La classificazione

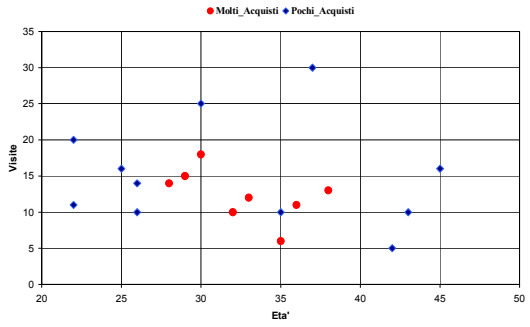
Sommario

- Intuizioni sul task di classificazione
- Classificazione e apprendimento induttivo
  - valutazione di un modello di classificazione
- Tipi di modelli di classificazione
  - alberi di decisione
- Algoritmi di induzione di alberi di decisione
- Un caso di studio

Intuizioni sul concetto di classificazione

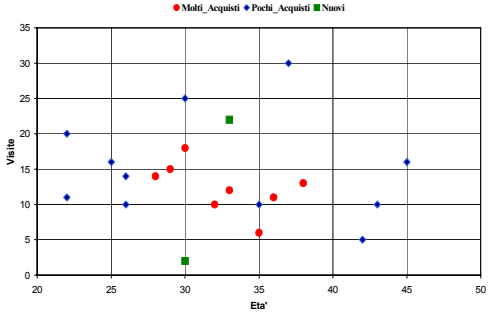
Descrivete la propensione all'acquisto dei clienti …

… in base ad età ed al numero di visite nell'ultimo periodo!

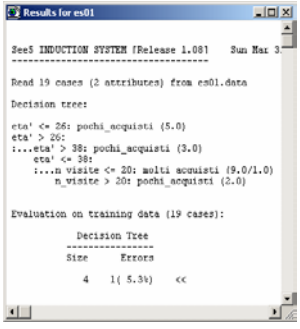


… ora provate a prevedere …

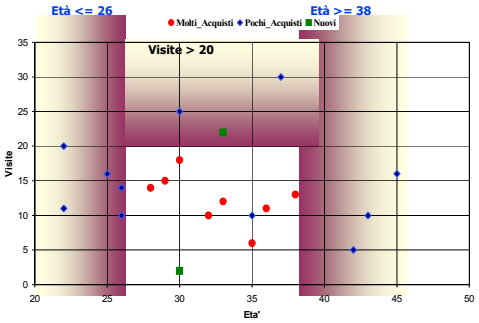
… la tipologia di due nuovi clienti, in base a età e numero visite



Una possibile soluzione …



… rappresentata graficamente



Il processo seguito

- Individuazione di una caratteristica “interessante” dei clienti: concetto target (classe)
  - Numero di acquisti (fare molti acquisti o pochi acquisti)
- Individuazione di altri attributi “meno interessanti”, ma che possono influenzare il target
  - Età, Numero di visite
- Raccolta di dati storici di cui si conoscono TUTTI gli attributi: training set
  - 19 clienti di cui si conoscono TUTTI gli attributi
- Induzione di un modello (di CLASSIFICAZIONE) a partire dai dati storici:
  - descrive la dipendenza del target dagli altri attributi
  - tale descrizione cerca di essere accurata, concisa, informativa
- Predizione (CLASSIFICAZIONE):
  - uso del modello per prevedere il valore del target per un nuovo caso (cliente)

Caratteristiche della classificazione

- Essa implica un processo di *apprendimento:*
  - *Supervisionato:* le classi sono note a priori e si dispone di esempi già classificati
  - *Induttivo:* si genera un modello che descrive gli esempi
- Ipotesi dell’apprendimento induttivo:
  - << Se un modello, indotto da un insieme di esempi “abbastanza rappresentativo”, ne riconosce le classi in modo accurato, sarà accurato anche nel riconoscere la classe degli altri oggetti del dominio >>
- Denominazioni alternative di tale processo:
  - Apprendimento di Concetti (la classe)
  - Induzione di Classificatori

Induzione vs. Deduzione

Verifica top-down

Deduzione

Modello o Pattern

Example A
Example B
Example C

Induzione

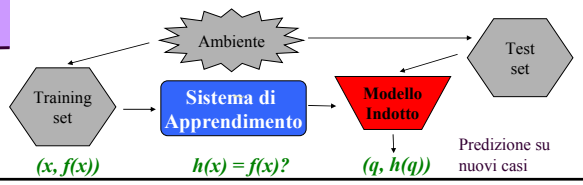
Costruzione bottom-up

Apprendimento induttivo

- *Scopo*:
  - Sviluppare un modello generale o *ipotesi* da un insieme specifico di esempi
- *Input:*
  - Concetto da apprendere
  - Instanze: esempi indipendenti e specifici del concetto
- *Output:*
  - Descrizione del concetto (o Ipotesi o Modello) in funzione delle proprietà delle istanze

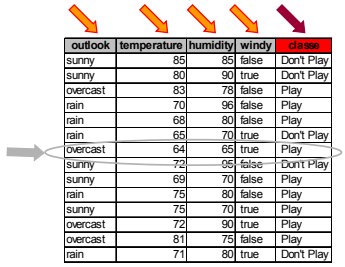
Sistemi di apprendimento induttivo

- Scenario:
  - Dati, una variabile target **y** (concetto) e una lista di attributi **X**
  - Si assume che **y** dipenda da **X** secondo una certa funzione **f**
  - Si ha un **insieme di esempi** di cui si conoscono i valori sia degli attributi **X** sia del target **y= f(X)**
- Problema:
  - trovare la migliore rappresentazione *h()* (ipotesi) per *f()*



Apprendimento induttivo: input

Un insieme di esempi (dette **istanze** o casi) che "caratterizzano" il concetto target (es., la **classe**) sulla base di **attributi** (detti anche features)



outlook	temperature	humidity	windy	classe
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

Classificazione: input

- La classe e gli attributi possono assume valori:
  - **Continui** (es. numeri reali)
    - Due valori sono confrontabili ed esiste una nozione di distanza
    - Esempi: età, reddito, costo, temperatura, ecc.
  - **Categorico o discreto** (elementi di insiemi finiti)
    - **ordinali** (insiemi finiti ed **ordinati**)
      - Due valori sono confrontabili, ma non esiste una distanza
      - Esempi: { freddo, tiepido, caldo }, {insufficiente, buono, ottimo }
    - **nominali** (insiemi finiti e **non ordinati**)
      - Due valori non sono tra loro confrontabili
      - Es.: {nuvoloso, soleggiato, ventoso}, {alimentari, abbigliamento }
- Discretizzazione:
  - trasformazione di valori continui in ordinali

Classificazione e regressione

- Si costruisce un modello predittivo
  - permette di stimare il valore del concetto target per nuovi casi (per cui esso non è noto) in base a quelli degli altri attributi
- Se il target è discreto, si parla di classificazione:

  Alberi di decisione, Regole di classificazione, Classificazione Bayesiana, K-nearest neighbors, Case-based reasoning, Genetic algorithms, Rough sets, Fuzzy logic, Association-based classification
- Se il target è continuo, si parla di regressione:

  Alberi di regressione, Reti neurali, regressione lineare e multipla, regressione non-lineare, regressione logistica e di Poisson, regressione log-lineare

Classificazione: Un esempio

- Dominio:
  - possibili giornate,descritte dai 4 attributi (Tempo, Temperatura, Umidità, Vento)
  - 2 classi:
    - *Si gioca* = {giornate in cui si può giocare}
    - *Non si gioca* = {giornate in cui non si può giocare}
- Obiettivo:
  - indurre un modello di classificazione per prevedere se in una generica giornata si può giocare o no in base ai valori degli attributi

Classificazione: Un esempio

training set: 14 giornate di cui è nota la classe

Tempo	Temper.(°F)	Umidità	Vento	Classe
Soleggiato	85	85	No	Non si gioca
Soleggiato	80	90	Si	Non si gioca
Nuvoloso	83	78	No	Si gioca
Piovoso	70	96	No	Si gioca
Piovoso	68	80	No	Si gioca
Piovoso	65	70	Si	Non si gioca
Nuvoloso	64	65	Si	Si gioca
Soleggiato	72	95	No	Non si gioca
Soleggiato	69	70	No	Si gioca
Piovoso	75	80	No	Si gioca
Soleggiato	75	70	Si	Si gioca
Nuvoloso	72	90	Si	Si gioca
Nuvoloso	81	75	No	Si gioca
Piovoso	71	80	Si	Non si gioca

Qualità di un algoritmo di induzione

- Efficienza
  - in fase di apprendimento
  - in fase di predizione (classificazione)
- Robustezza
  - rispetto a rumore e valori mancanti

Qualità di un modello di classificazione

- Capacità descrittiva
  - Misura l'interpretabilità della rappresentazione del modello e la conoscenza sui dati che esprime
  - Decresce all'aumentare dalla dimensione del modello (es., nr. nodi dell'albero, nr. regole)
- Capacità predittiva
  - *accuratezza*: numero di oggetti classificati correttamente sul numero totale di oggetti classificati
  - *errore di classificazione*: 1 – accuratezza

Accuratezza del modello di classificazione

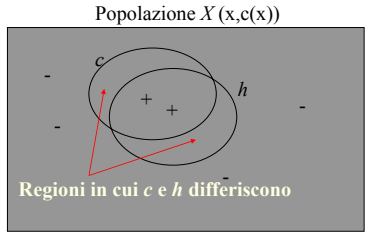
- accuratezza
  - percentuale di esempi la cui classe predetta coincide con la classe reale
- errore di classificazione
  - % di misclassificazione: % di esempi la cui classe predetta NON coincide con quella reale
- … di quali esempi?
  - Di esempi di cui si conosce la classe reale!
  - L'accuratezza sugli esempi del training set è ottimistica: si può anche avere errore=0

Accuratezza: 3 nozioni di errore

- True Error (ideale)
  - Misura quante volte $c(x) \neq h(x)$ sull'intera popolazione X
  - rappresenta la probabilità di errare su un oggetto estratto in modo random
- Training Error
  - Misura quante volte $c(x) \neq h(x)$ nel training set
- Sample Error
  - Misura quante volte $c(x) \neq h(x)$ su un test set independente

Accuratezza: errore di classificazione

x = attributi di input
c = funzione target di classificazione
h = ipotesi (modello)



Il *true error* di un modello h è la probabilità che h classifichi in modo errato una istanza estratta casualmente dalla popolazione X

$$err(h) = P[c(x) \neq h(x)]$$

Accuratezza: limiti inferiori

- Accuratezza di un classificatore casuale
  - classificatore che in modo casuale classifica in una delle k possibili classi
  - accuratezza = 1 / k * 100
  - esempio: per k = 2 l'accuratezza è del 50%
- Qualsiasi classificatore deve fare almeno meglio del classificatore casuale!
- Accuratezza di un classificatore a maggioranza
  - classificatore che classifica sempre come la classe di maggioranza nel training set
  - (supponendo la stessa percentuale nella popolazione) accuratezza >= 1 / k * 100
  - esempio: con due classi rappresentate al 98% e al 2% nel training set, l'accuratezza è del 98%

Stima accuratezza: metodo "holdout"

- Test set
  - insieme di esempi indipendenti dal training set
  - di ciascun esempio si conoscono gli attributi e la classe reale
- Metodo "holdout" (training and test)
  - valutare l'accuratezza del modello sul test set
  - l'insieme di esempi si divide (casualmente) in una parte per il training e una per il test set
    - in genere 2/3 per il training e 1/3 per il test set

Stima accuratezza: "cross-validation"

- Metodo "stratified n-fold cross-validation"
  - Si dividono gli esempi in *n* insiemi *S1*, …,*Sn* con stessa dimensione e stessa distribuzione delle classi
  - Per ciascun insieme Si
    - Si costruisce un classificatore sugli altri n-1 insiemi
    - Si usa *Si* come insieme di test del classificatore
  - L'accuratezza finale è la media delle accuratezze degli *n* classificatori
- Metodo standard: n = 10

  (stratified 10-fold cross validation)

Dettagli sugli errori di classificazione: la matrice di confusione

Name	Gender	Height	Actual	Pred.
Kristina	F	1.6m	Short	Medium
Jim	M	2m	Tall	Medium
Maggie	F	1.9m	Medium	Tall
Martha	F	1.88m	Medium	Tall
Stephanie	F	1.7m	Short	Medium
Bob	M	1.85m	Medium	Medium
Kathy	F	1.6m	Short	Medium
Dave	M	1.7m	Short	Medium
Worth	M	2.2m	Tall	Tall
Steven	M	2.1m	Tall	Tall
Debbie	F	1.8m	Medium	Medium
Todd	M	1.95m	Medium	Medium
Kim	F	1.9m	Medium	Tall
Amy	F	1.8m	Medium	Medium
Wynette	F	1.75m	Medium	Medium

- colonne: classi predette
- righe: classi effettive
- cella (**i**,**j**): numero di istanze di classe **i** che il modello assegna alla classe **j**
- Le predizioni corrette sono sulla diagonale, mentre quelle scorrette (*misclassificazioni*) sono fuori dalla diagonale

Actual Membership	Assignment Short	Medium	Tall
Short	0	4	0
Medium	0	5	3
Tall	0	1	2

Accuratezza del modello rispetto ad una delle classi

- True Positive (TP):

  esempi di classe C classificati come C
- True Negatives (TN):

  esempi di altre classi non assegnati a C
- False Positive (FP)

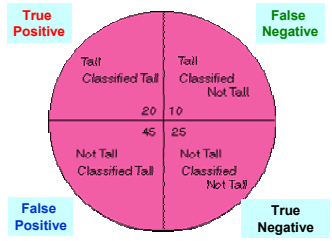
  esempi di altre classi classificati come C (errore!)
- False Negatives (FN):

  esempi di C assegnati ad un'altra classe (errore!)
- Precision
  - **TP/(TP+FP)**
  - Misura di correttezza
- Recall
  - **TP/(TP+FN)**
  - Misura di completezza
- F-measure:

  (2*recall*precision) / (recall+precision)

Accuratezza rispetto ad una delle classi (*Tall*)

Actual Membership	Assignment Short	Medium	Tall
Short	0	4	0 (FP)
Medium	0	5	3 (FP)
Tall	0 (FN)	1 (FN)	2 (TP)

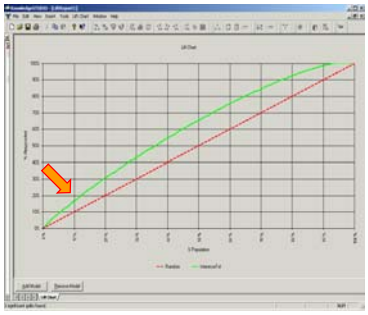


Lift report

- Alcuni modelli associano ad ogni classificazione un grado di confidenza
  - Ordiniamo gli esempi del test set in base alla stima della probabilità di appartenere ad una determinata classe A
- Lift report per la classe A
  - Sull'asse delle X consideriamo la % (rispetto al test set) di esempi secondo l'ordine precedente
  - Sull'asse delle Y consideriamo la % (rispetto al test set) degli esempi in X che hanno classe reale A
  - bisettrice rappresenta la performance del classificatore random
- Uso
  - Stima del numero di esempi da selezionare nello score set per ottenere una determinata percentuale di esempi con classe A

Lift report

- **Con il 20% degli esempi si cattura il 30% di tutti coloro che hanno classe 'interesse alto'**
- **Lift di 1.5 rispetto al classificatore random**



Modelli di classificazione

K-NN
Alberi di decisione

Classificazione – Tipi di modelli

- Vari tipi di modelli di classificazione
  - Differiscono per il formalismo utilizzato per rappresentare la funzione di classificazione (Linguaggio delle ipotesi)
- Alcuni tipi di modelli di classificazione:
  - Basati sugli esempi (es. Nearest neighbor)
    - memorizzano tutti gli esempi del training set ed assegnano la classe ad un oggetto valutando la "somiglianza" con gli esempi memorizzati (la cui classe è nota)
  - Matematici (es. Reti Neurali Artificiali, SVM)
    - la funzione di classificazione è una funzione matematica, di cui si memorizzano i vari parametri

Classificazione – Tipi di modelli

- … alcuni tipi di modelli:
  - Statistici
    - memorizzano i parametri delle varie distribuzioni di probabilità relative alle classi ed agli attributi
      - → per classificare un generico oggetto si possono stimare le probabilità di appartenenza alle varie classi
    - es.: Naive Bayes
  - Logici
    - la funzione di classificazione è espressa mediante condizioni logiche sui valori degli attributi
    - es.: Alberi e Regole di Decisione

Instance-Based Learning

- Apprendimento:
  - il processo di apprendimento ("lazy") si riduce alla memorizzazione degli esempi del training set
- Predizione:
  - assegna ad un nuovo caso un valore dell'attributo target "simile" a quelli che hanno gli esempi "più vicini"
  - il valore è ottenuto con una funzione "di interpolazione"
- Variante:
  - Se la classe assegnata è sbagliata si può memorizzare il nuovo caso con la classe corretta
- Il target può essere discreto o numerico
  - numerico (regressione): assegna la media dei valori
  - discreta (classificazione): assegna il valore più frequente

k-Nearest Neighbor (k-NN)

- Vicinato di un nuovo caso
  - Ogni istanza corrisponde a un punto *n*-dimensionale
  - Il vicinato di una nuova istanza è costituito dalle *k* istanze punti del training set meno distanti
  - Dipende dalla funzione di distanza scelta (es. Euclidea)
- Distanza
  - Attributi nominali: *0* se i valori sono uguali, *1* altrimenti
  - Attributi numerici: differenza
  - Opportuno normalizzare i valori (in fase di preprocessing)
- Performance
  - Apprendimento veloce (il training set è solo memorizzato)
  - Predizione lenta (considera tutto il training set)
  - Accuratezza sensibile al rumore

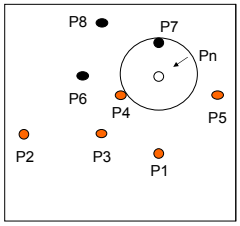
k-NN: Esempio di classificazione

- Dati 8 esempi di training
  - P1 (4, 2) =>Red
  - P2 (0.5, 2.5) =>Red
  - P3 (2.5, 2.5) =>Red
  - P4 (3, 3.5) =>Red
  - P5 (5.5, 3.5)=>Red
  - P6 (2, 4) =>Black
  - P7 (4, 5) =>Black
  - P8 (2.5, 5.5) =>Black
- nuovo caso :
  - Pn (4, 4) => ???

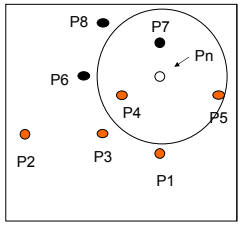
- Calcolo delle distanze:
  - d(P1, Pn) = $\sqrt{(4-4)^2+(2-4)^2}=2$
  - d(P2, Pn) = 3.80
  - d(P3, Pn) = 2.12
  - d(P4, Pn) = 1.12
  - d(P5, Pn) = 1.58
  - d(P6, Pn) = 2
  - d(P7, Pn) = 1
  - d(P8, Pn) = 2.12
- 1-NN = P7
  - Pn(4, 4) => **Black**
- 3-NN = P7, P4, P5
  - Pn(4, 4) => **Red**

k-NN: Esempio di classificazione

1-Nearest Neighbor

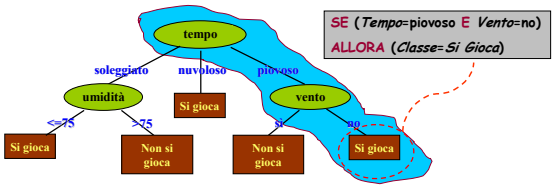


3-Nearest Neighbor



Alberi di decisione

Alberi di decisione: definizione

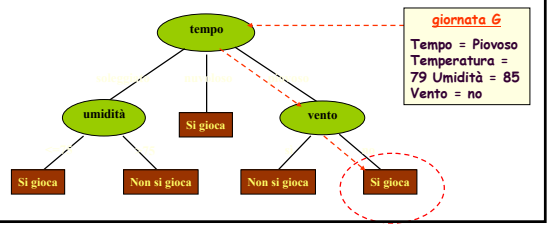


- ogni nodo interno *n* è associato ad un attributo *A(n)*
- ogni arco uscente da *n* è associato ad un insieme di valori di *A*
- ogni foglia è etichettata con il valore atteso della classe per gli oggetti descritti dal cammino che collega la foglia alla radice

→ **Un albero di decisione equivale ad un insieme di regole logiche mutuamente esclusive (una regola per ogni foglia)**

Alberi di decisione: uso

- Predizione (classificazione):
  - applicazione della funzione di classificazione ad un nuovo oggetto
- Esempio:
  - l'albero assegna la classe *Si Gioca* alla giornata G



Induzione di alberi di decisione

Algoritmi sviluppati indipendentemente negli anni 60 e 70 da ricercatori di varie aree:

- Statistica:
  - L. Breiman & J. Friedman - CART (Classification and Regression Trees)
- Pattern Recognition:
  - Univ. Michigan - AID
  - G.V. Kass - CHAID (Chi-squared Automated Interaction Detection)
- Intelligenza Artificiale e Teoria dell'Informazione:
  - R. Quinlan - ID3 (Iterative Dichotomizer)
  - R. Quinlan - C4.5

Indurre un albero di decisione

- Dati:
  - Un insieme di esempi con classe nota
- Problema:
  - Generare un albero di decisione per separare le classi un errore minimale
- Approccio:
  - produrre il modello più semplice che sia consistente con gli esempi di training
  - Formalmente: Trovare l'albero più semplice in assoluto è un problema intrattabile

Come produrre un albero ottimale?

- Euristica:
  - sviluppa l'albero in maniera top-down
  - piazza la variabile di test più importante alla radice del sottoalbero corrente (strategia greedy)
- La variabile più importante:
  - la variabile che ha la maggiore attendibilità nel distinguere l'insieme degli esempi di training
  - la variabile che ha la relazione più significativa con l'attributo target

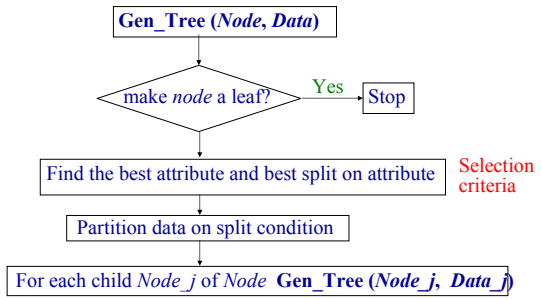
Schema per l'induzione di alberi

top-dow, per divisioni (split) successive, mediante condizioni logiche (test) su un attributo per volta:

1. Crea il nodo radice e assegnagli tutto il training-set
2. Per il nodo corrente:
   a) se tutti gli esempi del nodo hanno la stessa classe C, etichetta il nodo (foglia) con C e *stop*
   b) per ogni possibile test valuta una misura di "rilevanza" (*indice di split*) sui gruppi ottenuti
   c) associa al nodo il test che massimizza l'indice
   d) per ogni valore del test crea un nodo figlio, etichetta l'arco con la condizione di split ed assegna al nodo gli esempi che la soddisfano
3. Per ogni nuovo nodo creato ripeti il passo 2

Schema per l'induzione di alberi

(usato, ad es., in *ID3* e *J48*)



Indice di split

- Misura la qualità di una partizione (splitting)
  - sulla base della omogeneità dei gruppi rispetto alle classi
- Esempio: Guadagno Informativo
  - Dato un insieme S, p classi ed un attributo A con K valori
  - il Guadagno Informativo della partizione {S1, … , SK} di S, ottenuta in base ai valori di A è pari alla corrispondente diminuzione dell'entropia:

$$\textbf{Gain (S,A) = Entr(S) – (} |S_i| / |S| \textbf{)*} \Sigma_{i=1..K} \textbf{ Entr}(S_i)$$

- $\textbf{Entr(S) = -} \Sigma_{C=1..p} |S(c)| / |S| \textbf{ X } \log_2(|S(c)| / |S|)$
  - dove S (c) indica l'insieme degli elementi di S con classe c
  - indica l'Entropia Informativa dell'insieme S e rappresenta una misura della disomogeneità di S rispetto alle classi
  - è massima se le classi sono distribuite uniformemente in S

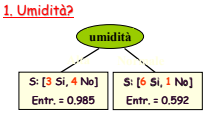
Induzione di alberi: esempio

Training set ***T***: 14 esempi [9 Si, 5 No]
Come partizionarlo? quale attributo scegliere per la radice? →

S : [9 Si, 5 No]
Entropia = 0.94

Tempo	Temp.	Umid.	Vento	Classe
Sol.	Alta	Alta	No	No
Sol.	Alta	Alta	Si	No
Nuv.	Alta	Alta	No	Si
Piov.	Media	Alta	No	Si
Piov.	Bassa	Norm.	No	Si
Piov.	Bassa	Norm.	Si	No
Nuv.	Bassa	Norm.	Si	Si
Sol.	Media	Alta	No	No
Sol.	Bassa	Norm.	No	Si
Piov.	Media	Norm.	No	Si
Sol.	Media	Norm.	Si	Si
Nuv.	Media	Alta	Si	Si
Nuv.	Alta	Norm.	No	Si
Piov.	Media	Alta	Si	No

1. Umidità?



Gain (S,Umidità)
= 0.94
– (7/14) * 0.985
– (7/14) * 0.592
= 0.151

2. Vento?

Gain (S,Vento)
= 0.94
– (8/14) * 0.811
– (6/14) * 1.000
= 0.048

Sull'intero training set *T Umidità* permette di discriminare le classi "meglio" di *Vento*: *Gain* (*T,Umidità*)=0.151 > *Gain* (*T,Vento*)=0.048

Modifiche all'indice di split

- Il *guadagno informativo* come indice di split tende a favorire attributi con molti valori
  - se un attributo K ha un valore distinto per ogni oggetto di S, la partizione di S con i valori di K ha entropia nulla;
  - infatti, ogni sottoinsieme Si della partizione è costituito da oggetti di una sola classe (contiene un unico elemento)
- *Gain Ratio* per compensare questo fenomeno

$$\text{GainRatio (S,A)} = \text{Gain (S,A)} / \text{SplitInfo(S,A)}$$

- *SplitInfo(S,A):*
  - misura l'informazione dovuta alla partizione di *S* in base ai valori di A:

$$\text{SplitInfo (S,A)} = - \Sigma_{i=1..K} |S_i|/|S| \text{ X } \log_2(|S_i|/|S|)$$

Split su attributi continui (numerici)

- Con un attributo "continuo" si può generare un numero "infinito" di partizioni
  - i comuni algoritmi considerano solo split binari
- Generare partizioni su un attributo continuo A:
  - ordinamento degli oggetti sulla base dei valori di A
  - se nell'insieme sono presenti n valori di A, si valuta ognuna delle le n-1 partizioni binarie possibili
  - scelta una partizione binaria su A si etichettano gli archi uscenti dal nodo con le condizioni:A≤v ' e A>v '
    - dove v ' è il punto medio fra il valore massimo del primo sottoinsieme ed il valore minimo del secondo

Overfitting

- Overfitting: sovradattamento al training set
  - Il modello classifica bene gli esempi da cui è stato indotto ma non è capace di classificare correttamente altri oggetti del dominio (è troppo adattato agli esempi usati)
- Alberi troppo grandi sono:
  - poco comprensibili
  - spesso soffrono di overfitting

Fondamento "filosofico" → Rasoio di Ockham:

"Ipotesi troppo complicate sono poco probabili"

(le spiegazioni più semplici sono spesso quelle esatte)

Ricerca della "giusta dimensione"

- Arresto della crescita dell'albero
  - durante la fase di costruzione (pre-pruning) in base a un criterio di significatività degli split:
    - soglia minima per il numero di oggetti dei nodi generati
    - soglia minima per l'indice di split (riduzione disordine)
- Pruning (potatura):
  - dopo aver costruito l'albero si eliminano i sotto-alberi che non contribuiscono all'accuratezza sul dominio,
    - sostituendoli con una foglia (*subtree replacement*)
    - oppure con un loro sotto-albero (*subtree racing*)
  - L'accuratezza sul dominio degli alberi considerati nella fase di pruning può essere stimata riservando una porzione del training set per il test (*reduced error pruning*)

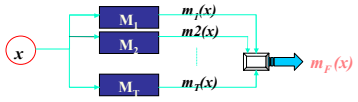
Come migliorare l'accuratezza

Come si può migliorare l'accuratezza di classificazione?

- Cambiare schema/parametri
- Modificare l'input (Data processing) :
  - Selezione
    - Selezionare altri insiemi di tuple o attributi (es.,feature selection)
  - Data cleaning
    - Sostituire valori mancanti, rimuovere outliers, correggere errori,…
  - Data transformation
    - Normalizzare valori, discretizzare attributi numerici, transformare attributi nominali/numerici in boolean, …
  - …
- Modificare l'output:
  - Meta-classificazione (Bagging, Boosting): il modello è una combinazione di più modelli di base
  - …

Meta-learning

- Apprendimento del modello (composto da più modelli)
  - Si qualche algoritmo di induzione di base (base learner) per costruire i modelli componenti
  - Ogni modello è indotto sul training set iniziale o su un dataset da esso derivato
- (meta-)Classificazione
  - Il classificatore risultante usa predice la classe sulla base delle predizioni dei classificatori che lo costituiscono (*M1*,…, *MT*)



Questa strategia si può applicare anche ai modelli di regressione

Meta-learning (1): *Bagging*

- Costruzione del modello (meta-apprendimento):
  - Dato un insieme di *n* istanze di esempio
    es.(n=6): **{D1, D2, D3, D4, D5, D6}**
  - Ad ogni passo:
    - Estrai *n* instanze senza rimuoverle dall'insieme
      es. : **{D3, D2, D3, D5, D3, D2}**
    - Usa le istanze estratte come training set ed applica l'algoritmo di apprendimento di base
    - Memorizza il modello appreso
- Classificazione:
  - Le predizioni dei singoli classificatori sono "mediate": si attribuisce la classe predetta con maggiore frequenza

Bagging: motivazioni

- L'errore atteso per il classificatore dipende da:
  - errore atteso del classificatore complessivo (*bias*)
  - errore dovuto al particolare training set usato (*varianza*)
- Il Bagging riduce l'influenza dell'errore del particolare training set scelto:
  - Le predizioni dei singoli classificatori (addestrati su diversi training set) sono "mediate"
  - All'aumentare del numero di classificatori si riduce la componente dell'errore dovuta al training set

Cos'è il *Bias* ?

- Indica l'influenza dell'algoritmo di learning usato sul modello che si può ottenere
- Dipende da:
  - Linguaggio delle ipotesi (formalismo per rappresentare il modello)
  - Strategia di ricerca del modello
  - Approccio per limitare l'overfitting
- Bias e accuratezza del modello
  - la componente di errore dovuta al bias può essere alta se il linguaggio delle ipotesi ha potere espressivo limitato
  - oppure se la strategia di ricerca è "troppo approssimativa"

Meta-learning (2): *Boosting*

- E' un altro schema di meta-classificazione
  - combina le predizioni dei singoli modelli pesandole in base alla loro accuratezza
- Procedura iterativa di apprendimento:
  - Costruisce una serie di modelli di classificazione
  - Nell'apprendimento di un nuovo modello dà un peso maggiore agli esempi classificati in modo scorretto dai classificatori generati in precedenza
    - I nuovi modelli sono influenzati dalle performance dei precedenti
- Risultato teorico:
  - Se I singoli classificatori hanno un'accuratezza "accettabile" (>50%) l'errore di training decresce esponenzialmente rispetto al numero di classificatori

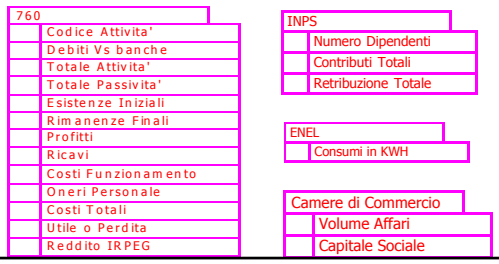
Boosting: algoritmo *AdaBoost.M1*

- Costruzione del modello :
  - Assegna stesso peso a tutte le istanze del training set
  - Ad ogni iterazione
    - Applica l'algoritmo di apprendimento di base al dataset pesato
    - Se l'errore del modello e>0.5 e *num_loop* < 25, riassegna lo stesso peso a tutte le istanze e ritorna all'inizio del ciclo
    - Per ogni istanza:
      - se è stata classificata correttamente moltiplicane il peso per $e/(1-e)$
    - Normalizza i pesi delle istanze
- Classificazione:
  - Per ogni modello:
    - assegna il peso – $log(e/(1-e))$ alla classe predetta dal modello
  - Restituisci la classe con peso massimo

Esempio di applicazione: analisi di dati fiscali

Contesto

Contesto: analisi di dati fiscali per scoperta di frodi
Task di mining: distinguere le due classi *evasori* e *non evasori*
Input: dati su ~15.000 Aziende di un settore



Problemi

- Difficoltà di disporre di grandi quantità di dato
- Scarsità di esempi per i casi fraudolenti
  - un modello con alta accuratezza (media) potrebbe non essere adeguato, se compie molti errori sulla classe dei fraudolenti (molto meno numerosa)
  - Alcuni degli esempi di *non evasori* potrebbero essere *evasori* non scoperti (su cui non si è fatto alcun accertamento)
- Variazione/adattamento del comportamento fraudolento
- Percentuale di frodi impredicibile

Esperimento: classificazione

- Obiettivo dell'analisi:
  - identificazione profilo contribuente
- Dati oggetto di analisi:
  - 14.000 contribuenti + 400 accertati
- Variabile target: evasore (si/no)

Matrice di Confusione

1	2	← Classificati come
TN	FP	Classe Effettiva 1
FN	TP	Classe Effettiva 2

TN (true negative) sono le tuple di classe 1 che vengono effettivamente classificate di classe 1, sono tuple che non subiranno controlli;

FP (false positive) sono le tuple di classe 1 che vengono misclassificate di classe 2. Queste tuple (in quanto classificate di classe 2) subiranno degli accertamenti che risulteranno essere non fruttuosi (RECUPERO <0) e costituiranno quindi per noi una fonte di spesa inutile.

Matrice di Confusione (Cont.)

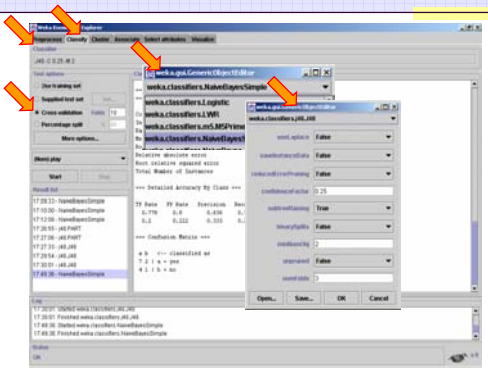
1	2	← Classificati come
TN	FP	Classe Effettiva 1
FN	TP	Classe Effettiva 2

- FN (false negative) sono le tuple di classe 2 che vengono misclassificate di classe 1. Queste tuple non subiranno dei controlli anche se sarebbero per noi fruttuosi. Sono gli evasori che perdiamo.

-TP (true positive) sono le tuple di classe 2 correttamente classificate di classe 2. Su queste tuple eseguiremo dei controlli fruttuosi. Sono la nostra fonte di recupero.

Classificazione con WEKA

WEKA



Istanze: rappresentazione

- ogni istanza è *una tupla* descritta da:
  - un insieme predefinito di proprietà (“attributi”)
- Possibili tipi:
  - **Categorici**: nominali, ordinali
  - **Numerici**: intervallo, rapporto

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Rappresentazione istanze: il formato ARFF

- **E’ possibile definire attributi *numerici* e *nominali***
- **La distanza fra valori nominali è**
  - ***0*** **se i valori sono uguali, *1* altrimenti**

@relation weather
@attribute outlook **{sunny, overcast, rainy}**
@attribute temperature **numeric**
@attribute humidity numeric
@attribute windy {true, false}
@attribute play? {yes, no}
@data
sunny, 85, 85, false, no
sunny, 80, 90, true, no
overcast, 83, 86, false, yes
...

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

WEKA: misure

- **True Positive (TP) rate**
  - **(# esempi di classe c classificati come c) / (# esempi di classe c)**
- **False Positive (FP) rate**
  - **(# esempi di classe non c classificati come c) / (# esempi di classe non c)**
- **Precision**
  - **(# esempi di classe c classificati come c) / (# esempi classif. come c)**
- **Recall (= TP rate)**
  - **(# esempi di classe c classificati come c) / (# esempi di classe c)**