

## Potatura

**problema 1:** Come effettuare la potatura ?

**problema 2:** Quando fermarsi con la potatura (o alternativamente con l'apprendimento) ?

## Quando Fermarsi...

**problema 1:** Come effettuare la potatura ?

**problema 2:** Quando fermarsi con la potatura (o alternativamente con l'apprendimento) ?

Consideriamo il **Problema 2**.

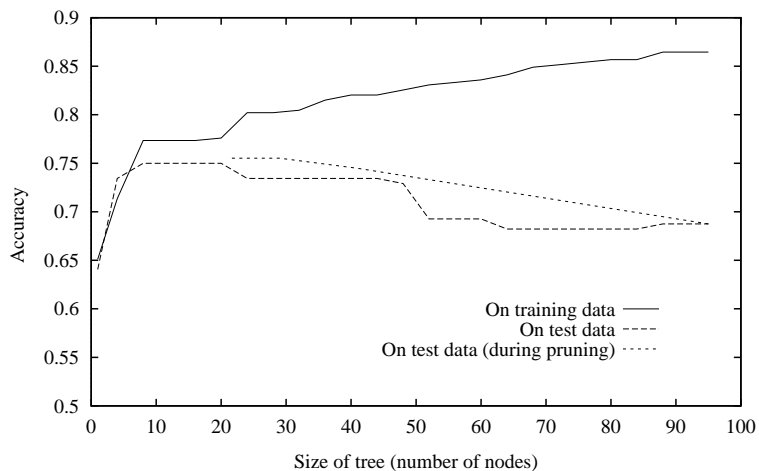
Le principali "soluzioni" sono:

1. Valutare le prestazioni sull'insieme di apprendimento (usando un test statistico);
2. Valutare le prestazioni su un insieme (separato) di validazione;
3. Usare un principio di **minimizzazione della lunghezza di descrizione (MDL)**

$$\min_{Tree} [size(Tree) + size(errori(Tree))]$$

In ogni caso è difficile pervenire ad una soluzione ottima...

## Usando l'insieme di validazione



## Come Potare...

**Problema 1:** Come effettuare la potatura ? Le principali "soluzioni" sono:

1. **Reduced Error Pruning**
  - Dividere  $Tr$  in  $Tr'$  e  $Vs$  (insieme di validazione);
  - Ripetere fino a quando le prestazioni peggiorano
    - (a) per ogni nodo (interno)  $n$  valutare l'impatto su  $Vs$  avendo potato il nodo (e i suoi discendenti);
    - (b) effettuare la potatura che porta alle prestazioni migliori su  $Vs$ ;

(potatura: al sotto-albero radicato in  $n$  si sostituisce una foglia con etichetta uguale alla classe più frequente nell'insieme degli esempi associati al nodo  $n$ )
2. **Rule-Post Pruning**

## Rule-Post Pruning

L'idea di base è quella di trasformare un albero di decisione in un insieme di regole, e poi effettuare la potatura delle regole:

1. Si genera una regola  $R_i$  per ogni cammino  $path(r, f_i)$  dalla radice  $r$  alla foglia  $i$ -esima  $f_i$ ;  $R_i$  sarà nella forma

IF  $(Attr_{i_1} = v_{i_1}) \wedge (Attr_{i_2} = v_{i_2}) \wedge \dots \wedge (Attr_{i_k} = v_{i_k})$  THEN  $label_{f_i}$   
*precondizioni*

2. Si effettua la potatura indipendentemente su ogni regola  $R_i$ :
  - si stimano le prestazioni ottenute utilizzando SOLO  $R_i$  come classificatore;
  - si rimuovono le precondizioni (una o più) che conducono ad un aumento della stima delle prestazioni usando un approccio greedy;
3. Si ordinano le  $R_i$  potate per ordine decrescente di prestazione (evita conflitti); eventualmente, aggiunge come classificazione di default la classe più frequente;

## Considerazioni

Alcune considerazioni sul Rule-Post Pruning:

- la stima delle prestazioni necessaria per effettuare la potatura può essere fatta sia usando un insieme di validazione che utilizzando un test statistico sui dati di apprendimento;
- la trasformazione **Albero**  $\rightarrow$  **Regole** permette di generare regole dove si possono considerare contesti per un nodo che non necessariamente contengono i suoi nodi avi (e in particolare la radice);
- di solito le regole sono più semplici da comprendere per un umano;

In genere il Rule-Post Pruning riesce a migliorare le prestazioni dell'albero di decisione di partenza e si comporta meglio del Reduced-Error Pruning

## Classificare una nuova istanza

La **classificazione** di una nuova istanza da parte delle regole ordinate avviene seguendo l'ordine stabilito per le regole:

- la prima regola la cui precondizione è soddisfatta dalla istanza è usata per generare la classificazione
- se nessuna regola ha le condizioni soddisfatte, si utilizza la regola di default per classificare l'istanza (cioè si ritorna la classe più frequente nell'insieme di apprendimento);

## Reti Neurali in Generale

Le Reti Neurali Artificiali sono studiate sotto molti punti di vista. In particolare, contributi alla ricerca in questo campo provengono da:

- Biologia (Neurofisiologia)
- Informatica (Intelligenza Artificiale)
- Matematica (Ottimizzazione, Proprietà di approssimazione)
- Statistica (Regressione, Classificazione)
- Ingegneria (Pattern Classification, Teoria del Controllo)
- Economia (Studio di serie temporali)
- Fisica (Sistemi Dinamici)
- Psicologia (Apprendimento e Scienze Cognitive)