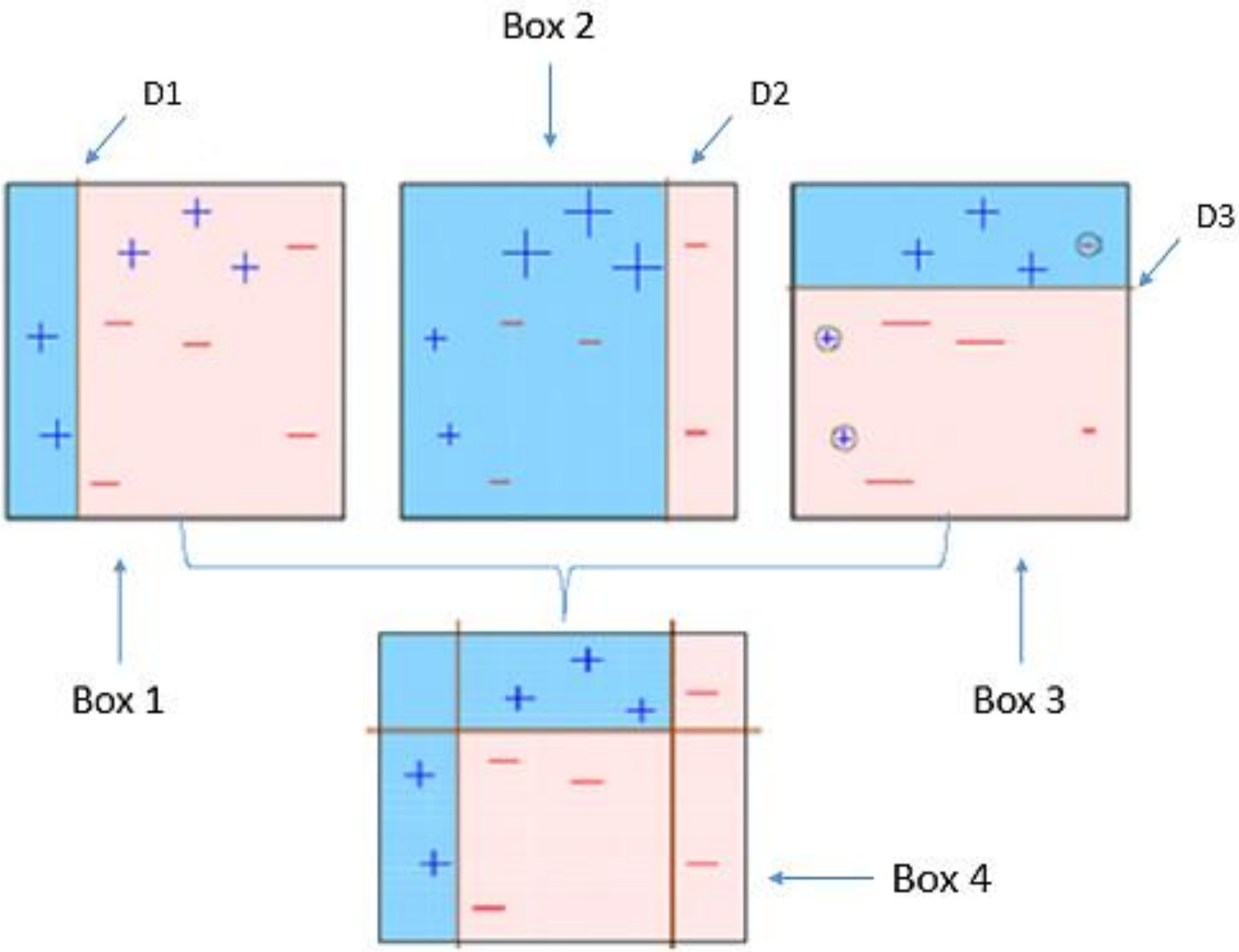
AdaBoost Classifier

A Journey through Pseudocode


Box 2
D1
D2
D3
Box 1
Box 3
Box 4

Ensembling

Ensembling is **a process where multiple diverse models are created, either by using many different modeling algorithms or using different training data sets**.

The ensemble model then aggregates the prediction of base models and results in once final prediction for the unseen data.

Ensembling

Bagging

- considers **homogeneous weak learners.**
- learns independently in **parallel.**
- **Combines them for overall prediction.**

Boosting

- considers **homogeneous weak learners.**
- learns **sequentially and adaptive manner.**.
- **Combines them for overall prediction.**

Stacking

- considers **heterogeneous weak learners.**
- Learns independently in **parallel.**
- **Combines them for overall prediction.**

AdaBoost Pseudocode walkthrough

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathscr{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathscr{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

https://hastie.su.domains/Papers/samme.pdf

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathscr{X}$, $y_i \in \{-1, +1\}$
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.

Here;

m = number of observations

(x_1, y_1) .. (x_m, y_m) refers to dataset consisting features (x) and label(y)

y ε {-1,1} Label can take value -1 or 1

$D_1(i) = 1/m$ – initialize weight of each observation with 1/m

For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathscr{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

Here a loop is initiated.

m = number of observations

(x_1, y_1) .. (x_m, y_m) refers to dataset consisting features (x) and label(y)

y ε {-1,1} Label can take value -1 or 1

$D_1(i)$ = 1/m – initialize weight of each observation with 1/m

h_t – build weak model (usually decision tree classifier)

- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

h_t – weak predictive model

ε_t – Estimate total error (sum of weights for rows with prediction error).

α_t – Amount of say

$D_{t+1}(i)$ – updated weight

Z_t – sum of weights

Output the final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

H(x) – overall prediction of T weak models

sign – if array value is greater than 0 it returns 1, if array value is less than 0 it returns -1, and if array value 0 it returns 0.

α_t – Amount of say of model ‘t’

$h_t(x)$ – predition of model ‘t’

Thank You