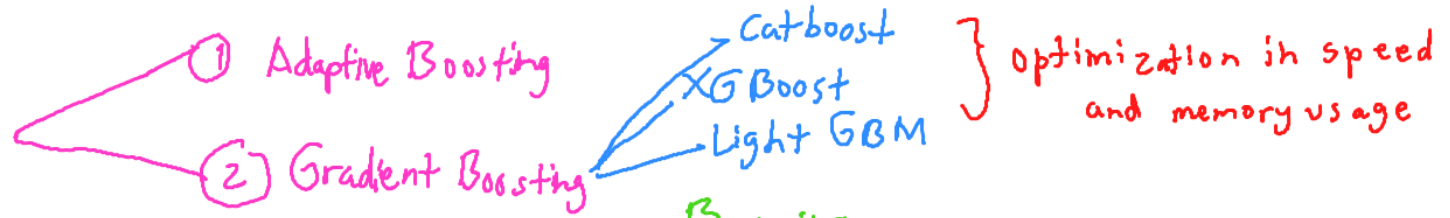
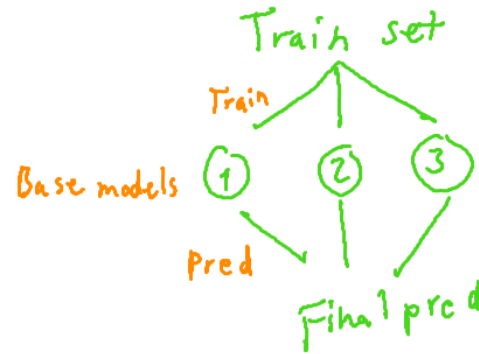
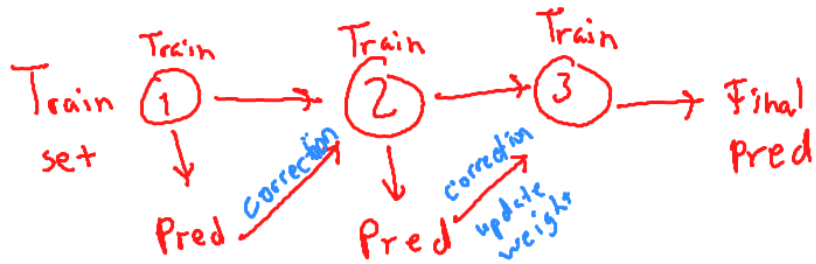


BOOSTING



Bagging



① Each data point that was incorrectly predicted by the previous model will be given more weight.

② Each base model will also be given a weight → model yang memiliki performa lebih baik akan mendapatkan bobot yang lebih tinggi

→ jumlah salah klasifikasi lebih sedikit.

Ada Boost

The default base estimator called STUMP \rightarrow max depth = 1
 \rightarrow only has 1 node & 2 child (leaf nodes)
 \downarrow
Hanya bekerja dengan 1 fitur

TRAINING

① Bobot untuk data point ke $i \rightarrow w^{(i)} = \frac{1}{m} \rightarrow$ jumlah data

② First base estimator training

③ Setelah selesai training & prediction, hitung cost function $r_j \rightarrow$ Bobot error model ke j

Jika base model pertama sudah selesai training, kita bisa menghitung seberapa banyak error yang dilakukan oleh tiap base model.

Menghitung bobot error model ke $j \leftarrow r_j =$

$$\frac{\sum_{i=1}^m w^{(i)} \mathbb{I}(\hat{y}_j^{(i)} \neq y^{(i)})}{\sum_{i=1}^m w^{(i)}}$$

$\left\{ \begin{array}{l} \text{jumlah bobot baris data yang salah diprediksi oleh model ke-} j \\ \text{karena baris data salah diprediksi} \end{array} \right\}$
 \rightarrow dibagi dengan jumlah bobot baris data

- ④ Hitung bobot model ke- $j \rightarrow \alpha_j \rightarrow$ Semakin besar α_j berarti semakin baik performa predictornya. and vice versa.

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

Predictor's weight

"eta" \rightarrow learning rate

semakin besar nilainya semakin besar pula perubahan yang terjadi

digunakan untuk mengatur seberapa banyak perubahan yang dilakukan.

- ⑤ Update $w^{(i)}$ \rightarrow update bobot tiap baris setelah mendapatkan nilai $\alpha_j \rightarrow$ Hanya data poin yang salah diprediksi yang akan diupdate bobotnya.

for $i = 1, 2, 3 \dots m$

$$w^{(i)} \rightarrow \begin{cases} w^{(i)} \\ w^{(i)} \exp(\alpha_j) \end{cases}$$

if $\hat{y}_j^{(i)} = y^{(i)} \rightarrow$ bobot data akan dibiarkan sama jika sudah diprediksi dengan benar

if $\hat{y}_j^{(i)} \neq y^{(i)} \rightarrow$ jika salah diprediksi, maka data poin tersebut akan diupdate bobotnya dengan menggunakan rumus yang tertera

⑥ Normalization → All the data points are normalized
(i.e. divided by $\sum_{i=1}^n w^{(i)}$)

$w^{(i)}$ → tiap data poin
 $\sum_{i=1}^n w^{(i)}$ → total bobot seluruh baris

⑦ Ulangi step 3-6 untuk base model selanjutnya.

Recap:

- ① Bobot kemungkinan sudah berubah pada saat memulai iterasi kedua. Jika semua baris diprediksi dengan benar, maka bobotnya akan tetap $\frac{1}{m}$.
- ② Train base model
- ③ Prediksi baris data dan lihat kesalahan prediksi yang dilakukan oleh model, lalu hitung r_j .
- ④ r_j digunakan untuk menghitung α_j
- ⑤ α_j akan digunakan untuk mengupdate bobot baris yang salah diprediksi.
- ⑥ Normalize all data points

PREDICT

① Setiap hasil prediksi dari tiap model akan digunakan kembali untuk final prediction (seperti bagging, various type, etc.)

② Kita dapat menghitung final prediction dengan cara:

$$\hat{y}(x) = \underset{k}{\operatorname{argmax}} \sum_{j=1}^N \alpha_j$$

α_j → the number of predictors / models

$\hat{y}(x) = k$ → kita akan mengambil kelas label k yang paling tinggi total bobot modelnya.

Dalam melakukan prediksi, AdaBoost secara simple hanya menghitung prediksi dari semua predictors dan kelas label yang akan diambil adalah yang paling tinggi bobotnya.

Illustration :

data point	α_j	1.2	1.7	0.2		
		Base model 1	Base model 2	Base model 3	<u>CLASS</u>	<u>FINAL PRED</u>
1		0	1	1	0 1	(1.2) (1.7+0.2) 1
2		1	0	1		(1.7) (1.2+0.2) 0
3		1	1	1	0	(1.2+1.7+0.2) 1

GRADIENT BOOSTING → Hanya berlaku untuk base model Decision Tree → GBRT
Gradient Boosting
Regression Tree

Perbedaan GB dengan AdaBoost adalah :

GB melakukan training pada error yang dihasilkan
oleh model sebelumnya.

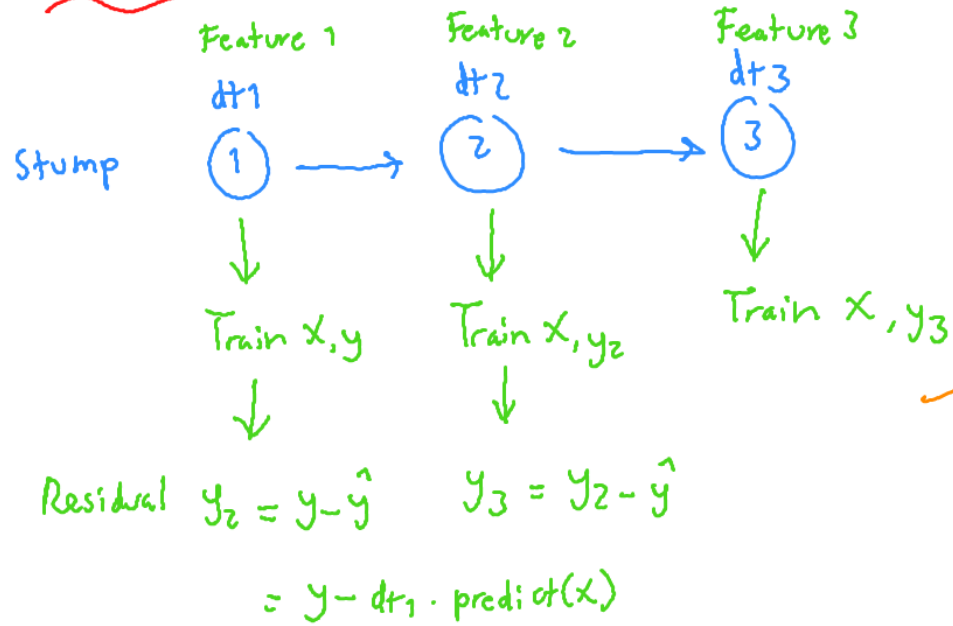
→ melihat errornya

↓
Minimize bias error

↓
Only use tree stump

→ The next model will be trained to residual of
the previous model

GB Regressor



Predict $(x_{\text{test}})^{(i)}$

$$\hat{y} = \sum (dt_1 \cdot \text{predict}(x_{\text{test}}) + dt_2 \cdot \text{predict}(x_{\text{test}}) + dt_3 \cdot \text{predict}(x_{\text{test}}))$$