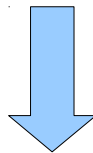


Overfitting

Errore di generalizzazione



Learning set



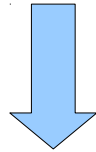
Definizione di sedia:

`(numero_gambe = 4) && (schienale == sì)`

Errore di generalizzazione

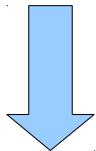


Test set



Sono sedie?

(numero_gambe = 4) && (schienale == sì)

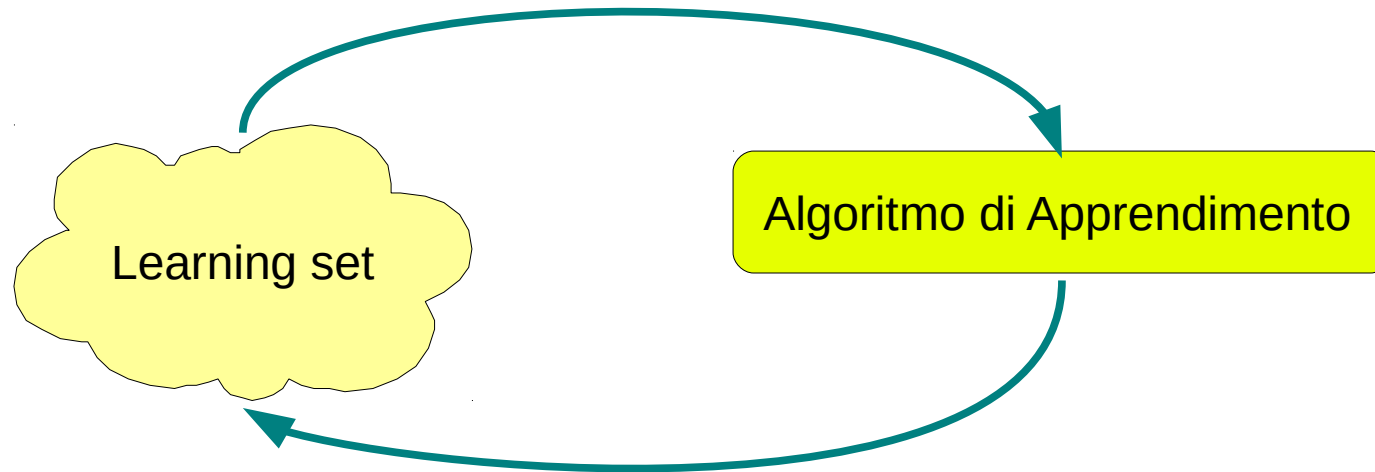


No !!



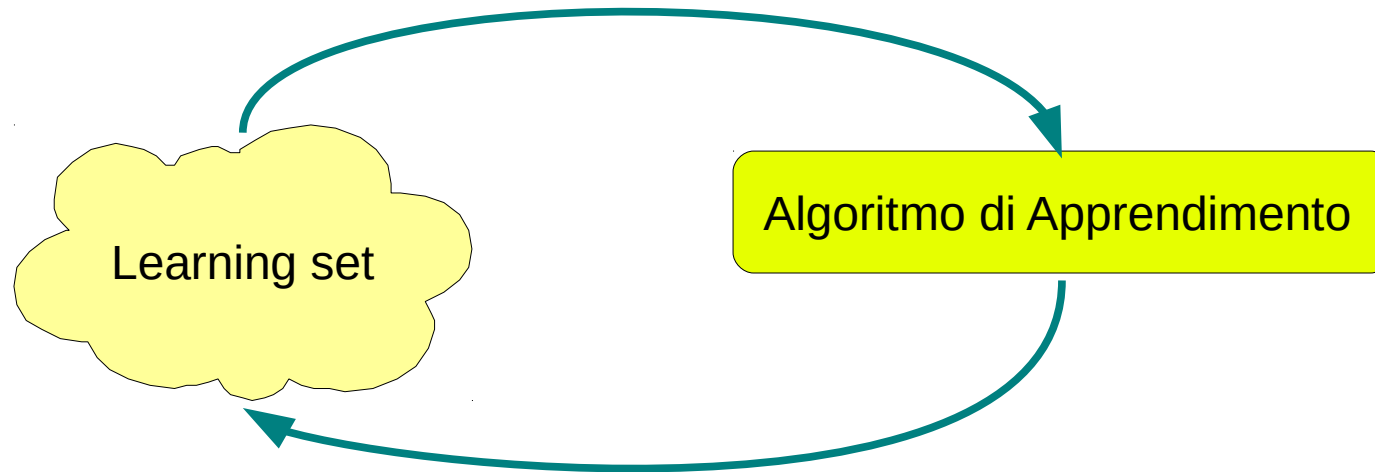
Il modello appreso è troppo specifico
NB: l'esempio è semplice per fornire un'intuizione, di solito si ha overfitting con alberi grandi (modelli complessi)

A cosa è dovuto?



Un possibile condizione di terminazione è: itera l'applicazione dell'algoritmo finché l'errore di classificazione degli esempi di training non scende al di sotto di una certa soglia

A cosa è dovuto?



Un possibile condizione di terminazione è: itera l'applicazione dell'algoritmo finché l'errore di classificazione degli esempi di training non scende al di sotto di una certa soglia

Caso 1: noise
per esempio alcune istanze sono classificate in modo errato

Caso 2: mancanza di esempi
il learning set non rappresenta tutti i casi significativi

Overfitting dovuto a confronti multipli

Consideriamo la costruzione di un albero di decisione: ad ogni iterazione occorre individuare un attributo su cui effettuare il test. Un attributo **viene preso in considerazione** se il guadagno che dà supera una soglia minima

Spesso la procedura di costruzione è **greedy**: cerca di massimizzare il guadagno

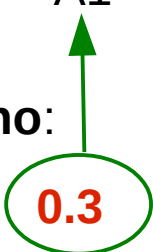
Overfitting dovuto a confronti multipli

Consideriamo la costruzione di un albero di decisione: ad ogni iterazione occorre individuare un attributo su cui effettuare il test. Un attributo **viene preso in considerazione** se il guadagno che dà supera una soglia minima

Spesso la procedura di costruzione è **greedy**: cerca di massimizzare il guadagno

 Nodo corrente

Attributi per cui il guadagno è sufficiente (di solito ci sono più possibilità):

	A1	A2	A3	A4
Guadagno:	 0.3	0.2	0.2	0.1

Guadagno massimo significa **miglior modellazione delle istanze di learning**

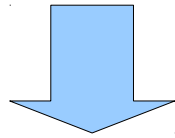
Overfitting

Modello ideale: modello che produce il *minor errore di generalizzazione* possibile. Come poter approssimare il modello ideale quando si ha a disposizione solo un insieme di esempi di learning?

Overfitting

Modello ideale: modello che produce il *minor errore di generalizzazione* possibile. Come poter approssimare il modello ideale quando si ha a disposizione solo un insieme di esempi di learning?

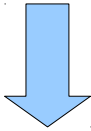
Rasoio di Occam!



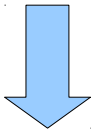
Incorporare una nozione di complessità nel modello
A parità di errore i modelli più semplici sono preferibili

Minimum description length

Implementazione del rasoio di Occam: la migliore ipotesi per la modellazione di un data set è quella che consente la *massima compressione* dei dati



Modello = strumento che consente di rappresentare i dati in modo compatto catturando le loro regolarità

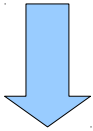


Apprendimento = strumento per catturare regolarità nei dati

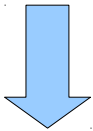
È migliore un modello **accurato** che al contempo è **poco costoso da comunicare** ad un'altra parte che desideri utilizzarlo

Minimum description length

Implementazione del rasoio di Occam: la migliore ipotesi per la modellazione di un data set è quella che consente la *massima compressione* dei dati



Modello = strumento che consente di rappresentare i dati in modo compatto catturando le loro regolarità



Apprendimento = strumento per catturare regolarità nei dati

si usa la formulazione di base del MDL detta **two-part code**, in generale:

Siano $H(1)$, $H(2)$, ... dei **modelli candidati**, contenenti **ipotesi**.

L'ipotesi migliore $H \in H(1) \cup H(2) \cup \dots$ per spiegare i dati D è quella che **minimizza la somma** $L(H) + L(D|H)$, dove:

- $L(H)$ è la lunghezza, in bit, della descrizione dell'ipotesi
- $L(D|H)$ è la lunghezza, in bit, delle descrizioni dei dati codificati con l'aiuto dell'ipotesi.

Il modello migliore per spiegare D è il modello più piccolo.

Minimum description length: nota

Nota: in questa formalizzazione un modello $H(1)$ cattura una famiglia di possibili funzioni (nel nostro caso di classificazione) che hanno tutte la stessa forma.

Un'ipotesi è un'istanza della forma di una funzione.

Esempio:

Modello: $y = A \cdot x^2$

Ipotesi: $y = 0.75 \cdot x^2$

MDL e decision trees

L'MDL deriva dalla *teoria dell'informazione*, la lunghezza in bit indica il costo della trasmissione del modello e dei dati

Esempio di calcolo dell'MDL su alberi di decisione:

$$\text{Costo}(\text{albero}, \text{dati}) = \text{Costo}(\text{albero}) + \text{Costo}(\text{dati} \mid \text{albero})$$

codifica di un nodo: identificatore dell'attributo su cui si fa il test

codifica di una foglia: identificatore della classe associata

Costo(albero): costo della codifica di tutti i suoi nodi

Costo(dati | albero): codifica basata sull'errore di classificazione

MDL e decision trees

L'MDL deriva dalla *teoria dell'informazione*, la lunghezza in bit indica il costo della trasmissione del modello e dei dati

Esempio di calcolo dell'MDL su alberi di decisione:

$$\text{Costo}(\text{albero}, \text{dati}) = \text{Costo}(\text{albero}) + \text{Costo}(\text{dati} \mid \text{albero})$$

codifica di un nodo: identificatore dell'attributo su cui si fa il test

**supponiamo di avere m attributi,
possiamo rappresentarli con un numero.**

Per codificare un numero *compreso fra 1 e m* occorrono $\log_2 m$ bit

Es. per codificare un numero fra 1 e 4 occorrono 2 bit

codifica di una foglia: identificatore della classe associata

Costo(albero): costo della codifica di tutti i suoi nodi

Costo(dati | albero): codifica basata sull'errore di classificazione

MDL e decision trees

L'MDL deriva dalla *teoria dell'informazione*, la lunghezza in bit indica il costo della trasmissione del modello e dei dati

Esempio di calcolo dell'MDL su alberi di decisione:

$$\text{Costo}(\text{albero}, \text{dati}) = \text{Costo}(\text{albero}) + \text{Costo}(\text{dati} \mid \text{albero})$$

codifica di un nodo: identificatore dell'attributo su cui si fa il test

codifica di una foglia: identificatore della classe associata

se abbiamo **k classi** occorrono **$\log_2 k$** bit

Costo(albero): costo della codifica di tutti i suoi nodi

Costo(dati | albero): codifica basata sull'errore di classificazione

MDL e decision trees

L'MDL deriva dalla *teoria dell'informazione*, la lunghezza in bit indica il costo della trasmissione del modello e dei dati

Esempio di calcolo dell'MDL su alberi di decisione:

$$\text{Costo}(\text{albero}, \text{dati}) = \text{Costo}(\text{albero}) + \text{Costo}(\text{dati} \mid \text{albero})$$

codifica di un nodo: identificatore dell'attributo su cui si fa il test

codifica di una foglia: identificatore della classe associata

Costo(albero): costo della codifica di tutti i suoi nodi

possiamo pensare che sia la **somma dei costi dei suoi nodi**

Costo(dati | albero): codifica basata sull'errore di classificazione

MDL e decision trees

L'MDL deriva dalla *teoria dell'informazione*, la lunghezza in bit indica il costo della trasmissione del modello e dei dati

Esempio di calcolo dell'MDL su alberi di decisione:

$$\text{Costo}(\text{albero}, \text{dati}) = \text{Costo}(\text{albero}) + \text{Costo}(\text{dati} \mid \text{albero})$$

codifica di un nodo: identificatore dell'attributo su cui si fa il test

codifica di una foglia: identificatore della classe associata

Costo(albero): costo della codifica di tutti i suoi nodi

Costo(dati | albero): codifica basata sull'errore di classificazione

L'errore è dato fornendo l'istanza classificata erroneamente, quindi per ogni errore viene aggiunto il costo di indicare l'istanza misclassificata, Sia N_E il numero degli errori di classificazione compiuti.

Se il **numero di istanze di training** è n occorrono $\log_2 n$ bit per rappresentarne ciascuna, quindi $\text{Costo}(\text{dati} \mid \text{albero})$ sarà $\log_2 n * N_E$

Si potrebbero tenere molte lezioni sul solo MDL, per approfondimenti:

Tutorial: P.Grünwald, A tutorial introduction to the minimum description length principle. In: Advances in Minimum Description Length: Theory and Applications (edited by P. Grünwald, I.J. Myung, M. Pitt), MIT Press, 2005. (<https://arxiv.org/pdf/math/0406077.pdf> sezione 1.3 in particolare)

Gestione dell'overfitting durante l'induzione

- **Pruning**: potatura dell'albero \Rightarrow semplificazione del modello \Rightarrow generalizzazione del modello
 - Prepruning
 - Postpruning

Gestione dell'overfitting durante l'induzione

- **Pruning**: potatura dell'albero \Rightarrow semplificazione del modello \Rightarrow generalizzazione del modello
 - **Prepruning**: la costruzione del DT si interrompe prima che l'albero sia completo. Si impone una regola di terminazione più restrittiva
 - **Problema**: definire la nuova condizione di terminazione
 - Postpruning

Gestione dell'overfitting durante l'induzione

- **Pruning**: potatura dell'albero \Rightarrow semplificazione del modello \Rightarrow generalizzazione del modello
 - Prepruning
 - **Postpruning**: prima si costruisce l'albero poi si potano alcuni rami, trasformando alcuni nodi interni in foglie
 - **Problema**: definire la condizione per decidere se un ramo è da potare o meno

Prepruning (early stopping rule)

Regola di interruzione, esempio: non eseguo lo split se il gain è al di sotto di una certa soglia

Vantaggio: evita l'overfitting dei dati di learning

Problema: è difficile scegliere la soglia, se troppo alta si ha *underfitting*

Post-pruning

Questo approccio consiste nel tagliare rami da un albero fatto crescere finché non si hanno più guadagni

Possibili strategie:

- (1) sostituisco un sottoalbero col solo suo cammino usato più di frequente;
- (2) sostituisco un sottoalbero con una foglia la cui classe corrisponde alla classe
Maggiormente rappresentata nelle foglie dell'albero rimosso.

Tende a dare *risultati migliori* del pre-pruning

Problema di efficienza: tagliare rami costruiti significa che il tempo trascorso a costruirli è stato sprecato