ADABOOST

Decision Stumps

Andrea Mugnai

1 Introduzione

L'elaborato in questione ha lo scopo di fornire un'implementazione dell'algoritmo di Boosting chiamato Adaboost, attraverso l'uso di "Decision Stumps" come base learner. L'obiettivo di questi algoritmi è quello di fare "Ensamble Learning" cioè combinare tanti singoli modelli per costruirne uno principale (Master model). L'algoritmo verrà applicato al dataset di Bank Marketing e i relativi risultati verranno poi confrontati con quelli riportati in "Moro et al. 2014".

2 Materials

Il Dataset considerato riguarda una campagna di marketing effettuata da una banca Portoghese, con l'obiettivo di convincere il cliente ad effettuare un deposito bancario nella loro banca. La campagna è basata su delle chiamate telefoniche. L'obiettivo è quello di predire attraverso diverse variabili (20) se il cliente deciderà di sottoscrivere un deposito con la banca (espresso dalla variabile y). Nel nostro caso abbiamo utilizzato un dataset composto da 41188 esempi e come già detto 20 input. Il Dataset è inoltre diviso in:

- Data: Rappresentato da "X". Valori che servono per effettuare la predizione. Sono tutte le colonne di input.
- Target: Chiamato "y". Questo rappresenta la variabile di output (binario: "yes,"no"). Cioè il risultato della classificazione (effettueranno o no un deposito?).

3 Implementazione dell'algoritmo

L'implementazione del "mio" Adaboost si basa sul definire due differenti metodi contenuti all'interno della classe "MyAdaboost": "fit" e "predict". Il primo inizialmente attribuisce a ciascuno dato un peso (per tutti lo stesso al passo 1) e successivamente utilizza la libreria "sk-learn" per costruire base learner (stumps). In particolare il metodo "DecisionTreeClassifier" crea l'albero con altezza 1 e max foglie 2, applicandoci i suoi rispettivi metodi di "fit" e "predict".

Quest'ultimo mi permette di trovare la predizione di y (y_pred). Utilizzando quindi lo pseudocodice di "Adaboost", visto a lezione, riesco a trovare tutti i parametri di cui ho bisogno per completare l'algoritmo:

- **error**. L'errore del mio weak learner.
- **alpha**. Valori positivi elevati indicano che l'ultimo "stump" ha effettuato un buon lavoro a classificare il campione.
- **weight**. Che sapevo già ma vado ad aggiornare a seconda se il mio "stump" si era sbagliato o no. Il nuovo peso verrà utilizzato nell'iterazione successiva così che il prossimo "stump" prenderà in considerazione gli errori di quello precedente.

L'algoritmo risulta completo solo dopo l'implementazione di "predict", che attraverso il calcolo di una funzione ci restituisce la classificazione, dopo aver effettuato l'operazione di Boosting. La funzione costruita fa uso del parametro "alpha" per effettuare la predizione.

4 Uso del Dataset e implementazione Main

Per una verifica più completa basata anche sul vecchio dataset, ho deciso di implementare la scelta tra vecchio e nuovo dataset per notarne anche le differenze. Attraverso un comando, l'utente potrà decidere se sottoporre ad AdaBoost il dataset più recente (premendo il tasto 1) oppure quello più datato (con il tasto 2). Una volta scelto, il programma inizierà a dividerlo in due parti come scritto sopra (data e target, alla sezione 2). La successiva divisione invece riguarda la creazione di "training set" e di "test set" sia per X (Data) che per y (Target). Infine il programma stampa a schermo "AUC" (Area Under the Curve) e "Accuracy" che ci serviranno per le considerazioni finali.
Lo stesso dataset è sottoposto alla funzione "AdaBoostClassifier" importata dalla libreria "sklearn", per un confronto con l'algoritmo creato da me.

5 Conclusioni

Quello che possiamo notare dalle considerazioni finali e in particolare dalle osservazioni del parametro "AUC" and "Accuracy" è che il nostro algoritmo sembra svolgere le operazioni in modo corretto. Infatti i due valori sono uguali per entrambe le implementazioni: "MyAdaBoost" e "AdaBoostClassifier" preso da sklearn. L'unica cosa che non torna perfettamente, forse dal fatto che non sono riuscito a trovare il "dataset" preciso, è l'uguaglianza con i dati forniti da "Moro et al. 2014". In questo articolo infatti è stato utilizzato un set ridotto rispetto a quello originale composto da 22 caratteristiche rilevanti. Nonostante l'utilizzo dei due "Dataset" presi dal sito: `https://archive.ics.uci.edu/ml/datasets/Bank+Marketing` non sono riuscito a trovare valori, per quanto riguarda la metrica AUC, simili a quelli descritti nell'articolo menzionato precedentemente.

References

Per la costruzione degli stumps ho utilizzato un articolo scritto da "Alvaro Corrales Cano". Towardsdatascience.com; Adaboost-from-scratch. https://towardsdatascience.com/adaboost.from-scratch-37a936da3d50