 usato **direttamente** dall'algoritmo di apprendimento;
- Validation Set** = $\{(x^{(1)}, f(x^{(1)})), \dots, (x^{(N_{val})}, f(x^{(N_{val})}))\}$
 usato **indirettamente** dall'algoritmo di apprendimento.

Il **Validation Set** serve per **scegliere** l'ipotesi $h \in \mathcal{H}$ migliore fra quelle **consistenti** con il **Training Set**



Errore Ideale

Instance Space X



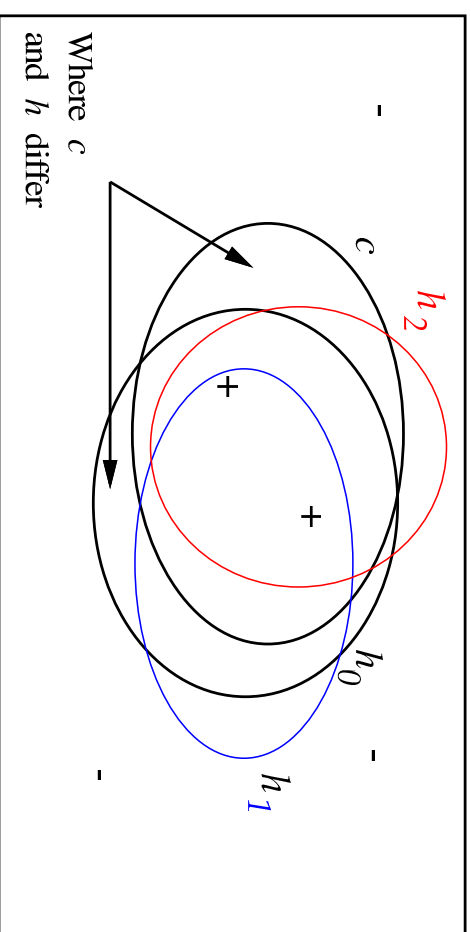
Supponiamo che la funzione f da apprendere sia una funzione booleana (**concetto**):

$$f : X \rightarrow \{0, 1\} (\{-, +\})$$

Def: L'**Errore Ideale** ($error_{\mathcal{D}}(h)$) di una ipotesi h rispetto al concetto f e la distribuzione di probabilità \mathcal{D} (probabilità di osservare l'ingresso $x \in X$) è la probabilità che h classifi chi erroneamente un input selezionato a caso secondo \mathcal{D} : $error_{\mathcal{D}}(h) \equiv Pr_{x \in \mathcal{D}} [f(x) \neq h(x)]$

Errore di Apprendimento

Instance Space X



Dato $T^r = \text{Training Set}$, più ipotesi possono essere consistenti: h_0, h_1, h_2 quale scegliere ?

Def: L'**Errore Empirico** ($error_{T^r}(h)$) di una ipotesi h rispetto a T^r è il numero di esempi che h classifi ca erroneamente: $error_{T^r}(h) \equiv \#\{(x, f(x)) \in T^r \mid f(x) \neq h(x)\}$

Def: Una ipotesi $h \in \mathcal{H}$ è **sovraspecializzata (overfit)** T^r se $\exists h' \in \mathcal{H}$ tale che $error_{T^r}(h) < error_{T^r}(h')$, ma $error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$.

Il **Validation Set** serve per cercare di selezionare l'ipotesi migliore (evitare **overfit**).

VC-dimension

Definizione: Frammentazione (Shattering)

Dato $S \subset X$, S è frammentato (shattered) dallo spazio delle ipotesi \mathcal{H} se e solo se

$$\forall S' \subseteq S, \exists h \in \mathcal{H}, \text{ tale che } \forall x \in S, h(x) = 1 \Leftrightarrow x \in S'$$

(\mathcal{H} realizza tutte le possibili dicotomie di S)

Definizione: VC-dimension

La VC-dimension di uno spazio delle ipotesi \mathcal{H} definito su uno spazio delle istanze X è data dalla cardinalità del sottoinsieme più grande di X che è frammentato da \mathcal{H} :

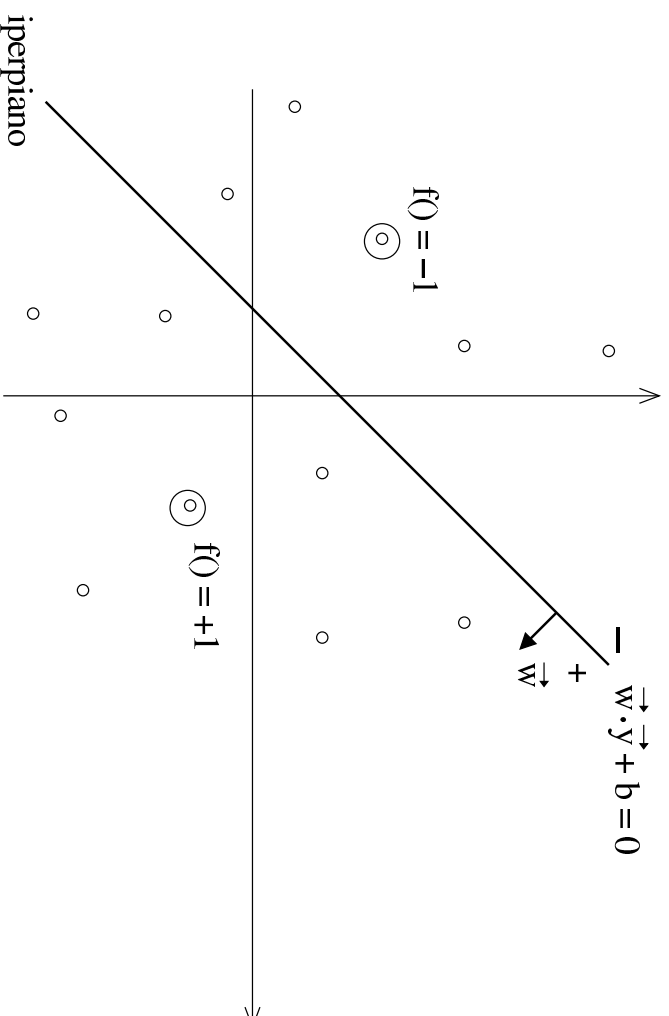
$$VC(\mathcal{H}) = \max_{S \subseteq X} |S| : \mathcal{H} \text{ frammenta } S$$

$VC(\mathcal{H}) = \infty$ se S non è limitato

VC-dimension: Esempio

Quale è la VC-dimension di \mathcal{H}_1 ?

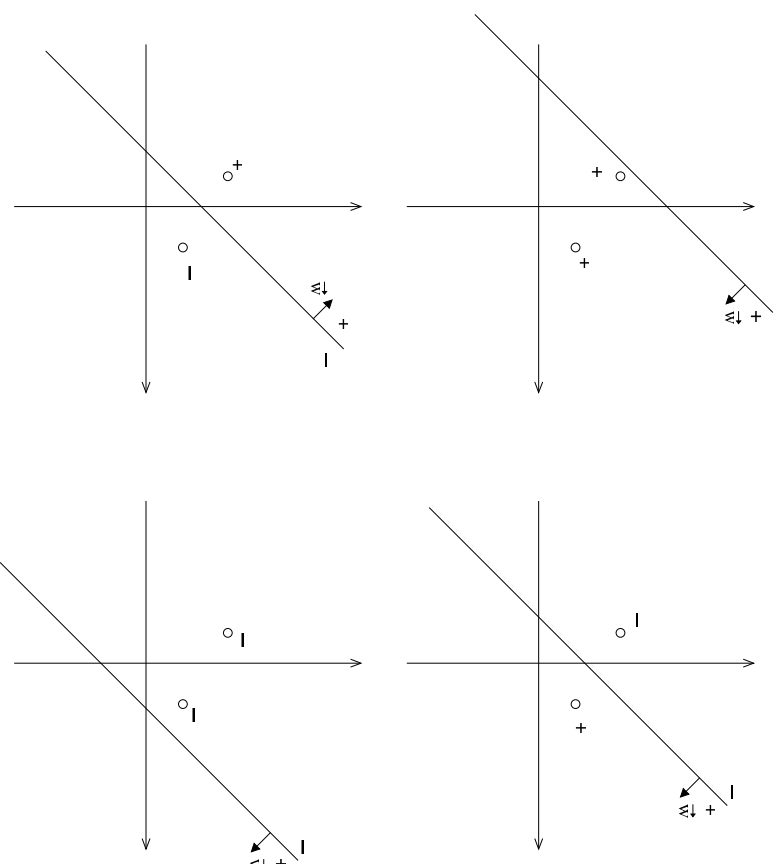
$$\mathcal{H}_1 = \{f_{(\vec{w}, b)}(\vec{y}) \mid f_{(\vec{w}, b)}(\vec{y}) = \text{sign}(\vec{w} \cdot \vec{y} + b), \vec{w} \in \mathbb{R}^2, b \in \mathbb{R}\}$$



VC-dimension: Esempio

Quale è la VC-dimension di \mathcal{H}_1 ?

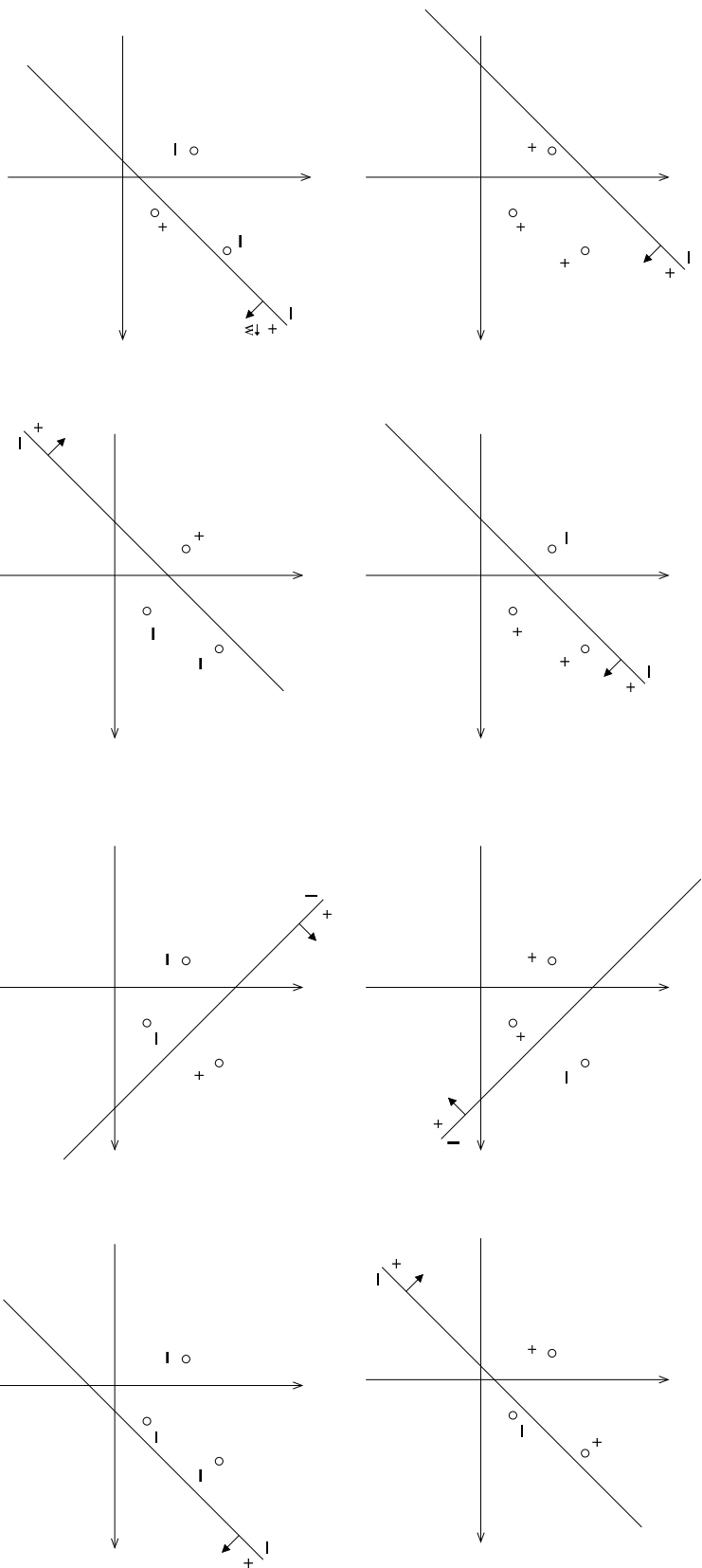
$VC(\mathcal{H}) \geq 1$ banale. Vediamo cosa succede con 2 punti:



VC-dimension: Esempio

Quale è la VC-dimension di \mathcal{H}_1 ?

Quindi $VC(\mathcal{H}) \geq 2$. Vediamo cosa succede con 3 punti:



VC-dimension: Esempio

Quale è la VC-dimension di \mathcal{H}_1 ?

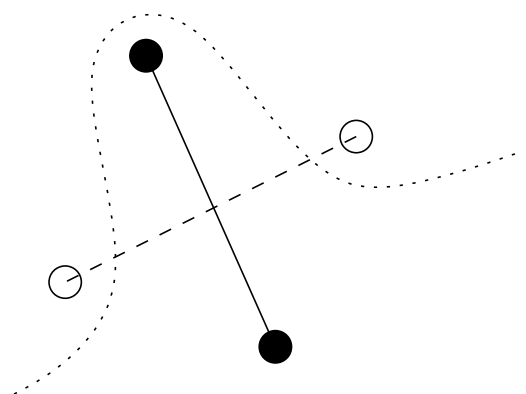
Quindi $VC(\mathcal{H}) \geq 3$. Cosa succede con 4 punti ?

VC-dimension: Esempio

Quale è la VC-dimension di \mathcal{H}_1 ?

Quindi $VC(\mathcal{H}) \geq 3$. Cosa succede con 4 punti ? Non si riesce a frammentare 4 punti!!

Infatti esisteranno sempre due coppie di punti che se unite con un segmento provocano una intersezione fra i due segmenti e quindi, ponendo ogni coppia di punti in classi diverse, per separarli non basta una retta, ma occorre una curva. **Quindi $VC(\mathcal{H}) = 3$**



Bound sull'Errore Ideale per Classificazione Binaria

Consideriamo un problema di classificazione binario (i.e., apprendimento di concetti). Dati

- Training Set $T_r = \{(\mathbf{x}^{(1)}, f(\mathbf{x}^{(1)})), \dots, (\mathbf{x}^{(N_{tr})}, f(\mathbf{x}^{(N_{tr})}))\}$
- Spazio delle ipotesi $\mathcal{H} = \{h_w(\mathbf{x}) | \mathbf{w} \in \mathbb{R}^k\}$
- Algoritmo di Apprendimento L che restituisce l'ipotesi $h_{w^*}(\mathbf{x})$, dove w^* minimizza l'errore empirico $error_{T_r}(h_w(\mathbf{x}))$

è possibile derivare dei bound sull'errore ideale (detto anche errore di generalizzazione), validi con probabilità $1 - \delta$, che hanno una forma del tipo

$$error_{\mathcal{D}}(h_{w^*}(\mathbf{x})) \leq error_{T_r}(h_{w^*}(\mathbf{x})) + \epsilon(N_{tr}, VC(\mathcal{H}), \delta)$$

Esempio:

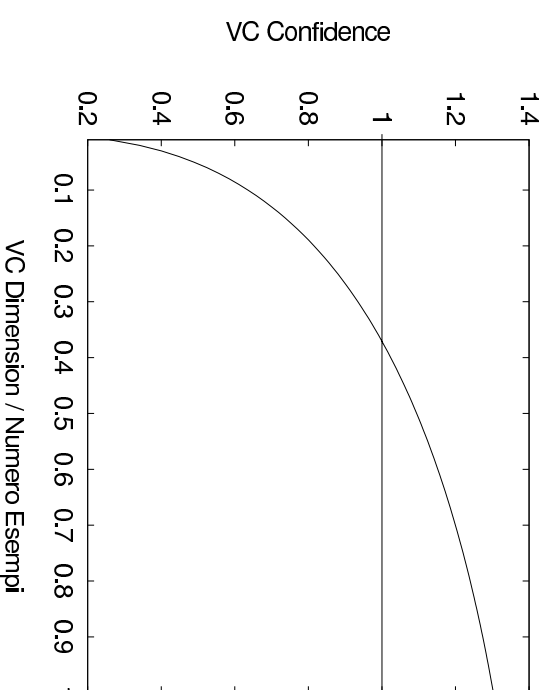
$$error_{\mathcal{D}}(h_{w^*}(\mathbf{x})) \leq \underbrace{error_{T_r}(h_{w^*}(\mathbf{x}))}_A + \underbrace{\sqrt{\frac{VC(\mathcal{H})}{N_{tr}} (\log(\frac{2N_{tr}}{VC(\mathcal{H})}) + 1) - \frac{1}{N_{tr}} \log(\delta)}}_B$$

Bound sull'Errore Ideale per Classificazione Binaria

Si noti che

- il termine **A** DIPENDE SOLO dalla ipotesi restituita dall'algoritmo di apprendimento L ;
- il termine **B** è **INDIPENDENTE** dalla ipotesi restituita dall'algoritmo di apprendimento L ; in particolare dipende dal rapporto fra VC-dimension dello spazio delle ipotesi \mathcal{H} e il numero di esempi di apprendimento (N_{tr}), oltre ovviamente che dalla confidenza $(1 - \delta)$ con cui il bound è valido.

Il termine **B** è usualmente chiamato VC-confidenza e risulta essere monotono rispetto al rapporto $\frac{VC(\mathcal{H})}{N_{tr}}$; fissato N_{tr} aumenta all'aumentare di $VC(\mathcal{H})$.



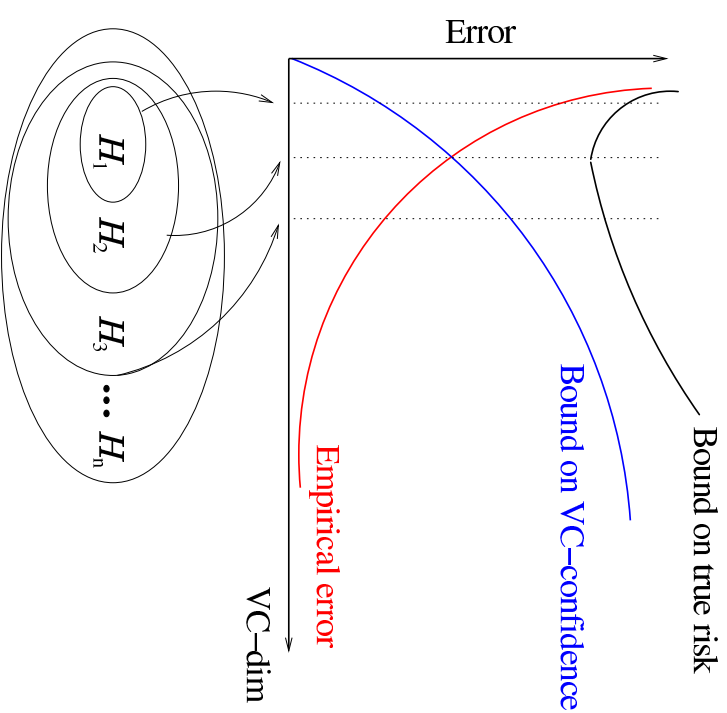
Structural Risk Minimization

Problema: all'aumentare della VC-dimension diminuisce l'errore empirico (termine A), ma **aumenta la VC confidence (termine B)!**

L'approccio **Structural Risk Minimization** tenta di trovare un compromesso tra i due termini:

Si considerano \mathcal{H}_i tali che

- $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots \subseteq \mathcal{H}_n$
- $VC(\mathcal{H}_1) \leq \dots \leq VC(\mathcal{H}_n)$
- si seleziona l'ipotesi che ha il bound sull'errore ideale pi' u basso



Esempio: Reti neurali con un numero crescente di neuroni nascosti

Apprendimento di concetti: alcune definizioni

Definizione: Un concetto in uno Spazio delle Istanze (Instance Space) X è definito come una funzione booleana su X .

Definizione: Un esempio di un concetto c su uno Spazio delle Istanze X è definito come una coppia $(x, c(x))$, dove $x \in X$ e $c()$ è una funzione booleana.

Definizione: Poniamo h essere una funzione booleana definita sullo Spazio delle Istanze X . Diciamo che h soddisfa $x \in X$ se $h(x) = 1$ (*true*).

Definizione: Poniamo h essere una funzione booleana definita sullo Spazio delle Istanze X e $(x, c(x))$ un esempio di $c()$. Diciamo che h è consistente con l'esempio se $h(x) = c(x)$. In più diciamo che h è consistente con un insieme di esempi Tr se h è consistente con ogni esempio in Tr .

Spazio delle Ipotesi: ordine parziale

Definizione: Siano h_i e h_j funzioni booleane definite su uno Spazio delle Istanze

X . Diciamo che h_i è più generale o equivalente di h_j ($h_i \geq_g h_j$) se e solo se

$$(\forall x \in X)[(h_j(x) = 1) \rightarrow (h_i(x) = 1)]$$

Esempi

- $l_1 \geq_g (l_1 \wedge l_2)$
- $l_2 \geq_g (l_1 \wedge l_2)$
- $l_1 \not\geq_g l_2$ e $l_2 \not\geq_g l_1$ (non comparabili)

Esercizio: apprendimento di congiunzioni di letterali

Algoritmo **Find-S**

/* trova l'ipotesi più specifica che è consistente con l'insieme di apprendimento */

- input: insieme di apprendimento T^r
- inizializza h con ipotesi più specifica
$$h \equiv l_1 \wedge \neg l_1 \wedge l_2 \wedge \neg l_2 \wedge \dots \wedge l_m \wedge \neg l_m$$
- per ogni istanza di apprendimento positiva $(x, true) \in T^r$
 - rimuovi da h ogni letterale che non sia soddisfatto da x
- restituisci h

Esempio di applicazione: $m = 5$

Esempio (positivo)	ipotesi corrente
	$h_0 \equiv l_1 \wedge \neg l_1 \wedge l_2 \wedge \neg l_2 \wedge l_3 \wedge \neg l_3 \wedge l_4 \wedge \neg l_4 \wedge l_5 \wedge \neg l_5$
1 1 0 1 0	$h_1 \equiv l_1 \wedge l_2 \wedge \neg l_3 \wedge l_4 \wedge \neg l_5$
1 0 0 1 0	$h_2 \equiv l_1 \wedge \neg l_3 \wedge l_4 \wedge \neg l_5$
1 0 1 1 0	$h_3 \equiv l_1 \wedge l_4 \wedge \neg l_5$
1 0 1 0 0	$h_4 \equiv l_1 \wedge \neg l_5$
0 0 1 0 0	$h_5 \equiv \neg l_5$

Notare che $h_0 \leq_g h_1 \leq_g h_2 \leq_g h_3 \leq_g h_4 \leq_g h_5$

Inoltre, ad ogni passo l'ipotesi corrente h_i è sostituita dall'ipotesi h_{i+1} che costituisce una *generalizzazione minima* di h_i : consistente con l'esempio corrente.

Pertanto **Find-S** restituisce l'ipotesi più specifica consistente con T^r