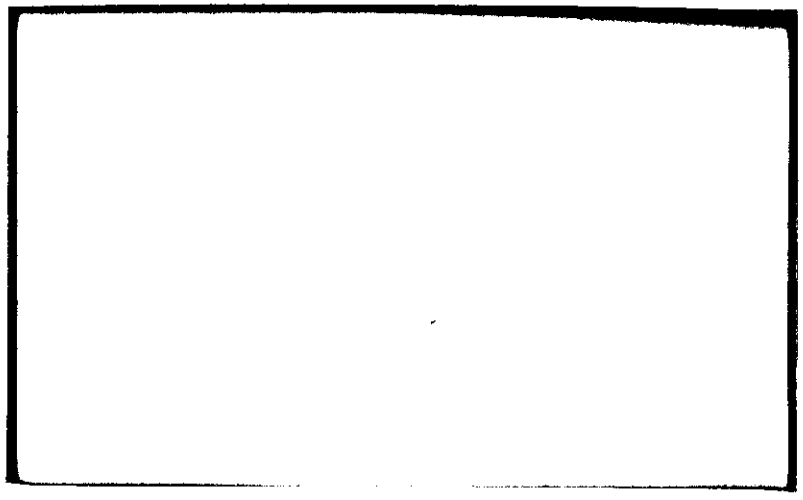

Rossella Miglio

Osservazioni sulla aggregazione di regole di classificazione

Serie Ricerche 2000, n.7

Dipartimento di Scienze Statistiche "Paolo Fortunati"
Università degli studi di Bologna

Il presente lavoro è stato svolto nell'ambito del progetto "Aspetti inferenziali nell'aggregazione di informazioni e nell'analisi statistica multidimensionale" finanziato con il contributo 1999 per le ricerche di interesse nazionale del Ministero dell'Università e della Ricerca Scientifica.

Finito di stampare nel mese di Luglio 2000
presso le Officine Grafiche Tecnoprint
Via del Legatore 3, Bologna

Indice

1 Introduzione Pag. 5

2 Alberi di classificazione e CART Pag. 5

3 Costruzione di un classificatore aggregato: *arcing*, *bagging* e *boosting*. Pag. 13

3.1 Uno studio di simulazione Pag. 20

4 Stabilità di una regola di classificazione ad albero e ricerca di una struttura di sintesi Pag. 24

4.1 Valutazione del consenso tra partizioni Pag. 26

4.2 Descrizione dell'algoritmo proposto. Pag. 29

4.3 Un esempio su dati simulati Pag. 30

5 Regole di classificazione ad albero e reti neurali Pag. 33

Riferimenti bibliografici Pag. 35

1. Introduzione

Un problema che negli ultimi anni ha ricevuto molta attenzione nella letteratura statistica riguarda l'instabilità delle regole di classificazione ottenute, per esempio, da classificatori ad albero o da reti neuronali artificiali; varie proposte sono state presentate al fine di aumentarne le capacità predittive combinando i risultati ottenuti da versioni multiple della stessa regola ottenute da replicazioni *bootstrap* del campione d'apprendimento (si vedano per esempio Breiman, 1994, Freund e Shapire, 1999, e Hand, 1996).

In questo lavoro si farà principalmente riferimento ai metodi di aggregazione utilizzati nel contesto degli alberi di classificazione; si introdurranno brevemente le caratteristiche di tali classificatori e dopo una breve rassegna dei principali contributi relativi all'argomento, si illustrerà un algoritmo sviluppato per la costruzione di un albero di classificazione che comporta un uso alternativo delle informazioni derivanti da alberi costruiti su diversi campioni generati con una procedura di *cross-validation*. Tale algoritmo si avvale di un procedimento iterativo basato su una misura di similarità tra partizioni. Esperimenti condotti su dati simulati mostrano come sia possibile determinare una nuova struttura di sintesi, meno complessa di quella determinabile con il *pruning* tradizionale, che pur garantisce la stessa accuratezza.

Un secondo aspetto trattato nell'ambito della aggregazione di regole di classificazione è quello relativo allo studio della sinergia tra alberi di classificazione e reti neurali. Alcune ricerche hanno evidenziato la possibilità di rappresentare la struttura degli alberi di classificazione mediante "reti neurali" multistrato (Chabanon *et al.* 1992, Miglio, Pillati, 1996), superando in tal modo alcuni dei limiti connaturati ai primi. Un'ulteriore proposta in tale ambito è relativa all'uso congiunto di alberi di classificazioni e reti con funzione a base radiale (Miglio e Pillati 2000)

2. Alberi di classificazione e CART

Lo sviluppo delle tecniche di segmentazione per la costruzione di

alberi binari - note anche come tecniche di partizione ricorsiva - risale ai lavori dei primi anni '60 di Morgan e Sonquist (1963), ed è a questi autori che si deve l'ideazione dell'algoritmo AID (Automatic Interaction Detection). Questi metodi avevano come obiettivo principale la previsione del valor medio condizionato di una variabile continua.

Negli anni successivi furono introdotte alcune varianti ed estensioni quali THAID (Messenger e Mandell, 1972), MAID-M (Gillo e Shelly, 1974) e CHAID (Kass, 1980). Ma è al contributo di Breiman, Friedman, Olshen e Stone (1984) che si deve il rinnovato interesse verso tali metodologie. Gli autori introducono la distinzione tra alberi di regressione (che hanno lo scopo di predire una variabile continua) e alberi di classificazione, in cui la variabile risposta è categorica. Il maggiore contributo di tale impostazione è sicuramente da attribuire al processo di validazione dell'albero introdotto.

Altri autori hanno esteso tale metodologia ad applicazioni non standard: ad esempio alla predizione della variabile tempo di sopravvivenza per dati "censurati" (Segal, 1992; Le Blank e Crowley, 1992; Ciampi, Thiffault, Sagman, 1989) o alla formalizzazione dei modelli lineari generalizzati (Ciampi, 1991).

Si considerino n unità statistiche, appartenenti a K classi, su cui sono stati osservati p caratteri $X_1,X_2,X_3,...,X_p$, che possono essere sia quantitativi sia qualitativi. Si indichi con Y la variabile che identifica la classe di appartenenza e con $\boldsymbol{X}$ il vettore delle variabili osservate. Una regola di classificazione determina una partizione dello spazio dei predittori $\boldsymbol{X}$ negli insiemi disgiunti $A_1,A_2,A_3,....,A_K$ tali che :

$\bigcup_{i=1}^{K} A_i = \mathbf{X}$

E' quindi possibile rappresentare la regola di classificazione d nel modo seguente

$d(\boldsymbol{x}) : (\boldsymbol{x} \in A_i) \Rightarrow (Y=i)$

Un albero di classificazione è la rappresentazione di una regola di classificazione individuata da una procedura di tipo ricorsivo. L'insieme di unità statistiche osservate è ripartito in gruppi mediante una successione di divisioni (o segmentazioni) dicotomiche[1] (o binarie) di

[1] La divisione potrebbe tuttavia avere anche come risultato più di due gruppi; un esempio è rappresentato dall'algoritmo CHAID (Kass, 1980).

tipo gerarchico, analizzando una variabile predittiva alla volta. Il risultato finale di tale procedura consente di identificare simultaneamente gruppi omogenei di unità e le variabili esplicative aventi maggior potere discriminante.

Nel primo passo, l'insieme di unità, detto gruppo genitore, è suddiviso in due sottoinsiemi che ottimizzano una particolare funzione obiettivo. Le modalità di un predittore sono suddivise a tal fine in due categorie: le unità che possiedono una modalità della prima categoria costituiscono il primo sottogruppo; le rimanenti unità, il secondo. Ogni sottogruppo è denominato gruppo figlio.

Nel secondo passo si effettua la suddivisione dicotomica dei due gruppi figli generati al passo precedente sulla base del medesimo criterio. La segmentazione è iterata fino a che non si avvera una delle condizioni di arresto del processo. Tale processo di bipartizione è rappresentabile mediante un albero binario T: i nodi che si diramano dal nodo radice costituito dall'intero campione rappresentano le singole suddivisioni e l'insieme dei nodi terminali $\tilde{T}$ (foglie) definisce gli elementi della partizione finale.

Una regola di classificazione rappresentata da un albero costituisce un modo interessante per sintetizzare l'informazione contenuta in un insieme di dati poiché definisce, attraverso una successione di domande sui predittori, piccoli gruppi di unità sui quali effettuare distinte predizioni. Dal punto di vista della interpretazione dei risultati, l'albero è preferibile ai metodi tradizionali, che trattano la popolazione come omogenea su tutto lo spazio dei predittori e gli effetti degli stessi come lineari. Gli alberi costituiscono una possibile soluzione per uscire dalla ipotesi restrittiva della linearità, in particolare nella direzione della interazione tra variabili.

A differenza di alcuni metodi di analisi discriminante, che per costruire la regola di classificazione richiedono che siano specificati modelli probabilistici sui predittori all'interno di ogni classe, gli alberi non impongono alcuna restrizione sulla distribuzione di riferimento e permettono di considerare congiuntamente predittori quantitativi e qualitativi. La regola di classificazione rappresentata dall'albero è di facile interpretazione e può approssimare direttamente il processo logico di decisione dell'esperto interessato allo specifico problema applicativo.

Il problema dei dati mancanti è risolto in CART[2] (Classification and Regression Trees, Breiman *et al.* 1984) mediante l'impiego di variabili supplementari.

In sintesi, le fasi di una procedura di segmentazione sono fondamentalmente tre:

i) scelta del criterio di suddivisione di ogni nodo t nei nodi figli t_s e t_d;

ii) individuazione di un criterio di arresto della procedura;

iii) attribuzione delle unità alle classi.

Ogni criterio di partizione si pone come obiettivo di individuare tra tutte le possibili suddivisioni dello spazio dei predittori quella che consente di migliorare la predizione della classe k, k=1,. .,K o della variabile Y. Questo equivale a definire un criterio per suddividere ciascun nodo in due sottogruppi il più possibile omogenei al loro interno, ovvero eterogenei tra loro, rispetto alla variabile risposta Y. Occorre, quindi, individuare tutte le possibili partizioni tra cui scegliere quella ottimale e i criteri sulla base dei quali effettuare tale scelta.

Ogni suddivisione coinvolge un predittore X_j, j=1,...,p o una combinazione lineare di predittori, ed il numero di suddivisioni realizzabile dipende dalla natura del predittore considerato. Le suddivisioni realizzate su singoli predittori sono in numero finito, poiché la costruzione della regola si basa su un campione di osservazioni.

Nella figura 1 sono riportati un esempio di albero ottenuto da un problema a due predittori continui e la partizione dello spazio dei predittori da questi generato.

[2] Salford system

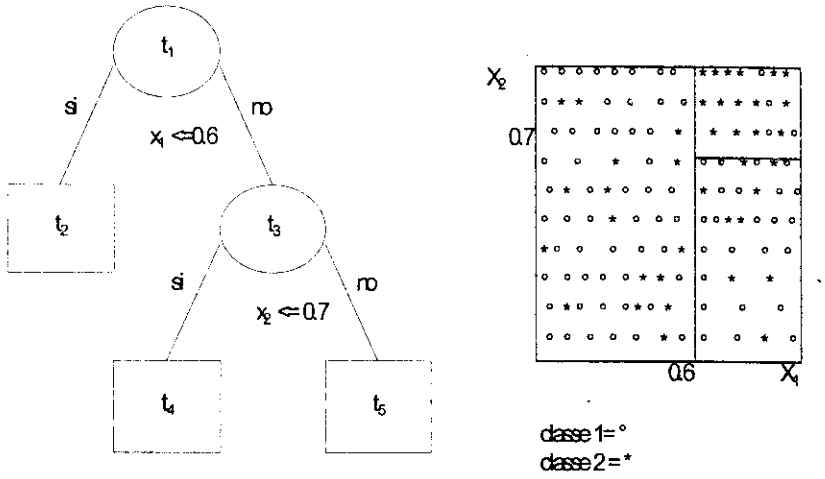

Figura 1. Esempio di albero di classificazione e corrispondente partizione dello spazio dei predittori da questi generato.

Individuate tutte le possibili suddivisioni realizzabili nel generico nodo *t* dell'albero, occorre un criterio che consenta di scegliere la suddivisione che separa al meglio le classi.

In letteratura sono state presentate varie proposte: criteri che fanno riferimento a misure di impurità (Breiman *et al.* 1984) o misure di informazione (Clark e Pregibon,1992).

Generalmente nella costruzione di una regola di classificazione l'obiettivo è di minimizzare l'errore complessivo di errata classificazione; qualora si intenda interpretare l'albero come criterio diagnostico, se pure complesso, occorre valutarne le potenzialità informative mediante indicatori quali la sensibilità e la specificità al fine di distinguere l'errore di classificazione commesso nella classe dei malati e dei sani, ovvero il complemento ad uno di sensibilità (*Se*) e specificità (*Sp*), rispettivamente. Una soluzione alternativa (Miglio, 1996,1997) può essere quella di considerare nella scelta della migliore suddivisione un indicatore che consenta di bilanciare i due tassi di errore, adottando per esempio l'indice di Youden(1950):

$J=1-(1-Se)-(1-Sp)=Se+Sp-1$.

Loh and Vanichsetakul (1988) hanno presentato una proposta per la scelta della migliore suddivisione basata su di un'applicazione ricorsiva della analisi discriminante lineare normale, mediante la quale è possibile fare riferimento a *split* non necessariamente paralleli agli assi.

Mola e Siciliano (1992) propongono un criterio a due stadi: nel primo si considera il contributo globale di un singolo predittore mediante la massimizzazione di una funzione di "predizione" della variabile dipendente; nel secondo stadio, per il predittore selezionato, si considerano tutte le possibili suddivisioni generabili dalle distinte modalità del predittore. Tale procedura consentirebbe di ridurre i tempi di calcolo, evitando di considerare tutti i possibili *split* su tutti i predittori osservati.

Un differente algoritmo proposto da Kass (1980) e denominato CHAID (CHi squared Automatic Interaction Detection) si basa sull'impiego del test di indipendenza tra due caratteri χ^2 ma differente è la procedura di costruzione dell'albero.

Altri indicatori sono stati proposti negli studi sull'intelligenza artificiale. Mingers (1989), mediante un confronto empirico tra diversi criteri di selezione, conclude che l'accuratezza dell'albero non è sensibile alla misura di selezione adottata; un successivo lavoro di Buntine e Niblett (1992) sembra confermare tali conclusioni. Tra le misure considerate nel lavoro di Mingers sono compresi l'indice di eterogeneità di Gini e le misure di entropia.

Individuato un criterio in base al quale scegliere la migliore suddivisione occorre definire un criterio di arresto della procedura di partizione ricorsiva, nell'impostazione di Breiman *et al.* (1984) si individua inizialmente un albero di grandi dimensioni con un criterio di arresto basato semplicemente sulla numerosità dei nodi finali, da questo ci si riconduce quindi mediante una procedura di sfolimento o *pruning* ad un sottoalbero di dimensione ridotta. Le unità che confluiscono in un particolare nodo terminale saranno assegnate alla classe che risulta maggiormente rappresentata nello stesso.

La capacità della regola di classificazione rappresentata dall'albero di generalizzare i risultati ad osservazioni campionarie diverse da quelle su cui la regola è stata costruita risulta gravemente compromessa dal problema dell'*overfitting,* comune a molti metodi di tipo esplorativo. Il

problema dell'*overfitting* è presente quando la regola di classificazione tende ad adattarsi eccessivamente al campione perdendo capacità di generalizzare oltre l'osservato. Questo comporta che il tasso di errore apparente ($R(T)$) presenta un andamento monotono decrescente al crescere delle dimensioni dell'albero, e valutazioni basate su tale tasso portano a proseguire nel processo di bipartizione.

Diverso invece è l'andamento osservabile con l'impiego di uno stimatore del tasso di errata classificazione con minore distorsione: si osserva prima un andamento decrescente e quindi crescente all'aumentare delle dimensioni dell'albero oltre un certo livello. Questo comportamento è la conseguenza di un bilanciamento tra la variabilità e la distorsione del classificatore offerto dall'albero.

Stabilire una regola di arresto basata sul raggiungimento di una soglia nel decremento della funzione di impurità è una strategia impropria. La novità fondamentale del lavoro di Breiman *et. al.* (1984) consiste nell'aver introdotto una procedura di validazione nella costruzione dell'albero e nella scelta di una procedura di potatura (*pruning*) di un albero di grandi dimensioni, in alternativa ad una regola di arresto.

Gli autori introducono un sofisticato meccanismo di *pruning*. CART cerca con il *pruning* di realizzare un compromesso tra due contrapposte esigenze: la necessità di ottenere una struttura semplice e nel contempo una stima accurata della vera probabilità di errata classificazione. Questi obiettivi sono perseguiti mediante l'impiego di una funzione di costo complessità. Sia T l'albero di classificazione utilizzato per classificare gli n elementi del *training set* C. Sia m la cardinalità dell'insieme degli elementi erroneamente classificati. Se indichiamo con $|T|$ il numero di foglie di T possiamo definire nel modo seguente la funzione di costo-complessità per un parametro α:

$$R_\alpha(T)=R(T)+\alpha\cdot|T| \quad \alpha\geq 0$$

dove $R(T)=m/n$ rappresenta la stima del tasso di errata classificazione ottenuta mediante il tasso di errore apparente. Tale funzione è quindi una combinazione lineare della stima dell'errore e di una penalità associata alla sua complessità. Se α è piccolo, la penalità associata ad un elevato numero di nodi terminali sarà bassa e quindi T sarà grande. Al variare di

si bilanciano in misura differente la necessità di avere un classificatore semplice e l'accuratezza dello stesso, sempre da intendersi in termini di predittività dei nodi stessi.

Il procedimento di *pruning* si sviluppa in due passi.

Nel primo passo mediante una procedura di potatura selettiva, si determina il miglior sottoalbero tra tutti i sottoalberi aventi le stesse dimensioni individuando una sequenza di sottoalberi annidati:

$T_0 \supset T_1 \supset T_2 \supset \ldots \supset \{t_1\}$

dove t_1 rappresenta l'albero costituito dal nodo radice, e ogni sottoalbero è ottenuto potando l'albero che immediatamente lo precede nella sequenza. La scelta di T_{i+1} da T_i è effettuata tra tutti i sottoalberi ottenuti potando T_i in ogni nodo non terminale, scegliendo tra questi il sottoalbero a cui corrisponde il minor valore di α. Se esistono più sottoalberi a cui è associato tale valore di α allora il nuovo sottoalbero sarà ottenuto potando entrambi i rami. La sequenza di sottoalberi individuata sarà costituita da alberi annidati cui corrispondono valori di α crescenti. Generalmente nelle fasi iniziali i rami potati saranno di dimensioni elevate, tenderanno quindi a ridursi nel seguito della potatura.

La seconda fase della procedura di *pruning* consiste nello scegliere all'interno della sequenza così individuata il miglior sottoalbero sulla base di una stima, $R^*(T)$, non distorta del tasso di errata classificazione associato al classificatore rappresentato dall'albero. Generalmente si fa riferimento a due possibili alternative: l'impiego di un campione di *test* e la *V-fold cross-validation*. Nel primo metodo Nel primo metodo le n unità statistiche vengono suddivise in due insiemi L_1 ed L_2. Le osservazioni di L_1 vengono utilizzate per la costruzione del classificatore, mentre le unità L_2 per stimare il tasso di errata classificazione. Generalmente la numerosità di L_2 è pari a 1/3 della numerosità campionaria complessiva. Avvalersi di un *test set* è dal punto di vista computazionale efficiente, ma presenta lo svantaggio di ridurre considerevolmente la dimensione del campione su cui la regola di classificazione viene costruita pertanto per campioni di dimensione ridotta si tende a preferire la *V-fold cross validation*. La costruzione del classificatore ad albero si avvale, in tal caso, dell'intero campione: le unità statistiche vengono suddivise, tramite procedura di estrazione

casuale, in V sottocampioni L_v $v=1,\ldots,V$, di pari dimensione. Il v-esimo *learning sample* è costituito da:

$$L^{(v)}=L-L_v \quad v=1,\ldots V$$

e contiene la frazione $(V-1)/V$ del totale delle osservazioni. Generalmente si sceglie $V=10$; in tal modo ciascun *learning sample* contiene i 9/10 delle unità totali. Ciascuno dei sottoinsiemi $L^{(v)}$ viene utilizzato per la costruzione di una regola di classificazione ad albero e le unità L_v per una stima $R(d^{(v)})$ del tasso di errata classificazione associato alla regola di classificazione $d^{(v)}$. La stima *cross validation* di R(d) è data dalla medie delle stime $R(d^{(v)})$. Tale metodologia, maggiormente onerosa dal punto di vista computazionale, consente di utilizzare tutte le osservazioni in maniera più efficiente ed offre informazioni circa la stabilità della struttura dell'albero.

3. Costruzione di un classificatore aggregato: *arcing*, *bagging* e *boosting*.

Uno dei problemi principali che caratterizza gli alberi, così come le reti neurali, più volte messo in evidenza in letteratura, è che entrambi i metodi generano classificatori instabili, ovvero classificatori sensibili a variazioni anche modeste del *training set* o, nel caso delle reti neurali, del processo di apprendimento. Tali metodi presentano, in generale, una distorsione contenuta e un'elevata varianza. Diverse sono le soluzioni proposte per individuare una struttura che realizzi un compromesso tra la flessibilità del classificatore e il pericolo dell'*overfitting*.

Nel seguito si analizzeranno, brevemente, quelle che non si propongono di individuare un unico modello, ma determinano la regola di classificazione mediante la sintesi di diversi classificatori.

La proposta di combinare diversi classificatori o diverse versioni degli stessi, riprende in realtà un concetto già presente nel contesto della regressione, e cioè l'idea di prevedere i valori di una variabile combinando tra loro le previsioni ottenute da modelli diversi.

L'idea di fondo è legata all'implicita assunzione che difficilmente si è in grado di individuare il processo che ha generato le osservazioni, cioè il modello "vero", ma che l'impiego di diversi modelli può permettere proprio di cogliere diversi aspetti dell'informazione contenuta nei dati.

I metodi di aggregazione proposti in letteratura si propongono di migliorare l'accuratezza del classificatore rappresentato da un albero di classificazione costruendo ed aggregando versioni multiple dell'albero ottenute a partire delle sole informazioni presenti nel campione di apprendimento L.

La principale differenza che caratterizza i diversi algoritmi dipende dal differente campionamento adottato. Le perturbazioni nel campione di apprendimento introdotte dal *bagging* (Breiman, 1994) sono casuali ed indipendenti, mentre le perturbazioni introdotte dai metodi *boosting* (Freund e Shapire, 1999) ed *arcing* (Breiman, 1996) sono di tipo sequenziale, in quanto la i-esima perturbazione dipende dalle regole precedentemente generate.

Il metodo di aggregazione *bagging* costituisce una delle prime proposte avanzate per superare il problema della instabilità delle regole di classificazione ad albero.

Il presupposto su cui tale metodo si basa è quello di poter estrarre dalla medesima popolazione da cui proviene il campione di apprendimento una successione di campioni L_w $w=1,\ldots,W$, di dimensione n, indipendenti da L, a cui si associa una successione di classificatori $d(\mathrm{x}, L_w)$. Tali classificatori possono essere aggregati mediante un criterio di "votazione" (*voting*): dato un vettore di predittori **x**, la classe assegnata ad y sarà quella maggiormente prescelta dall'insieme dei classificatori.

Definito

$$v_j=\#\{\mathrm{w};d(\mathrm{x}, L_w)=\mathrm{j}\}$$

la regola di classificazione aggregata può essere formalizzata come segue:

$$d(x) = \underset{\mathrm{j}}{argmax}\ v_j$$

Nella realtà poiché si dispone di un unico campione di apprendimento, la molteplicità dei campioni necessaria per costruire il classificatore

aggregato è ottenuta estraendo dal campione iniziale W campioni *bootstrap* di dimensione n. Tale procedimento conduce alla costruzione di uno pseudo universo campionario, di elementi $L_W{}^b$, con cui si costruiscono i W classificatori $d(\mathrm{x}, L_w)$. Ciascuna osservazione è assegnata alla classe più votata dai W classificatori da cui l'acronimo *bagging*, da *bootstrap aggregating*, che identifica tale procedura di aggregazione.

I metodi di aggregazione agiscono sulla variabilità dei classificatori instabili lasciando sostanzialmente invariata la componente del tasso di errata classificazione dovuta alla distorsione del classificatore.

Al fine di valutare le *performance* di tale metodo, sono stati presi in considerazione in letteratura diversi data set reali e simulati evidenziando rispetto alla procedura CART standard riduzioni del tasso di errata classificazione, valutato su campioni indipendenti, che vanno dal 20 al 47%. Chiaramente il guadagno in accuratezza dipende in modo cruciale dall'instabilità della regola di classificazione: il *bagging* può migliorare l'accuratezza se le perturbazioni indotte nel campione di apprendimento mediante il *bootstrap* determinano cambiamenti rilevanti nell'albero costruito, al contrario, se l'aggregazione riguarda regole stabili il predittore aggregato non si discosterà significativamente dalla regola singola originaria e pertanto il guadagno in accuratezza sarà minimo.

Le varianti di tale metodo proposte in letteratura riguardano fondamentalmente la costruzione della successione dei campioni estratti dal campione originario. Il campionamento adottato non è più un campionamento casuale semplice bensì di tipo "adattivo", ad ogni stadio del processo sequenziale di costruzione dei campioni, e quindi dei classificatori, le probabilità di estrazione associate alle singole osservazioni, costituenti il campione di apprendimento, aumentano ad ogni passo per le osservazioni erroneamente classificate, in tal modo l'algoritmo è costretto a concentrarsi sulle unità problematiche del *training set*.

La selezione dei diversi campioni su cui costruire i classificatori da aggregare non è effettuata generando W campioni indipendenti ma assegnando un peso λ_i^w alla i-esima osservazione nel w-esimo passo.

Uno di tali algoritmi, il *boosting* si articola nei seguenti passi:

1. il vettore dei pesi è inizializzato ponendo il peso di ciascuna osservazione pari a 1/n

2. $\forall$ ripetizione w=1,...,W
   i) ciascun peso λ_i^w rappresenta la probabilità di estrazione[3] associata alla i-esima osservazione nella w-esima ripetizione dell'algoritmo, si estrae da L un campione di apprendimento L^w di dimensione n da cui si costruisce un albero di classificazione, tale procedura equivale a determinare il classificatore d^w sotto la distribuzione λ^w;
   ii) si determina il tasso di errore ε^w del w-esimo classificatore mediante la somma dei pesi delle osservazioni erroneamente classificate:
   $$\varepsilon^w = \sum_i \lambda_i^w I\left(d^w(\mathbf{x}_i) \neq y_i\right)$$
   dove I() è una funzione indicatrice che assume valore 1 se la regola di classificazione non individua correttamente la classe di appartenenza;
   iii) se ε^w >0,5, la replicazione ha termine e si pone W=w-1; se il classificatore d_w classifica correttamente tutte le unità, W è posto pari a w. Diversamente si procede passando alla replicazione successiva: il vettore dei pesi λ^{w+1} è aggiornato moltiplicando i pesi delle osservazioni erroneamente classificate da d^w per il fattore $\beta^w = (1-\varepsilon^w)/\varepsilon^w$ e quindi normalizzando il vettore dei pesi;
3. il classificatore finale è ottenuto mediante una votazione ponderata dei risultati dei classificatori $d_1,d_2,\ldots,d_W$, dove il peso associato al w-esimo classificatore è pari a $\log(\beta^w)$ ed è pertanto funzione della sua accuratezza.

Come per il metodo *bagging*, il successo del *boosting* nel ridurre la varianza del classificatore dipende implicitamente dalla instabilità del metodo di classificazione: se il classificatore d_w non si discosta dal classificatore, ottenuto al passo precedente d_{w-1}, il sistema di aggiustamento dei pesi fa sì che i classificatori successivi a d_w avranno

[3] Nella formulazione originaria i λ_i^w possono anche rappresentare dei pesi associati alle singole unità, tutte incluse nella costruzione dei classificatori.

pesi pressoché nulli nel classificatore finale. Un inconveniente associato a tale metodo si presenta quando l'algoritmo iniziale produce un classificatore in totale accordo con i dati campionari, in tale situazione la procedura si arresta al primo passo.

In numerose applicazioni il metodo *boosting* ha mostrato un maggior successo nella riduzione della varianza rispetto al *bagging*. Una variante di questo, che ne costituisce una versione semplificata, è stata proposta da Breiman (1996) che conia per tali metodi l'acronimo *arcing* (*adaptively resampling and combining*) con l'intento di sottolinearne la base comune, vale a dire il procedimento di ricampionamento di tipo adattivo che genera il predittore aggregato.

L'algoritmo *arcing* si compone dei seguenti passi:

1. il vettore dei pesi è inizializzato ponendo il peso di ciascuna osservazione pari a 1/n
2. $\forall$ ripetizione w=1,...,W
   i) ciascun peso λ_i^w rappresenta la probabilità di estrazione associata alla i-esima osservazione nella w-esima ripetizione dell'algoritmo, si estrae da L un campione di apprendimento L^w di dimensione n su cui è costruito un albero di classificazione
   ii) si determina il tasso di errore ε^w come il numero di volte in cui una unità è stata erroneamente classificata dagli alberi $d_1,d_2,\ldots d_w$, il vettore dei pesi è quindi aggiornato:
   $$\lambda_i^{w+1} = \frac{1+\left(\varepsilon_i^w\right)^4}{\sum_i 1+\left(\varepsilon_i^w\right)^4}$$
3. il classificatore finale è ottenuto mediante il *voting* dei classificatori $d_1,d_2,\ldots,d_W$.

Una interessante caratteristica del metodo boosting è quella di poter essere riletto come un particolare modello statistico di tipo additivo poiché presenta delle analogie con la classe dei modelli MART (*Multiple Additive Regression Trees*) recentemente introdotti da Friedman (1999a, 1999b). Tali modelli vengono utilizzati per studiare la relazione tra una variabile aleatoria Y e un vettore p-dimensionale $\mathbf{X}=(X_1,X_2,\ldots X_p)\subset D\in R^p$ di variabili esplicative o predittori, sulla

base di un insieme di n osservazioni $\{(y_i, \mathbf{x}_i);\ i = 1,2,...n\}$. Se l'oggetto d'interesse è costituito da:

$$E[Y \mid \mathbf{X}] = f(X_1, X_2, ...X_p) = f(\mathbf{X})$$

si dovrà ricercare una funzione $\hat{f}(\mathbf{x})$ che approssimi in modo accurato $f(\mathbf{X})$ nel dominio d'interesse D, definita una opportuna funzione di perdita $L(Y, f)$.

La classe di funzioni additive considerate è del tipo seguente:

$$\hat{f}(\mathbf{x}) = \sum_{w=1}^{W} \beta_w h(\mathbf{x}; \mathbf{a}_w)$$

dove $h(\mathbf{x};\mathbf{a}_w)$ è una funzione con una struttura semplice (*base learner*), che può essere rappresentata anche da un albero di regressione di dimensioni contenute. Il vettore $\mathbf{a}_w$ risulta quindi essere costituito dalle variabili coinvolte nel processo di costruzione dell'albero, dai relativi valori coinvolti nelle suddivisioni e dalla media della variabile nei nodi dell'albero.

La stima dei parametri β_w e degli elementi di $\mathbf{a}_w$ non viene realizzato simultaneamente ma mediante una procedura a stadi in cui il parametro β_w e gli elementi $\mathbf{a}_w$ caratterizzanti la funzione h vengono individuati minimizzando la funzione di perdita sui residui dello stadio precedente denominati pseudo-residui. L'algoritmo può essere quindi sintetizzato come segue:

1. si individua $\hat{f}_0(\mathbf{x})$: $\hat{f}_0(\mathbf{x}) = \underset{y'}{argmin} \frac{1}{n} \sum_{i=1}^{n} L(y_i; y')$
2. $\forall\ w = 1, \ldots, W$

   si determina $(\beta_w, \mathbf{a}_w) = \underset{\beta, \mathbf{a}}{argmin} \frac{1}{n} \sum_{i=1}^{n} L(y_i; f_{w-1}(\mathbf{x}) + \beta h(\mathbf{x}; \mathbf{a}))$

   mediante un processo a due stadi:

   a. si stimano con il criterio dei minimi quadrati gli elementi $\mathbf{a}_w$:

$$\mathbf{a}_w = \underset{\mathbf{a}, \rho}{argmin} \sum_{i=1}^{n} [\tilde{y}_{iw} - \rho h(\mathbf{x}_i; \mathbf{a})]^2$$

sugli "pseudo residui" definiti come:

$$\tilde{y}_{iw} = -\left[\frac{\partial L(y_i; f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)} \right]_{f(\mathbf{x}) = f_{w-1}(\mathbf{x})}$$

   b. data $h(\mathbf{x};\mathbf{a}_w)$ si determina il valore ottimale del coefficiente β_w:

$$\beta_w = \underset{\beta}{argmin} \sum_{i=1}^{n} L(y_i; f_{w-1}(\mathbf{x}_i) + \beta h(\mathbf{x}_i; \mathbf{a}_w))$$

   ed infine

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_m h(\mathbf{x}; \mathbf{a}_m)$$

3. $\hat{f}(\mathbf{x}) = f_W(\mathbf{x}) = \sum_{w=1}^{W} \beta_w h(\mathbf{x}; \mathbf{a}_w)$

L'approssimazione di funzioni è vista come un problema di ottimizzazione numerica nello spazio delle funzioni piuttosto che nello spazio dei parametri e presenta delle analogie con il metodo di minimizzazione del tipo *steepest descent*.

Friedman ha quindi mostrato come l'accuratezza di tale algoritmo e la velocità dello stesso in termini di tempi di elaborazione possano essere migliorati introducendo ad ogni stadio una componente di aleatorietà. In particolare ad ogni stadio il processo di minimizzazione della funzione di perdita viene applicato su di un sottocampione dell'intero campione. Non è ancora chiaro il ruolo giocato da tale procedura di randomizzazione. Sicuramente le migliori performance osservate sono da ascrivere alla individuazione di modelli dotati di minore varianza. L'uso di campioni di dimensione ridotta in ciascuno stadio comporta una maggiore variabilità nel *base learner*; ma comporta altresì una minore correlazione tra le stime ottenute nei diversi passi. Ne discende quindi che il modello combinato in quanto media dei *base learner* presenta una ridotta varianza.

In riferimento a problemi di classificazione è la probabilità π_j di osservare un valore di classe c_j, o una trasformata di questa che dovrà essere analizzata in funzione del vettore p-dimensionale $\boldsymbol{X}$. Qualora si tratti di un problema a due classi, tale procedura determina una estensione del modello di regressione logistica:

$$\pi(\mathbf{x}) = \frac{1}{1+\exp(-2f(\mathbf{x}))}$$

e la funzione di perdita è costituita dalla misura di devianza, ovvero il doppio dell'opposto della log-verosimiglianza:

$$L(y; \hat{f}(\mathbf{x})) = 2\ln(1+\exp(-2y\hat{f}(\mathbf{x}))$$

3.1 Uno studio di simulazione

Lo studio di simulazione che segue è stato realizzato al fine di confrontare *bagging* e *arcing* non solo dal punto di vista del tasso di errata classificazione ma anche disaggregando i risultati rispetto alle diverse classi con l'obiettivo di valutare quale dei due metodi presenti le migliori caratteristiche soprattutto in situazioni particolari quali quelle dell'esempio prescelto.

Il data set utilizzato è presente nel sito internet[4] del Department of Information and Computer Science dell'Università della California.

Il problema indagato è di tipo diagnostico; le otto variabili rilevate fanno riferimento a caratteri biodemografici relativi a 768 donne di età superiore ai 21 anni appartenenti alla tribù indiana dei Pima (popolazione vivente vicino a Phoenix in Arizona). La variabile che identifica la classe assume quindi due valori che distinguono le donne affette e non affette da diabete.

Il data set iniziale è stato opportunamente suddiviso in un campione di apprendimento con il quale si è costruito l'albero di classificazione standard ed i classificatori aggregati con il *bagging* e l'*arcing* ed un campione di test su cui sono si sono valutate le regole costruite. La dimensione del campione di apprendimento è stata scelta pari al 90% della numerosità campionaria complessiva.

Il classificatore combinato è stato ottenuto aggregando 50 classificatori. Il tasso di errata classificazione è stato stimato sulla base del campione di

[4] ftp.ics.uci.edu/pub/machine-learning-databases

test replicando 100 volte la procedura, i risultati riportati costituiscono quindi la sintesi di tali replicazioni.

Tabella 1. *Risultati medi ottenuti da 100 replicazioni.*

	Tasso di errata classificazione	Riduzione % rispetto a CART
CART	28,3	
BAGGING	23,8	15,9
ARCING h=3	24,7	12,7
ARCING h=4	25,5	9,9
ARCING h=5	26,4	6,7
ARCING h=6	25,1	11,3

Tabella 2. *Risultati medi ottenuti da 100 replicazioni distinti per classe.*

	1-Sensibilità	1-Specificità
CART	21,6	41,4
BAGGING	14,9	40,5
ARCING h=3	15,8	42,1
ARCING h=4	15,1	44,7
ARCING h=5	16,1	44,7
ARCING h=6	14,8	44,5

In relazione a tali dati come appare in tabella 1 il metodo *bagging* presenta migliori perfomance rispetto alle altre procedure pur mantenendo un tasso di errata classificazione che si assesta su valori decisamente elevati. La tabella 2 evidenzia i risultati ottenuti distinguendo per la classe di appartenenza delle osservazioni. Nel gruppo dei sani, il valore minimo del tasso di errata classificazione (ovvero 1-specificità) per il metodo *arcing* si ottiene in corrispondenza di h=6 ma tale valore non si discosta notevolmente dal valore ottenuto con il più semplice *bagging*. Mentre decisamente a favore di quest'ultimo propendono i risultati relativi al gruppo delle donne malate; alla luce di tali risultati disaggregati sembra

quindi preferibile. Si noti tuttavia come l'algoritmo presenti una elevata difficoltà nel classificare proprio la classe di maggiore interesse.

Tabella 3. *Risultati medi ottenuti da 100 replicazioni in relazione a differenti scelte delle probabilità a priori.*

π(1)	π(2)	CART standard	BAGGING	ARCING h=4
0,1	0,9	35,6	28,5	26,4
0,2	0,8	34,9	27,5	25,4
0,3	0,7	34,2	27,3	26,7
0,4	0,6	30,6	25,5	25,9
0,5	0,5	29,3	25,1	25,0
0,6	0,4	27,8	24,3	24,2
0,651	0,349	28,4	23,9	25,6
0,7	0,3	28,6	24,5	25,7
0,8	0,2	27,3	24,6	25,4
0,9	0,1	28,2	25,4	24,8

Una ulteriore analisi è stata condotta utilizzando diversi valori per le probabilità a priori delle due classi considerate. Dall'analisi della tabella 3 emergono alcune interessanti considerazioni. La procedura CART è la più sensibile alle variazioni sulle probabilità a priori, passando da un massimo del 35,6% ad un minimo del 27,3%; i valori minimi del tasso di errore si osservano in corrispondenza dei valori delle probabilità a priori che assegnano maggior peso alla classe di minor interesse ma con un tasso di errata classificazione inferiore. Risulta pertanto essenziale osservare come varia il tasso di errata classificazione disaggregato per classe di appartenenza, per verificare se e dove esiste un miglioramento nella classificazione dell'altra classe ovvero quella dei diabetici.

Le procedure di aggregazione inoltre, sembrano essere meno sensibili alle variazioni sulle probabilità a priori; in particolare tale ridotta variabilità è da ascrivere in particolare al metodo *arcing* che fa registrare valori del tasso di errata classificazione compresi nell'intervallo 24,2-26,7; il metodo *bagging* presenta una accuratezza maggiore in corrispondenza di valori delle probabilità a priori pari ai valori stimabili dai dati campionari.

Le tabelle 4 e 5 riportano i risultati disaggregati per classe. Relativamente alla classe dei sani risulta evidente come tutti e tre i metodi

analizzati consentano di ottenere, all'aumentare della corrispondente probabilità a priori, un rilevante guadagno in termini di accuratezza della regola di classificazione ma è sicuramente dall'esame della seconda tabella che emergono i risultati di maggior interesse.

Tabella 4. *Tassi di errata classificazione nella classe dei sani*

π(1)	π(2)	CART standard	BAGGING	ARCING h=4
0,1	0,9	45,8	34,7	22,3
0,2	0,8	42,9	32,0	21,8
0,3	0,7	39,8	30,2	22,6
0,4	0,6	32,9	24,8	21,7
0,5	0,5	27,8	22,1	17,1
0,6	0,4	21,8	17,0	14,9
0,651	0,349	21,5	14,8	15,1
0,7	0,3	20,4	13,0	12,9
0,8	0,2	13,2	8,8	11,4
0,9	0,1	9,8	5,1	11,4

Tabella 5 *Tassi di errata classificazione nella classe dei malati*

π(1)	π(2)	CART standard	BAGGING	ARCING h=4
0,1	0,9	15,5	16,3	34,2
0,2	0,8	18,7	18,4	32,3
0,3	0,7	23,0	21,4	35,0
0,4	0,6	26,4	26,8	33,2
0,5	0,5	32,3	30,7	40,2
0,6	0,4	39,3	38,3	42,0
0,651	0,349	41,5	40,6	44,8
0,7	0,3	43,7	46,7	47,9
0,8	0,2	54,2	55,6	51,4
0,9	0,1	63,3	60,2	50,7

Come era lecito attendersi i risultati riguardanti la classe dei diabetici si contrappongono a quelli dell'altra classe: più le probabilità a priori risultano sbilanciate a favore dei malati minore è il tasso di errata classificazione ad essa associato. Rispetto al tasso del 40,6% inizialmente

ottenuto stimando le probabilità a priori sulla base dei dati campionari, tassi di errata classificazione dell'ordine del 15,5% per la procedura standard o del 16,3% per il *bagging* sono del tutto accettabili. La procedura *arcing*, al contrario, presenta un tasso di errore ancora elevato (32,3%); occorre sottolineare che tuttavia il metodo *arcing* presenta un vantaggio rispetto agli altri due metodi in quanto mostra un minor sbilanciamento tra i tassi di errore delle due classi.

Gli studi di simulazione svolti evidenziano come sia importante valutare i risultati ottenuti alla luce degli obiettivi specifici dell'analisi e delle priorità suggerite da coloro che dovranno servirsi della regola di classificazione, ma anche in ragione delle caratteristiche dei dati. Ancora una volta risulta importante non limitarsi a considerare il tasso di errata classificazione complessivo (Hand, 1996). Secondo le aspettative il metodo *arcing*, effettuando un campionamento di tipo adattivo, avrebbe dovuto garantire la massima diminuzione del tasso di errore rispetto alla procedura CART standard; dagli studi effettuati è invece emerso che tale metodo, rischiando di concentrarsi proprio sulle unità problematiche del collettivo è meno robusto del *bagging*; una eventuale riformulazione dello schema di aggiornamento dei pesi attribuiti alle osservazioni potrebbe, forse, essere di ausilio per risolvere situazioni analoghe a tale problema, in cui esiste una elevata sproporzione nei tassi di errata classificazione associati alle singole classi.

4. Stabilità di una regola di classificazione ad albero e ricerca di una struttura di sintesi

L'instabilità intrinseca alla procedura di costruzione dell'albero determina il successo delle procedure di aggregazione. In tal modo viene però meno una delle caratteristiche più interessanti degli alberi: la semplicità di rappresentazione della regola di classificazione.

Gli alberi di classificazione, come analizzato in precedenza possono dar luogo a classificatori instabili: campioni estratti dalla medesima popolazione, inducono strutture differenti, anche se si differenziano per poche unità.

La costruzione di un albero di classificazione procede tramite suddivisioni ricorsive dello spazio dei predittori, di conseguenza la dimensione campionaria disponibile per un determinato split decresce rapidamente all'aumentare della profondità dell'albero: gli split determinati nei rami più bassi sono più sensibili alla variabilità campionaria ne consegue quindi l'instabilità dell'albero. Questo comportamento è ancor più accentuato dalla natura gerarchica del processo di costruzione di un albero; ad un determinato livello, la suddivisione scelta garantisce una soluzione ottima per le unità statistiche osservate, ma condiziona il processo di crescita dell'albero da quel punto in poi. A tale proposito, quando si utilizza la *cross-validation*, mediante la suddivisione del campione in V sottocampioni, per stimare il tasso di errata classificazione, può verificarsi che all'allontanarsi dal nodo radice, aumenti la variabilità campionaria e i V alberi tendano a differenziarsi nella scelta delle variabili di partizione e nella suddivisione delle variabili stesse. Può accadere che le prime suddivisioni coinvolgano una sola variabile, anche se su livelli leggermente differenti; ma nei livelli più alti le strutture finiscono per differenziarsi e per non avere più caratteristiche comuni.

In generale non ci si attende che uno dei V alberi sia nettamente migliore rispetto agli altri, anche se è ragionevole pensare che si possa individuare una struttura, con caratteristiche di maggiore robustezza, definendo un albero "comune" sotteso a quelli trovati. Tale problema presenta analogogie con il confronto tra diverse partizioni individuate sulle medesime osservazioni, ma generate da algoritmi diversi o dall'impiego di diverse misure di dissimilarità tra le unità osservate, nell'ambito dell'analisi dei gruppi. Una interessante rassegna delle diverse proposte avanzate per determinare una struttura di consenso tra partizioni è presente in Gordon (1987).

Il confronto tra partizioni al fine di individuare misure di similarità tra di esse e la ricerca di una partizione di consenso presentano nell'ambito degli alberi caratteristiche peculiari che meritano una diversa trattazione. La partizione delle unità statistiche osservate, operata dai metodi di partizione ricorsiva, è nota nella struttura, nel senso che è nota la variabile che di volta in volta è coinvolta nel processo iterativo di segmentazione. E' opportuno quindi, nella scelta del sottoalbero comune,

tenere conto di tali informazioni e considerare tutte le possibili suddivisioni intervenute nel processo di costruzione dei singoli alberi individuati dai campioni ottenuti dalla procedura di *cross-validation*. L'obiettivo, non è quello di determinare una partizione di consenso, quanto un "albero di consenso". L'interesse è rivolto all'albero inteso come rappresentazione della regola di classificazione.

Una possibile soluzione, (si veda anche Miglio, 1996; Miglio e Pillati, 1997)), consiste nel definire un algoritmo che consenta di determinare tale albero con un procedimento iterativo basato su una misura di similarità tra partizioni. L'impiego della cross-validation è generalmente collegato alla determinazione di una stima più efficiente del tasso di errata classificazione per determinare un sottoalbero ideale, a partire da quello costruito sull'intero campione. Nella soluzione proposta si adotta un uso alternativo delle informazioni presenti nei campioni ottenuti dalla *cross-validation*, volto a determinare una struttura più semplice.

Come analizzato nel paragrafo 3 vari sono i metodi proposti in letteratura per combinare classificatori diversi o versioni diverse degli stessi, tali soluzioni consentono di ottenere un classificatore con un minor tasso di errore, anche se purtroppo si perde la possibilità di interpretare la nuova regola di classificazione. La procedura proposta consente di ottenere strutture più semplici, ancora interpretabili come una regola di classificazione ad albero.

4.1 Valutazione del consenso tra partizioni

Nel confronto tra diverse partizioni si può fare riferimento a misure di similarità o a misure di dissimilarità. Consideriamo inizialmente misure di similarità tra due partizioni relative alle medesime n unità statistiche; a tale proposito si introducono le misure proposte da Faith e Balbin (1986) basate sulla teoria di Day (1983) relativa al confronto di partizioni ordinate.

In generale il numero di elementi componenti le due partizioni sarà differente; indichiamo con k il numero di elementi della prima partizione Π_1 e con h il numero di elementi della seconda partizione Π_2.

Due partizioni possono essere poste a confronto misurando "l'ammontare di struttura" comune ad entrambe. Questo confronto può essere effettuato sulla base della intersezione tra le due partizioni contando le coppie di elementi che sono classificati insieme, cioè che appartengono allo stesso elemento, in entrambe le partizioni. Indichiamo con m_{ij}, $i=1,...,k$ e $j=1,...,h$, il numero di elementi in comune tra l'elemento i-esimo della partizione Π_1 e l'elemento j-esimo della partizione Π_2 e con M la matrice k x h degli elementi m_{ij}.

La misura che sarà utilizzata nel seguito è analoga a quella proposta da Fowlkes e Mallow (1983). Tali autori suggeriscono di utilizzare per il confronto tra due partizioni generate da algoritmi di *cluster analysis* il seguente indicatore[5]:

$$B_{kh} = \frac{T_{kh}}{(P_k Q_k)^{\frac{1}{2}}}$$

dove

$$T_{kh} = \sum_{i=1}^{k}\sum_{j}^{h} m_{ij}^2 - n$$

$$P_k = \sum_{i=1}^{k} m_{i.}^2 - n \qquad Q_h = \sum_{j=1}^{h} m_{.j}^2 - n$$

B_{kh} è, per costruzione, un indice che varia tra 0 e 1: il valore massimo sarà raggiunto quando $k=h$ e le due partizioni sono equivalenti.

Se $k=h=n$, M è una matrice di permutazione e B_{kh} è indeterminato. Tale indicatore ha una interessante interpretazione. Il numeratore di B_{kh} corrisponde al numero di tutte le possibili coppie di unità che appartengono al medesimo gruppo nelle due diverse partizioni. E' possibile inoltre determinare la media e la varianza di B_{kh} a patto di tenere fisse le marginali della matrice M.

[5] In realtà l'indicatore proposto è relativo al confronto tra partizioni aventi lo stesso numero di elementi, ma facilmente generalizzabile se si considerano partizioni aventi un numero differente di elementi.

Se una tale misura è adottata per il confronto tra due partizioni generate dagli alberi T_1 e T_2, occorre introdurre una leggera modifica. Poiché l'albero rappresenta una regola di classificazione, il confronto tra partizioni deve tener conto della classe assegnata a ciascun elemento della partizione. In tal modo si introduce una ulteriore variabile c_{ij}, che assume valore 1 solo se all'elemento i-esimo della partizione generata dall'albero T_1 corrisponde la medesima classe dell'elemento j-esimo della partizione generata dall'albero T_2. La misura di similarità è così modificata:

$$B(T_1,T_2) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{h} m_{ij}^2 c_{ij} - \sum_{i=1}^{k}\sum_{j=1}^{h} m_{ij} c_{ij}}{\sqrt{P_k \cdot Q_h}}$$

A numeratore compaiono solo le coppie di elementi che sono classificati insieme in entrambe le partizioni, a condizione che a tali elementi sia assegnata la medesima classe.

Esempio:

Consideriamo la matrice M generata dal confronto tra la partizione associata all'albero T_1 e la partizione associata all'albero T_2 aventi, rispettivamente, 5 e 3 foglie.

Si ipotizzi inoltre che il problema riguardi unità statistiche appartenenti a 3 classi distinte e che le classi associate alle foglie siano per il primo albero (2,3,1,2,1) e (2,1,3) nel secondo.

Tabella 6 *Confronti tra le due partizioni associate agli alberi* T_1 *e* T_2

		T_1					
T_2		f_{11} (2)	f_{12} (3)	f_{13} (1)	f_{14} (2)	f_{15} (1)	
f_{21}	(2)	**100**	120	0	**132**	0	352
f_{22}	(1)	0	0	**150**	0	**147**	297
f_{23}	(3)	0	**121**	0	0	0	121
		100	241	150	132	147	770

In tabella 6 sono riportati in grassetto i valori di m_{ij}, cui corrisponde un valore di c_{ij} pari ad uno, e che rientrano quindi nel calcolo di $B(T_1,T_2)$. Il valore di $B(T_1,T_2)$ è pari a 0,5067, contro un valore di 0,5913 corrispondente alla misura proposta da Fowlkes e Mallow.

Poiché il nostro obiettivo è quello di individuare una misura di similarità tra un albero di consenso T_c e i V alberi T_k costruiti sui campioni ottenuti dal processo di *cross-validation*, appare naturale assumere come misura di consenso la media delle misure di similarità tra coppie di partizioni in cui un elemento della coppia è rappresentato sempre dall'albero T_c. Definiamo quindi questa misura di consenso nel modo seguente:

$$CVCON = \frac{1}{V}\sum_{k=1}^{V} B(T_c,T_k)$$

4.2. Descrizione dell'algoritmo proposto.

L'algoritmo di seguito descritto si propone di individuare un nuovo albero da quelli costruiti sui campioni individuati dal processo di *cross-validation*, avente caratteristiche di maggiore robustezza poiché ottenuto come sintesi di questi[6].

Passo 1

Si costruiscono V alberi dai campioni di *cross-validation*. Ciascun albero è potato sulla base dell'algoritmo di L. Breiman et al. (1984) basato su una funzione di costo-complessità.

Passo 2

Si identificano come possibili suddivisioni tutte quelle intervenute nella costruzione dei V alberi precedenti.

Passo 3

[6] Nel seguito si fà sempre riferimento ad alberi determinati su campioni ottenuti da una procedura di cross-validation, ma il discorso si può estendersi a strutture costruite su campioni ottenuti con diverse tecniche di ricampionamento.

Si procede alla costruzione dell'albero con un procedimento di tipo iterativo. La prima suddivisione è scelta tra le suddivisioni ottenute nel passo precedente, ed è quella che determina il maggior valore nella misura media di consenso. Se esistono più partizioni che danno luogo alla medesima misura media di consenso, la prescelta risulta essere quella con la più alta percentuale di presenza nella costruzione dei diversi sottoalberi.

Nei passi successivi si sceglierà la suddivisione che consente di ottenere il maggior incremento nella misura media di consenso; tale scelta è iterata e il processo termina quando nessuna suddivisione comporta un aumento nella misura di consenso media rispetto al passo precedente. Inoltre non si considerano ulteriormente suddivisibili i nodi puri e i nodi che contengono un numero minimo prefissato di unità.

4.3. Un esempio su dati simulati

I risultati che seguono sono stati ottenuti applicando tale algoritmo ad un insieme di dati simulati tratto da Breiman *et al.* (1984). Si tratta di un problema a tre classi in cui il vettore dei predittori **x** è costituito da 21 elementi. Sono date tre onde $h_1(t)$, $h_2(t)$,$h_3(t)$ rappresentate in figura 2.

Ogni classe è ottenuta da una combinazione lineare convessa di due delle tre onde, cui si somma un termine di errore distribuito secondo una normale standardizzata.

Per generare un vettore di classe 1, si generano indipendentemente un numero casuale u estratto da una distribuzione uniforme e 21 numeri casuali normalmente distribuiti con media 0 e varianza 1. E quindi si pone :

$$x_m = uh_1(m) + (1-u)h_2(m) + \varepsilon_m \qquad m = 1,...21$$

Per generare un vettore di classe 2, si ripete il procedimento e si pone:

$$x_m = uh_1(m) + (1-u)h_3(m) + \varepsilon_m \qquad m=1,...21$$

I vettori di classe 3 sono generati dalla :

$$x_m = uh_2(m) + (1-u)h_3(m) + \varepsilon_m \qquad m=1,...21$$

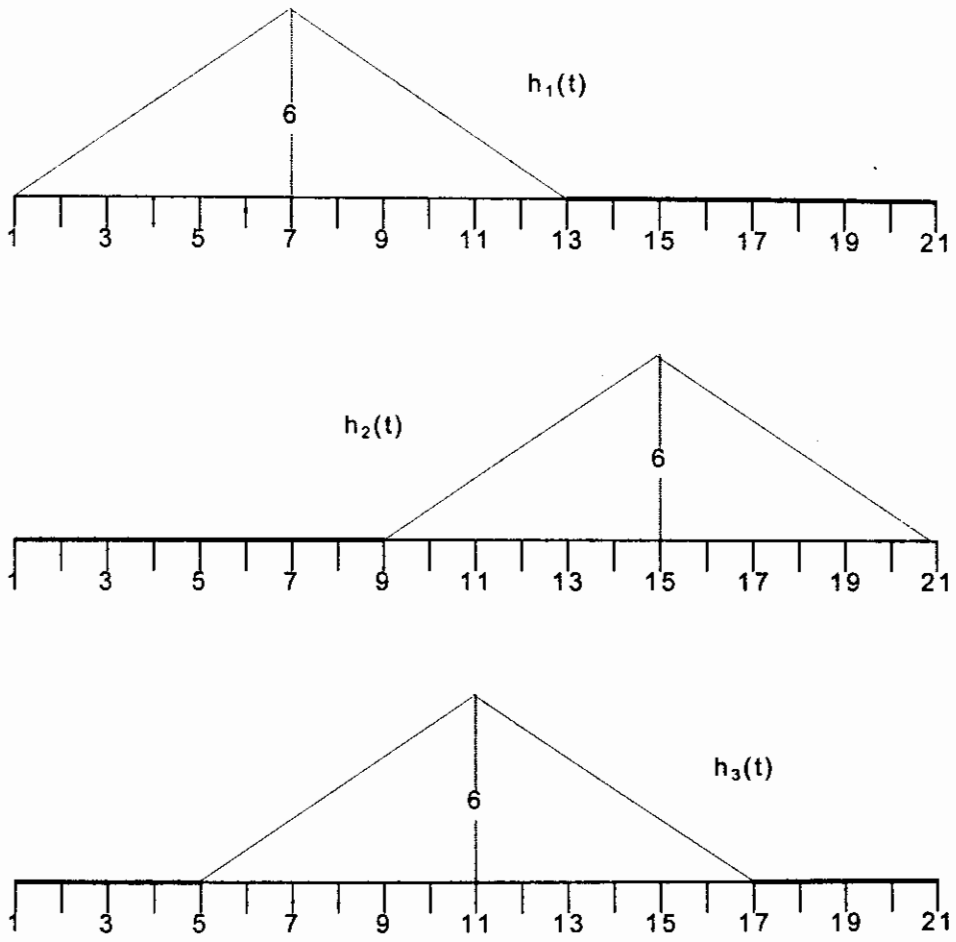

Figura 2

Il campione è costituito da 300 osservazioni, ottenute assegnando pari probabilità a priori a ciascuna classe. E' stato costruito un albero

dall'intero campione e dai 10 campioni ottenuti dalla *cross-validation*. Analogamente sono state generate 900 ulteriori osservazioni da usare come insieme di *test* per una stima corretta del tasso di errata classificazione.

La simulazione precedente è stata ripetuta 30 volte seguendo due diversi procedimenti: il primo albero T_p è costruito sull'intero campione e potato secondo la procedura di Breiman; il secondo, T_c è costruito sulla base della procedura proposta.

In tabella 7 sono riportati i tassi di errata classificazione valutati sul campione di test per entrambi gli alberi nell'insieme delle diverse simulazioni:

Tabella 7. *Risultati medi (dev. st.) nelle 30 simulazioni.*

	Campione di test	Numero medio foglie
T_p	30.9% (± 2.2)	10.6(±2.2)
T_c	31.1% (± 2.4)	7.5(±1.9)

Chiaramente l'albero costruito con la procedura proposta non consente di ottenere prestazioni differenti da quelle ottenute con la procedura di potatura tradizionale, ma la struttura è sicuramente più semplice. Analizzando i singoli sottoalberi relativi ai campioni di cross-validation, si può notare come questo in un certo senso sintetizzi l'informazione presente in ciascuno di tali alberi.

Tabella 8. *Valori medi delle misure di consenso (dev. st.) nelle 30 simulazioni.*

$B(T_p, T_c)$	0.754 (±0.126)
$CVCON(T_p)$	0.686 (± 0.070)
$CVCON(T_c)$	0.739 (± 0.091)

Tale considerazione appare maggiormente evidente se si confrontano le misure di consenso proposte, valutate per i due distinti alberi e relativamente ai campioni di *cross-validation*. I valori medi osservati

confermano le precedenti considerazioni: l'albero di consenso è maggiormente rappresentativo dei diversi alberi di cross-validation e per la struttura più semplice può sostituire l'albero ottenuto da una procedura di pruning.

5. Regole di classificazione ad albero e reti neurali.

Con riferimento a problemi di classificazione, studi recenti hanno mostrato i vantaggi derivanti da un uso congiunto di alberi di classificazione e reti neurali. In diversi lavori, si veda a tale proposito Chabanon *et al.* (1992), Ciampi e Lechevalier (1995), Miglio e Pillati (1996), si è mostrato come un albero possa essere rappresentato mediante una rete neurale: in particolare un percettrone multistrato con due strati intermedi. Le diverse impostazioni presentano il comune obiettivo di considerare tale struttura come architettura di riferimento da cui ottenere, mediante una successiva stima dei parametri della rete, una regola di classificazione dotata di maggior potere discriminante rispetto agli alberi da cui è stata dedotta.

Una rete neurale di tipo *feedforward* con due livelli nascosto, h_1 e h_2 unità nascoste, s unità di input e k unità di output corrisponde ad una funzione non lineare, relativamente alla j-esima unità di output, del tipo:

$$g(\boldsymbol{x},\theta) = F\left(\alpha_0 + \sum_{i=1}^{h_2} \alpha_i \Psi_{(2)}\left(\delta_0 + \sum_{j=1}^{h_1} \delta_j \Psi_{(1)}\left(\gamma_j' \boldsymbol{x}\right)\right)\right)$$
$$= F\left(\alpha' \cdot \Psi_{(2)}\left(\delta' \Psi_{(1)}\left(\gamma' \boldsymbol{x}\right)\right)\right)$$

dove $\theta = \left(\alpha_1,\ldots,\alpha_k,\delta_0,\ldots,\delta_{h_2},\gamma_1,\ldots,\gamma_{h_1}\right)$ rappresenta il vettore dei parametri del modello, $\gamma = \left(\gamma_1,\ldots,\gamma_{h_1}\right)$ è una matrice di dimensioni $(p+1)*h_1$ e rappresenta i pesi delle connessioni in ingresso alle unità del primo stadio intermedio; $\delta = \left(\delta_0,\ldots,\delta_{h_2}\right)$ è una matrice di dimensioni $(h_1+1)*h_2$ e rappresenta i pesi in ingresso alle unità del secondo strato intermedio; $\alpha = \left(\alpha_1,\ldots,\alpha_k\right)$ è una matrice di dimensioni $(h_2+1)*k$ e

W. Buntine, T. Niblett (1992) *A further comparison of splitting rule for decision-tree induction* Machine Learning, 8, pp 75-86.

C. Chabanon, Y. Lechevallier, S. Milleman (1992) *An efficient neural network by a classification tree*, in Y. Dodge, J. Whittaker, *Computational Statistics*, Physica-Verlag, Heidelberg.

A. Ciampi, J. Thiffault (1988) *Recursive partition in biostatistics: stability of trees and choice of the most stable classification,* Compstat 1988, Physica Verlag.

A. Ciampi, J. Thiffault, U. Sagman (1989) *RECPAM: a computer program for recursive partitioning and amalgamation for censored survival data and other situations frequently occurring in biostatistic. II. Application to data on small cell carcinoma of the lung (SCLL),* Computer Methods and Programs in Biomedicine, 30, pp 283-296.

A. Ciampi (1991) *Generalized regression trees,* Computational Statistics & Data Analysis, 12, 57-78.

L. A. Clark, D. Pregibon (1992) *Tree-based models,* Capitolo 9 di J. M. Chambers and T.J. Hastie *Statistical Models in S*, Chapman and Hall, New York.

E.B. Fowkles, C.L. Mallow (1983) *A method for comparing two hierarchical clustering,* Journal of the American Statistical Association, 78, pp 553-584.

Y. Freund, R.E. Shapire (1999) *A short introduction to boosting,* Journal of Japanese Society for Artificial Intelligence, 14 (5), pp. 771-780.

Y. Freund, R.E. Shapire (1999) *Discussion of the paper "Arcing classifiers" by Leo Breiman* The annals of statistics, 26 (3), pp. 824-832.

J.H. Friedman (1999a), *Greedy Function Approximation: A Gradient Boosting Machine*, Technieal report, Department of Statistics, Stanford University.

J.H. Friedman (1999b), *Stochastic Gradient Boosting*, Technical report, Department of Statistics, Stanford University.

M.A. Gillo, M.W. Shelly (1974) *Predictive modeling of multivariable and multivariate data,* Journal of the American Statistical Association, 69, pp 646-653.

C. Gini (1951) *Lezioni di statistica,* Eredi Virgilio Veschi, Roma.

A.D. Gordon (1987) *A review of Hierarchical Classification,* Journal of Royal Statistical Society, A, 150, 2, pp 119-137.

D.J Hand., *Construction and Assessment of Classification Rules*, Wiley, Chichester.

D.J. Hand, J.J. Oliver (1996) *Averaging over Decision Trees,* Journal of Classification, 13, pp 281-297.

J. Hertz, A. Krogh, R. Palmer (1991), *Introduction to the theory of neural computation*, Addison Wesley, Redwood City.

K. Hornik, M. Stinchcombe, H. White (1990), *Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Networks*, 3, pp.551-560.

G.V. Kass (1980) *An exploratory technique for investigating large quantities of categorical data,* Applied Statistics, 29, pp 119-127.

M. Le Blank, J. Crowley (1992) *Relative risk trees for censored survival data,* Biometrics, 48, 411-425.

Journal of the American Statistical Association, 87, pp 407-418.

C.E. Shannon, W. Weaver (1949) *The Mathematical Theory of Communications*, University of Illinois Press, Urbana, Illinois.

Y. Yuan, M.J. Shaw (1995) *Induction of fuzzy decision trees*, Fuzzy Sets and Systems, 69, pp 125-139.

R. Tibshirani (1996) *Bias, variance and prediction error for classification rules* Thecnical report,Department of Statistics, University of Toronto.

H. White (1989), *Learning in artificial neural networks: a statistical perspective*, Neural Computation, 1, pp. 425-464.