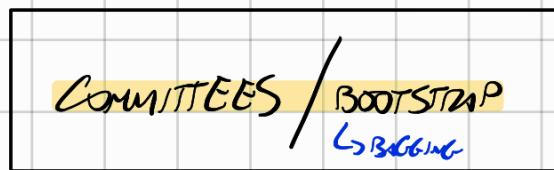


FINO AD ORA I MODELLI VISTI PER LA REGRESSIONE E LA CLASSIFICAZIONE ERANO SINGOLARI, ONGRA SI UTILIZZAVA SOLO UN UNICO MODELLO
È POSSIBILE MIGLIORARE A MIGLIORARE LE PREVISIONI COMBINANDO PIÙ MODELLI, AD ESEMPIO

- SI POTREBBERO ALLENARE M MODELLI E CONSIDERARE LA MEDIA DELLE LORO PREVISIONI
- SI POTREBBERO ALLENARE IN MODO SEQUENZIALE PIÙ MODELLI IN MODO DA CONSIDERARE IL GIORNO DEL MODELLO PRECEDENTE NELLA COTENZA
- SI POTREBBERO ALLENARE PIÙ MODELLI DIVERSI E Poi CONSIDERARE IL NUOVO SUSSO BASE DELLE PREVISIONI)

QUESTI MODELLI SONO DEFINITI **ENSEMBLE** POICHÉ RENDONO DELLE PREVISIONI SULLA BASE DI UN INSIGNE DI MODELLI ALLENATI, INVOCI CUI DI UN SINGOLO MODELLO.



ABBIAMO GIÀ VISTO CHE COMBINARE MODELLI MOLTIPI E Poi CONSIDERARE LA MEDIA DI PERMETTE DI RIDURRE LA VARIANZA

SE CONSIDERAMO UN PROBLEMA DI REGRESSIONE (Dove si vuole PRENDERE UN SINGOLO VALORE TARGET CONTINUA) SI HA CHE IL METODO DI RIDUZIONE DELLA VARIANZA SI OTTENGUE FITTANDO UN MODELLO CON LOW BIAS (\Rightarrow ALTA VARIANZA) A DATASETS MOLTIPI ANDANDO Poi AD ESSEGUIRE LA MEDIA DEI RISULTATI OTTENUTI.

TUTTAVIA, NELLA PRATICA, SI DISPONE SOLO DI UN UNICO DATASET. IL METODO **Bootstrap** CI PERMETTE QUINDI DI CAMPIONARE M DATASETS DI DIMENSIONE $N \times N$ DAL DATASET ORIGINALE D CON RIPETIZIONE. QUINDI, OBTENUTO DI QUESTI DATASETS, OTTENERE UNA DISTRIBUZIONE INTRASSETA DI D MA IN MANIERA INCOMPLETA.

IL POSSO SUCCESSIVO È QUINDI QUALE SI FITTARE UN MODELLO $y_m(\bar{x})$ A CIASCU

BOOTSTRAP DATASET E POI ANDARE A FORMARE IL COMMITTEE MODEL MEDIANO DI M MODELLI BASE:

$$y_{\text{com}}(\bar{x}) = \frac{1}{M} \sum_{m=1}^M y_m(\bar{x})$$

(A MEDIA DI TUTTI GLI M MODELLI)

E, SE CONSIDERANO $h(\bar{x})$ COME LA VERA FUNZIONE CHE HA GENERATO D, POSSUMO ALLORA ASSUNGERE L'OUTPUT DI CIASCU MODELLO COME LA VERA FUNZIONE + GRANDE:

$$y_m(\bar{x}) = h(\bar{x}) + \epsilon_m(\bar{x})$$

↓
DATI SIMULATI
per modello

↑
VERA FUNZIONE
generatrice

errore

POSSIAMO ALLORA CALCOLARE L'EXPECTED SQUARED ERROR DI CIASCU MODELLO:

$$\begin{aligned} E_{\bar{x}} \left[\{h(\bar{x}) - y_m(\bar{x})\}^2 \right] &= E_{\bar{x}} \left[\{h(\bar{x}) - h(\bar{x}) + \epsilon_m(\bar{x})\}^2 \right] \\ &= E_{\bar{x}} [\epsilon_m(\bar{x})^2] \end{aligned}$$

ALLO STESSO MODO SI PUÒ CALCOLARE L'EXPECTED ERROR DEL COMMITTEE:

$$E_{\text{com}} = E_{\bar{x}} \left[\left\{ \frac{1}{M} \sum_{m=1}^M \epsilon_m(\bar{x}) \right\}^2 \right]$$

errore atteso del
committee

TUTTAVIA FARE IL QUADRATO DI QUESTA SOMMA È UN BEL CASINO, E PER QUESTO SI ASSUME

CHE C'È DI NUOVO:

$$\cdot \underline{\mathbb{E}_x [\mathcal{E}_m(\bar{x})]} = 0 \quad \text{MEDIA nulla}$$

$$\cdot \underline{\mathbb{E}_x [\mathcal{E}_m(\bar{x}) \mathcal{E}_l(\bar{x})]} = 0 \quad m \neq l \quad \text{SCORRELATI}$$

QUESTO CI PERMETTE DI SEMPLIFICARE IL TUTTO E DI OTTENERE

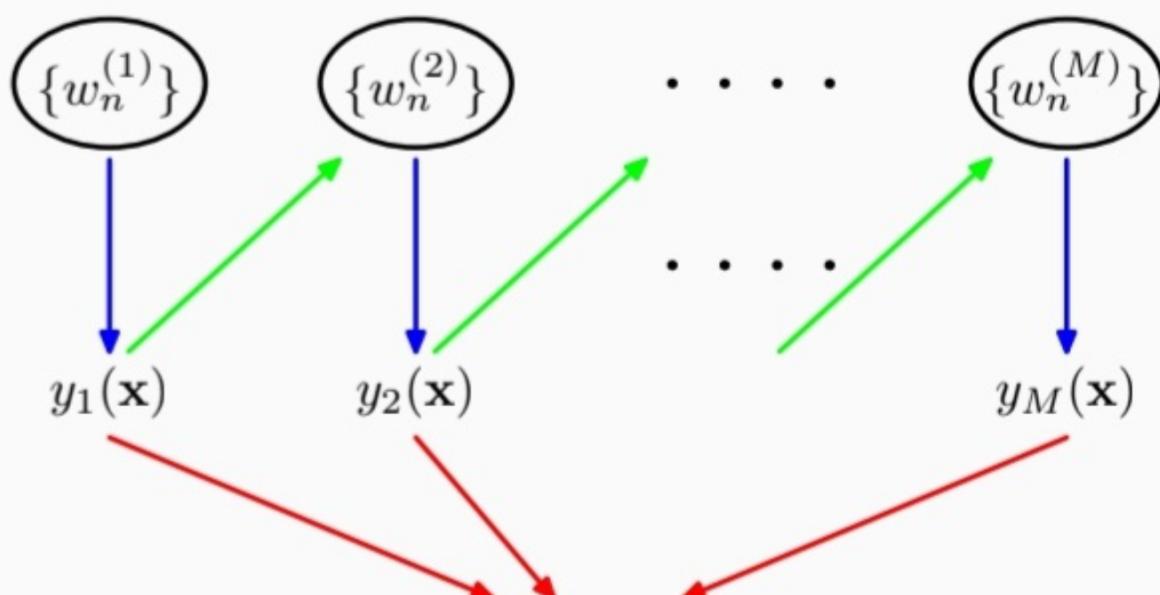
$$\mathcal{E}_{\text{com}} = \frac{1}{M} \mathcal{E}_{\text{AV}}$$

quando si si cicla la media delle M modelli si va a ridurre l'expected error del singolo modello di un fattore $\frac{1}{M}$. Proprio perché abbiamo assunto che gli errori sono tra di loro scorrelati, ovviamente negli casi dove questo veramente accade, ma le risultanti ci garantisce almeno che non si possono ottenere risultati peggiori, anziché $\mathcal{E}_{\text{com}} \leq \mathcal{E}_{\text{AV}}$.

BOOSTING

→ CIASCUndo modello si concentra sui errori complessi del precedente

L'idea più o' quella di non considerare più tutti i modelli cattivi, ma allenerli in sequenza in modo che "si aiutino a vicenda". Può essere l'obiettivo del Boosting:



$$Y_M(\mathbf{x}) = \text{sign} \left(\sum_{m=1}^M \alpha_m y_m(\mathbf{x}) \right)$$

PREDIRORE DEL
MODELLO

→ PESO DI "peso" DEL
MODELLO!

ADABoost

L'algoritmo **ADABoost** esce questa tecnica nell'ambito della classificazione binaria. quindi, partendo da un dataset iniziale $D = \{(x_1, t_1), \dots, (x_N, t_N)\}$ con $t_n \in \{-1, 1\}$. Si escono i seguenti passi:

- Si associa a ciascun campione un peso, inizialmente $\frac{1}{N}$
 → WEAK LEARNER
- Si ASSOCIA di aggiornare un classificatore con i campioni pesati
- Dopo aver allenato UN classificatore, si unisce a combinarlo in un committee dove si avranno degli specifici coefficienti che variano a seconda ciascun classificatore sulla BIG delle PESAZIONI

AdaBoost

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = \frac{1}{N}$ for $n = 1, \dots, N$.
2. For $m = 1, \dots, M$:
 - (a) Fit a classifier $y_m(x)$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^N w_n^{(m)} I(y_m(x_n) \neq t_n) \quad (14.15)$$

where $I(y_m(x_n) \neq t_n)$ is the indicator function and equals 1 when $y_m(x_n) \neq t_n$ and 0 otherwise.

- (b) Evaluate the quantities

GRAN PESO DI y_m

$$\epsilon_m = \frac{\sum_{n=1}^N w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^N w_n^{(m)}} \quad (14.16)$$

and then use these to evaluate

ASSOCIA UN VERTO A CLASSIFICATORE, $\alpha_m = \ln \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\}$. (14.17)

(c) Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp \{ \alpha_m I(y_m(\mathbf{x}_n) \neq t_n) \} \quad (14.18)$$

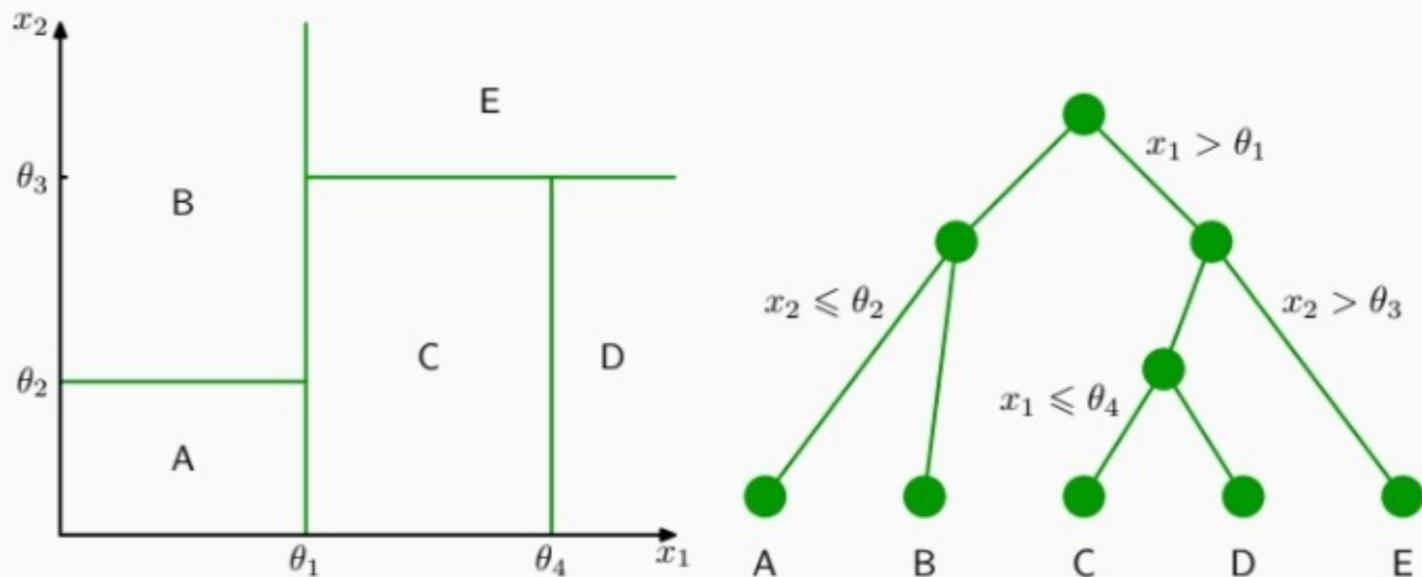
→ puoi servire per dare più importanza ai valori misclassificati!

3. Make predictions using the final model, which is given by

$$Y_M(\mathbf{x}) = \text{sign} \left(\sum_{m=1}^M \alpha_m y_m(\mathbf{x}) \right). \quad (14.19)$$

TREE MODELS

POSSIAMO PENSARE AI CONCETTI DI CLASSIFICAZIONE E REGRESSIONE CONG DELL'EQUAZIONE COSTANTE SU DELLE PARTIZIONI DELLO SPAZIO DI INPUT.



i **TREE MODELS** sono dei modelli che lavorano partizionando lo spazio. In particolare suddividono lo spazio in cubi allineati con gli assi. Fanno parte dei modelli ensemble, dove ciascun singolo modello è responsabile per ciascuna partizione. Al fine di decidere quale modello effettivamente utilizzare, si attraversa l'insieme iniziale con **processo di decisione sequenziale**.

Consideriamo un problema di regressione con dataset $D = \{(x_1, t_1), \dots, (x_n, t_n)\}$ dove t_m sono i targets. Vogliamo quindi partizionare lo spazio \mathbb{R}^d in modo che in ogni partizione la stima ricorsa dei targets minimizzi lo squared error. Per ottenere questo si media il numero di targets t_n di tutti i punti x_n che ricadono in quella partizione, ma come si può ottenere questa partizione?

Si utilizza un approccio greedy learning:

- Si parte da un qualsiasi modo radice corrispondente all'intero spazio \mathbb{R}^d
- Si spartono su tutte le possibili \rightarrow caratteristiche del dataset splitting variables
 - Si considera ciascuna possibile threshold per separare il dataset in due insiem
 - Si seleziona quella che minimizza la sum of squared errors nello split.
- Si applica ricorsivamente la procedura ai figli dei nodi iniziali

\rightarrow Profondità massima / Criterio minimo di informazione
 La condizione di arresto si ha quando si è raggiunto lo spazio di input in modo da aver creato un albero profondo e largo. Tuttavia purtroppo procedura può essere problematica e contiene guai overfitting. Per evitare questo si estende una regola dell'albero al fine di bilanciare ed eseguire quindi un trade-off tra complessità e la minimizzazione dell'errore sul training data.

Assumiamo di avere un partizionamento (guai in albero) con i nodi fissati individuati da $T=1, \dots, T$

Definiamo la partizione corrispondente a ciascun nodo R_T . Definiamo inoltre:

• Predizione ottimale:

$$Y_T = \frac{1}{N} \sum_{x_n \in R_T} t_n$$

• Squared Error:

$$Q_T(T) = \sum_{x_n \in R_T} (t_n - Y_T)^2$$

POSSIAMO QUINDI DEFINIRE UN CRITERIO DI POTUTO:

$$C(T) = \sum_{r=1}^{|T|} Q_r(T) - \lambda |T|$$

FOLLE

→ DEFINISCE quanto complesso
il modello

↳ MURO IL TRADE-OFF TRA GRANDEzza E COMPLESSITÀ.

SE INVECE ABBIAMO UN PROBLEMA DI CLASSIFICAZIONE SI COMBINA SOLUZIONE

COSÌ VIENE DEFINITO L'ERRORE. DEFINIAMO P_{TH} LA PROPORTIONE DI CAMPIONI DELLA CLASSE K AL NODO T:

SI POSSONO QUINDI DEFINIRE DUE TIPOLOGIE DI ERRORE DIVERSE:

• NEGATIVE cross entropy:

$$Q_r(T) = \sum_{h=1}^K P_{Th} \ln P_{Th}$$

• GINI INDEX:

$$Q_r(T) = \sum_{h=1}^K P_{Th} (1 - P_{Th})$$

CONDITIONAL MIXTURE MODELS

SE CONSIDERAMO IL SOTTO-INSERTE $D = \{(x_1, t_1), \dots, (x_n, t_n)\}$,
COSÌ SUCCIDE SE LE INFORMAZIONI SONO RISULTATO CORRETTAMENTE
SPECIFICATE DA UNA SIMILE FUNZIONE DI REGRESSIONE CHIAMATA $\tilde{w}^T x + b$?

SI USANO UTILIZZANDO UNA INTERPRETAZIONE PROBABILISTICA DELLA REGRESSIONE
LINEARE E, INOLTRÉ CHE TRAVERSO UNA SIMILE STIMA MAXIMUM LIKELIHOOD
PER UN SIMILE MODELLO, SI UTILIZZA UNA EXPECTATION MAXIMIZATION PER
TUTTI I PARAMETRI DELLA MIGRAZIONE DI REGRESSIONE PROBABILISTICO!

DEFINIMMO quindi una **CONDITIONAL MIXTURE DISTRIBUTION**:

$$p(t|\bar{\theta}) = \sum_{h=1}^K \pi_h N(t|\bar{w}_h^\top \phi, \beta^{-1})$$

E, POSSIAMO AL LAVORARE:

$$\ln p(t|\theta) = \sum_{m=1}^N \ln \left(\sum_{h=1}^K \pi_h N(t|\bar{w}_h^\top \phi_m, \beta^{-1}) \right)$$

INTRODUCENDO POI IL **VARIABILE LATENTE Z**:

$$\ln p(t, z|\theta) = \sum_{m=1}^N \sum_{h=1}^K \left\{ z_{mh} \ln N(t|\bar{w}_h^\top \phi_m, \beta^{-1}) \right\}$$

INFINE POSSIAMO UTILIZZARE I POSSI DGLS **EXPECTED MAXIMIZATION PER GRADIENTE**
IL SIME MLE DI TUTTI I PARAMETRI:

- SI CALCOLA IL **RESPONSE BIAS**:

$$Y_{mh} = E(z_{mh}) = P(H|t, \phi_m, \theta) = \frac{\pi_h N(t|\bar{w}_h^\top \phi_m, \beta^{-1})}{\sum_{j=1}^K \pi_j N(t|\bar{w}_j^\top \phi_m, \beta^{-1})}$$

M-STEP MIXING COEFFICIENTS:

$$\pi_k = \frac{1}{N} \sum_{m=1}^N \gamma_{mk}$$

M-STEP WEIGHTS:

$$\bar{w}_k = (\bar{\phi}^\top \bar{R}_n \bar{\phi})^{-1} \bar{\phi}^\top \bar{R}_n \bar{t}$$

M-STEP PRECISION

$$\frac{1}{\beta} = \frac{1}{N} \sum_{m=1}^N \sum_{h=1}^K \gamma_{mh} (t_m - \bar{w}_n^\top \phi_m)^2$$

