

Alberi di Decisione per l'Apprendimento

- La rappresentazione è un albero di decisione.
- La parzialità è verso alberi di decisione semplici.
- Cerca attraverso lo spazio degli alberi di decisione, da semplici alberi di decisione ad altri più complessi.

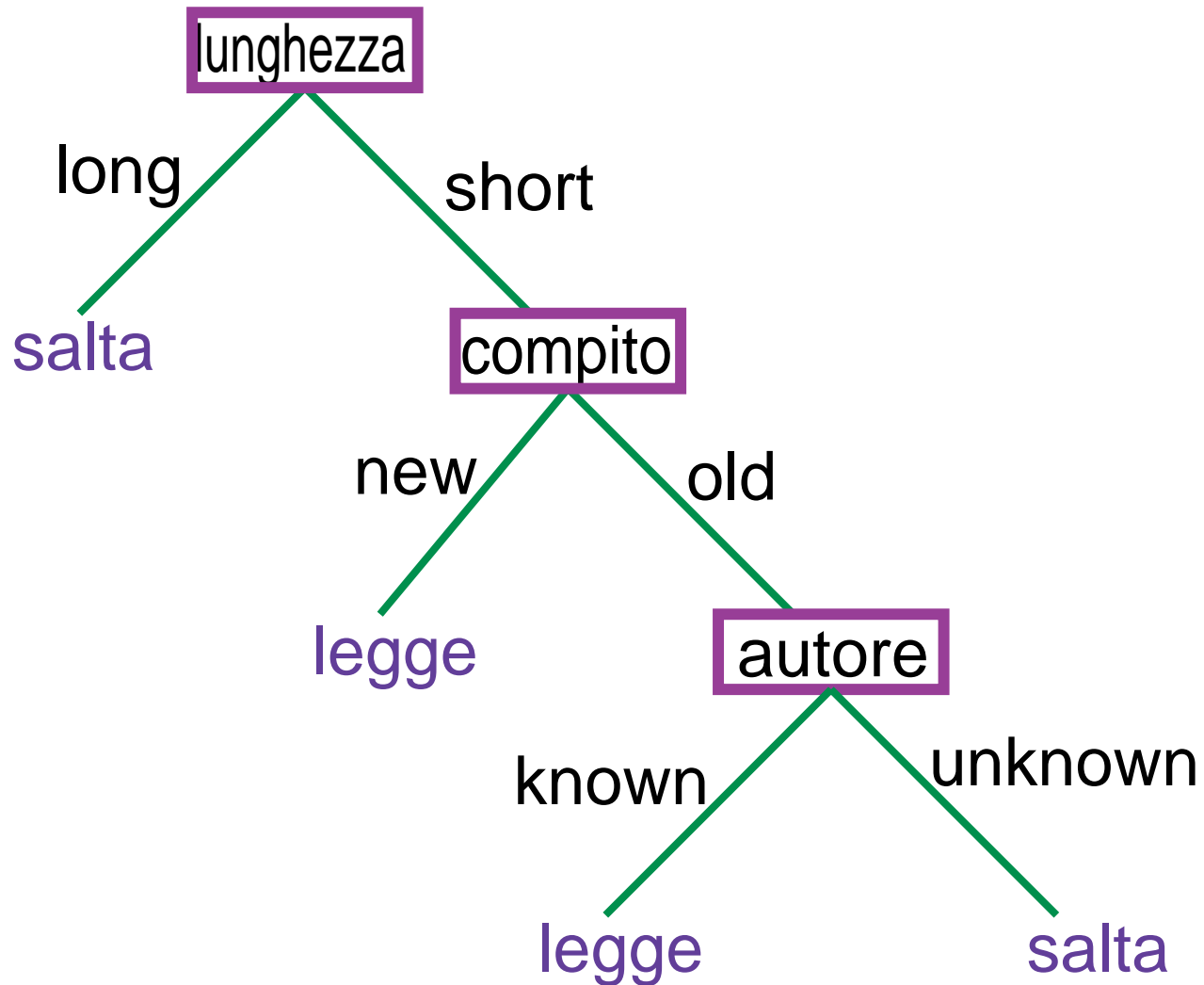


Alberi di Decisione

Un **albero di decisione** è un albero dove:

- I nodi non foglia sono etichettati con attributi.
- Gli archi al di fuori di un nodo etichettato con attributo A sono etichettati con ognuno dei possibili valori dell'attributo A .
- Le foglie dell'albero sono etichettate con classificazioni.

Albero di Decisione di Esempio



Programma Logico Equivalente

$prop(Obj, user_action, skips) \leftarrow$
 $prop(Obj, length, long).$

$prop(Obj, user_action, reads) \leftarrow$
 $prop(Obj, length, short) \wedge prop(Obj, thread, new).$

$prop(Obj, user_action, reads) \leftarrow$
 $prop(Obj, length, short) \wedge prop(Obj, thread, old) \wedge$
 $prop(Obj, author, known).$

$prop(Obj, user_action, skips) \leftarrow$
 $prop(Obj, length, short) \wedge prop(Obj, thread, old) \wedge$
 $prop(Obj, author, unknown).$



Problemi nell'Apprendimento con alberi di decisione

- Dati alcuni dati, quali alberi di decisione dovrebbero essere generati? Un albero di decisione può rappresentare qualsiasi funzione discreta degli input.
- Hai bisogno di una **parzialità**. Esempio, preferire l'albero più piccolo. Meno profondo? Meno nodi? Quale albero è il miglior predittore di dati non visti?
- Come costruisci un albero di decisione? Lo spazio degli alberi di decisione è troppo grande per la ricerca sistematica per l'albero di decisione più piccolo.



Trovare un buon albero di decisione

- L'input è un attributo destinazione (il *Goal*), un insieme di esempi, e un insieme di attributi.
- Fermati se tutti gli esempi hanno la stessa classificazione.
- Altrimenti, scegli un attributo da dividere,
 - per ogni valore di questo attributo, costruisci un sottoalbero per questi esempi con questi valori di attributi.

Apprendimento con alberi di decisione: attributi booleani

% *dtlearn*(*Goal*, *Examples*, *Attributes*, *DT*) dato *Examples*
% and *Attributes* costruisci albero di decisione *DT* per *Goal*.

dtlearn(*Goal*, *Exs*, *Atts*, *Val*) \leftarrow

all_examples_agree(*Goal*, *Exs*, *Val*).

dtlearn(*Goal*, *Exs*, *Atts*, *if*(*Cond*, *YT*, *NT*)) \leftarrow

examples_disagree(*Goal*, *Exs*) \wedge

select_split(*Goal*, *Exs*, *Atts*, *Cond*, *Rem_Atts*) \wedge

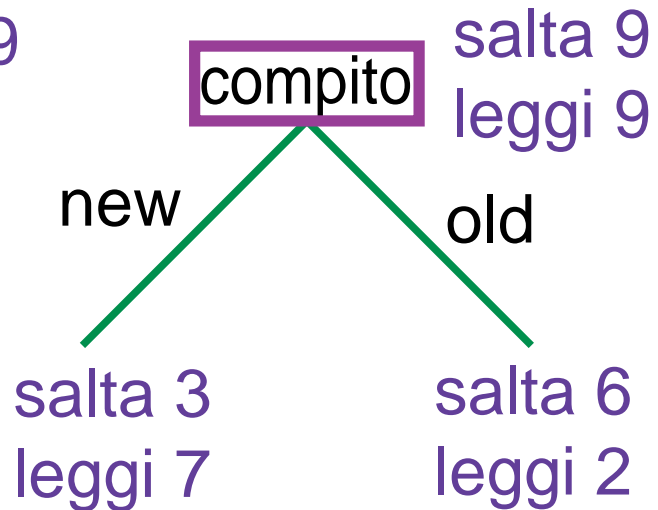
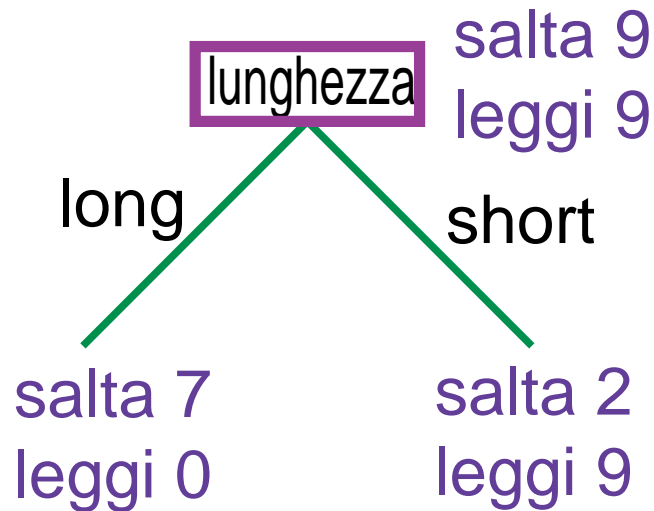
split(*Exs*, *Cond*, *Yes*, *No*) \wedge

dtlearn(*Goal*, *Yes*, *Rem_Atts*, *YT*) \wedge

dtlearn(*Goal*, *No*, *Rem_Atts*, *NT*).



Esempio: possibili divisioni



Usando quest'algoritmo in pratica

- Gli attributi possono avere più di 2 valori. Questo complica l'albero.
- Questo assume che gli attributi siano adeguati a rappresentare il concetto. Puoi restituire le probabilità alle foglie.
- Quale attributo da scegliere da dividere non è definito. Tu devi scegliere l'attributo che risulta nell'albero più piccolo. Spesso usiamo la teoria delle informazioni come funzione di valutazione nell'hill climbing.
- L'overfitting è un problema.



Gestire l'Overfitting

- Questo algoritmo crea problemi nell'overfitting dei dati. Questo succede con rumore e correlazioni negli insiemi di addestramento che non sono riflessi interamente nei dati.
- Per gestire l'overfitting:
 - Puoi restringere la divisione, in modo da dividere solo quando la divisione è utile.
 - Puoi permettere la divisione non ristretta e tagliare l'albero risultante quando crea distinzioni non garantite.

