

Selezione degli Attributi (feature selection)

Input e Complessità

Non la complessità computazionale del problema, ma la complessità del problema stesso, determinata dalla natura del problema.

- alta dimensionalità
- alta correlazione
- alta ridondanza
- alta variabilità
- alta non linearità
- alta non omogeneità
- alta non stazionarietà
- alta non normalità
- alta non gaussianità
- alta non simmetria
- alta non isotropia
- alta non omogeneità
- alta non stazionarietà
- alta non normalità
- alta non gaussianità
- alta non simmetria
- alta non isotropia

Servono tutti gli attributi?

Non tutti gli attributi servono allo stesso modo. Il modello di apprendimento può beneficiare di un sottoinsieme di attributi.

- attributi irrilevanti
- attributi ridondanti
- attributi correlati
- attributi non stazionari
- attributi non normali
- attributi non gaussiani
- attributi non simmetrici
- attributi non isotropi

Selezione degli Attributi (feature selection)

Si può pensare alla selezione degli attributi come a un processo di "filtraggio" degli attributi in base a una serie di criteri di selezione.

- attributi irrilevanti
- attributi ridondanti
- attributi correlati
- attributi non stazionari
- attributi non normali
- attributi non gaussiani
- attributi non simmetrici
- attributi non isotropi

Metodi Generali

Si noti che Forward e Backward Selection sono strategie generali, applicabili a ogni tipo di algoritmo di apprendimento e dipendono dal tipo di algoritmo di apprendimento e dal tipo di attributi da selezionare.

- Forward Selection
- Backward Selection
- Stepwise Selection
- Best Subset Selection
- Exhaustive Search
- Genetic Algorithms
- Simulated Annealing
- Tabu Search
- Particle Swarm Optimization
- Genetic Programming
- Evolutionary Algorithms
- Differential Evolution
- Genetic Algorithms
- Evolutionary Algorithms
- Differential Evolution

Input e Complessità

Sia la complessità computazionale dell'apprendimento che della classificazione/predizione dipendono dalla dimensionalità dell'input

Esempio apprendimento alberi decisione con

- N esempi di apprendimento
- istanze rappresentate con x attributi booleani

Caso pessimo: albero completo appreso con numero di operazioni pari a (c costante > 1):

$$\begin{aligned} \text{NumOp} &= c \left(xN + 2(x-1)\frac{N}{2} + 2^2(x-2)\frac{N}{2^2} + \dots + 2^{x-1}(1)\frac{N}{2^{x-1}} \right) \end{aligned}$$

$$= cN \sum_{i=1}^x i$$

$$= cN \frac{x(x+1)}{2}$$

Inoltre:

meno attributi → albero più semplice

→ classificazione + veloce

complessità quadratica rispetto al numero di attributi!

Servono tutti gli attributi ?

Solo se utili alla classificazione/predizione:

- altrimenti rendono il compito di apprendimento più difficile

Inoltre:

- meno attributi implicano modelli (p.e., alberi di decisione) di classificazione/predizione più compatti e che hanno bisogno di meno esempi di apprendimento per ottenere buoni risultati (minore varianza)
- modelli che usano pochi attributi sono:
 - più semplici da comprendere per un umano
 - più facilmente rappresentabili

Due possibili approcci per ridurre il numero di attributi:

- feature selection ← vediamo esempio di questo approccio
si **seleziona un sottoinsieme** degli attributi fra quelli originari
- feature extraction
 - si **derivano nuovi attributi** da quelli originari

tipici da comprendere per un unia lmente rappresentabili

Due possibili approcci per ridurre il numero di attributi:

- feature selection ← vediamo esempio di questo approccio
si **seleziona un sottoinsieme** degli attributi fra quelli originari
- feature extraction
 - si **derivano nuovi attributi** da quelli originari

Selezione degli Attributi (subset selection)

Si cerca il sottoinsieme "migliore" degli attributi a disposizione;

Si può procedere utilizzando due strategie greedy:

- **Forward Selection**

si parte con l'insieme vuoto degli attributi e poi si aggiungono via via gli attributi "migliori"

insieme corrente di attributi selezionati

$$j = \arg \min_i E(F \cup x_i)$$

errore su Validation Set (pointing to E)
attributo i-esimo (pointing to x_i)

$$\text{add } x_j \text{ to } F \text{ if } E(F \cup x_j) < E(F)$$

aggiungi attributo j-esimo solo se l'errore sul Validation Set migliora

- **Backward Selection**

si parte con l'insieme di tutti gli attributi e poi si rimuovono via via gli attributi "peggiori"

$$j = \arg \min_i E(F - x_i)$$

$$\text{remove } x_j \text{ from } F \text{ if } E(F - x_j) < E(F)$$

rimuovi attributo j-esimo solo se l'errore sul Validation Set migliora

Metodi Generali

Si noti che Forward e Backward Selection:

- sono strategie generali, applicabili a ogni tipo di classificatore/predittore
- sono basate sulla informazione di supervisione a disposizione nel training set
- sono strategie greedy perché non esplorano l'insieme di tutti i sottoinsiemi degli attributi a disposizione:
 - ad esempio, supponendo di avere x attributi a disposizione, non si esplorano tutti i 2^x possibili sottoinsiemi di attributi
- possibile miglioramento con backtracking, ma la complessità computazionale peggiora...