

Ada boost

🔥 Combiner des classificateurs faibles ?

Au lieu de s'appuyer sur un unique classifieur, AdaBoost construit une séquence de classifieurs faibles. À chaque itération, on porte une attention particulière aux exemples qui ont été mal classés lors des étapes précédentes.

L'algorithme :

- on initialise un poids uniforme sur chacun des N samples d'entraînement
- à chaque itération:
 - on **entraîne un nouveau weak classifieur** $h_t(x)$ en se basant sur ces poids (pour les séparer le mieux possible)
 - on calcule l'erreur pondérée du classifieur :

$$\varepsilon_t = \frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i I(h_t(x) \neq y_i)$$

- on **attribue ensuite un poids au classifieur** en fonction de son erreur. ça sera important plus tard quand on le fera voter (plus le taux d'erreur est faible, plus le α_t est important).

$$\alpha_t = \log \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

- on **met à jour les poids des samples** (uniquement ceux qui sont mal classifiés, les autres on les laisse comme avant!)

$$w_i = w_i \cdot \exp(\alpha_t I(y_i \neq h_t))$$

- Si $h_t(x_i) \neq y_i$ (mal classé), alors $\exp(\alpha_t \cdot 1) > 1$ augmente le poids.
- Après T itérations, on dispose de T classifieurs faibles h_1, h_2, \dots, h_T chacun associé à un poids $\alpha_1, \alpha_2, \dots, \alpha_T$.

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