**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

****

**MÔN HỌC: Học Máy**

**Đề Tài: Tìm Hiểu về Boosting và thuật toán BrownBoost**

**Giảng viên: Trần Nhật Quang**

**Nhóm sinh viên thực hiện:**

**○ Lê Đức Anh 16110005**

* **Đào Lê Văn Vinh 16110267**

**Mục Lục**

[**I.** **Tìm hiểu về Boosting** 1](#_Toc9542200)

[**1.** **Giới thiệu** 1](#_Toc9542201)

[**2.** **Ý tưởng của Boosting** 1](#_Toc9542202)

[**2.1.** **Ý Tưởng :** 1](#_Toc9542203)

[**2.2.** **Cách hoạt động** 1](#_Toc9542204)

[**3.** **Một số loại thuật toán phổ biến của Boosting** 1](#_Toc9542205)

[**4.** **Ví dụ về AdaBoost** 2](#_Toc9542206)

[**II.** **Thuật toán BrownBoost** 2](#_Toc9542207)

[**1.** **Giới thiệu** 2](#_Toc9542208)

[**2.** **Ý tưởng của BrownBoost** 3](#_Toc9542209)

[**2.1.** **Ý tưởng :** 3](#_Toc9542210)

[**2.2.** **Thuật toán BrownBoost** 3](#_Toc9542211)

[III. Code Demo 5](#_Toc9542212)

[**IV.** **Tài liệu tham khảo** 9](#_Toc9542213)

# **Tìm hiểu về Boosting**

## **Giới thiệu**

* Boosting là một dạng của phương pháp **Ensemble**, phương pháp Ensemble này sử dụng nhiều thuật toán khác nhau cùng train những weak learner thành các strong learner nhằm có được predictive performance (hiệu năng dự đoán) cao.
* Boosting tổng hợp kết quả đầu ra của các thuật toán bằng cách dựa trên số lượng Vote (đối với Classification) hoặc tính giá trị trung bình (đối với Numeric prediction , VD : Regression,…) [1]
* Ví dụ : Với việc dự đoán email có phải là spam hoặc not spam, ta có tổng cộng 5 thuật toán dự đoán, trong đó 3 thuật toán dự đoán là spam và 2 thuật toán dự đoán là không spam, Boosting sẽ cho ra output email đó là spam vì số lượng Vote của spam là cao hơn không spam.

## **Ý tưởng của Boosting**

### **Ý Tưởng :**

Ý tưởng của Boosting là tạo ra các weak learner (dự đoán kém),mỗi weak learner sẽ cố gắng sửa lỗi cho weak learner trước, cuối cùng là tổng hợp lại kết quả của các weak leaner để tạo ra một strong learner có performance cao.

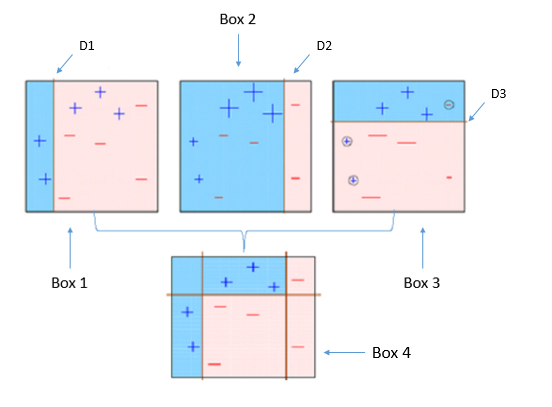
### **Cách hoạt động**

* + **Bước 1** : Sử dụng một thuật toán để dự doán DataSet
  + **Bước 2** : Kiểm tra model vừa rồi có gì sai sót, nếu có thì đánh trọng số Weight ưu tiên cho các sai số đó và sau đó áp dụng thuật toán tiếp theo
  + **Bước 3** : Lặp lại bước 2 cho đến khi hết số lượng thuật toán hoặc hoặc đạt được độ chính xác thích hợp.

## **Một số loại thuật toán phổ biến của Boosting**

* AdaBoost (Adative Boosting)
* Gradient Boosting
* XGBoost

### **Ví dụ về AdaBoost**



+ Tại Box1 ta thấy dự đoán 3 dấu **+** phía trên bị sai, ngay sau đó Box2 đã tăng Weight ưu tiên cho 3 dấu **+** ấy nhưng lại phát hiện ra 3 dấu **–** phía dưới dự đoán sai, vì vậy đến Box3 đã tăng Weight ưu tiên cho 3 dấu **-**

+ Tại Box4 ta có tổng hợp các trước hợp của 3 Box trước đó và có được model có Performance cao nhất có thế

# **Thuật toán BrownBoost**

## **Giới thiệu**

* Với AdaBoost, ta có thể tổng hợp lại với việc tổng hợp các Weak Learner để cho ra kết quả tốt, nhưng đó là khi bộ Dataset của ta tốt, có ít hoặc không có dữ liệu nào bị Noisy Dataset (dữ liệu bị nhiễu), khi đó kết quả sẽ chắc chắn bị vì mặc dù train đi train lại rất nhiều lần nhưng cho ra kết quả bị sai. Từ đó, thuật toán BrownBoost được sinh ra để giải quyết vấn đề Noisy Dataset này.
* Trái với AdaBoost, tập trung tránh đi việc bỏ qua dữ liệu thì BrownBoost lại cố gắng bỏ qua các dữ liệu bị Noisy.

## **Ý tưởng của BrownBoost**

### **Ý tưởng :**

Tương tự với các loại Boosting khác, BrownBoost sẽ bắt đầu tạo ra các Weak Learner , nhưng sẽ kiểm tra khi tổng hợp các weak learner, nếu số lượng dự đoán bị dự đoán sai quá Error Rate đạt mức chấp nhận đề ra ban đầu thì có chấp nhận, còn vượt quá Error Rate thì bỏ các dự đoán sai đó, tiến trình tiếp tục diễn ra với những dự đoán đúng, không bị ảnh hưởng bởi những dự đoán sai nữa.

### **Thuật toán BrownBoost**

* Input :

+ Trainning Set

+ Giá trị C dương (số lần lặp)

+ v > 0 (Hằng số nhỏ để tránh bị **Degenerate)**

* Initilizer : S = C
* Chạy thuật toán :

Do For i = 1,2,3,…

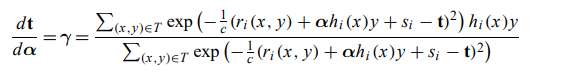
+ Với mõi dòng DataSet ta đặt trọng số dương là :

**Wi(x, y) =**

(với là khoảng cách từ điểm đó đến decision boundary)

**+** Tìm những hi(x) sao cho

+ Đặt các biến , α và t tuân theo phương trình vi phân sau :



( Với là hằng số,từ đó giải phương trình trên để tìm ra

= t\*>0 , thỏa điều kiện )

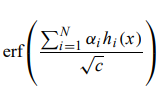
+ Cập nhật dự đoán cũng từng dòng :



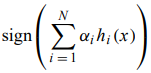
+ Cập nhật giá trị lặp còn lại của S : 

Until  **<= 0**

* Output :



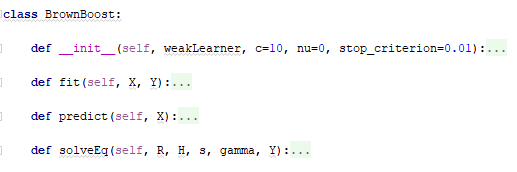
+ Nếu p(x) ∈ [-1,1] thì p(x) =



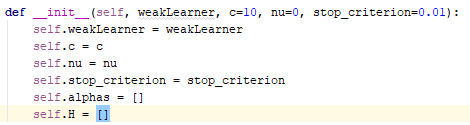
+ Nếu p(x) ∈ {-1,1} thì p(x) =

# Code Demo

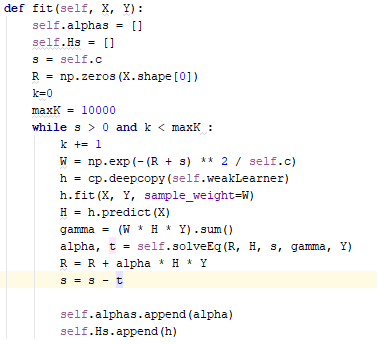
1. **Tạo đối tượng BrownBoost**



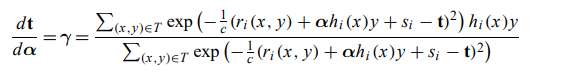
1. **Init cho BrownBoost**



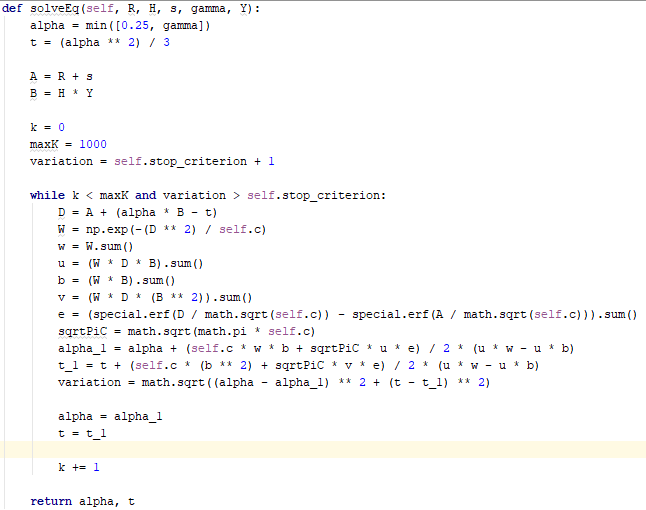
1. **Tạo hàm train cho BownBoost**



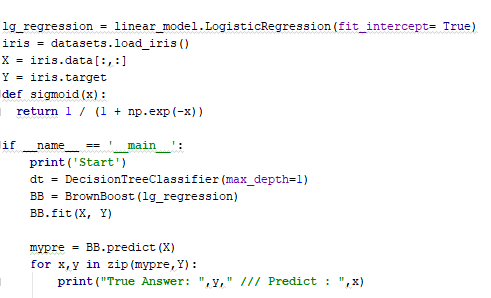
1. **Hàm xử lý phương trình :**



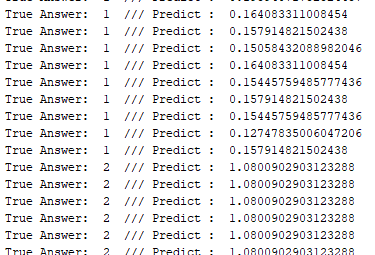
Để tính t và

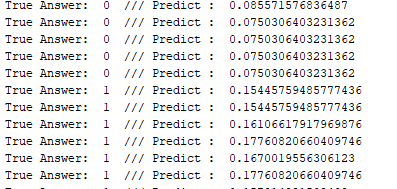


1. **Sử dụng Logistic Regression và iris DataSet để áp dụng BrownBoost**



1. **Kết Quả**





1. **Đánh giá thuật toán**

Thất Bại :

+ Không thể tự tạo Class BrownBoost

+ Phần dự đoán không ra đúng theo phân loại của DataSet iris (0,1,2)

# **Tài liệu tham khảo**

[1] Witten 2011 Data Mining Practical machine learning tools and techniques , (page 358)

[2] Retrieved : <https://becominghuman.ai/ensemble-learning-bagging-and-boosting-d20f38be9b1e?fbclid=IwAR2pv3GVcnvVDbSDT91DUvFFHO0NPPzM6XHztBS5bbdJoRtk3L56arcou_4>

Code Tham Khảo Tại :

<https://github.com/yannrs/BrownBoost>