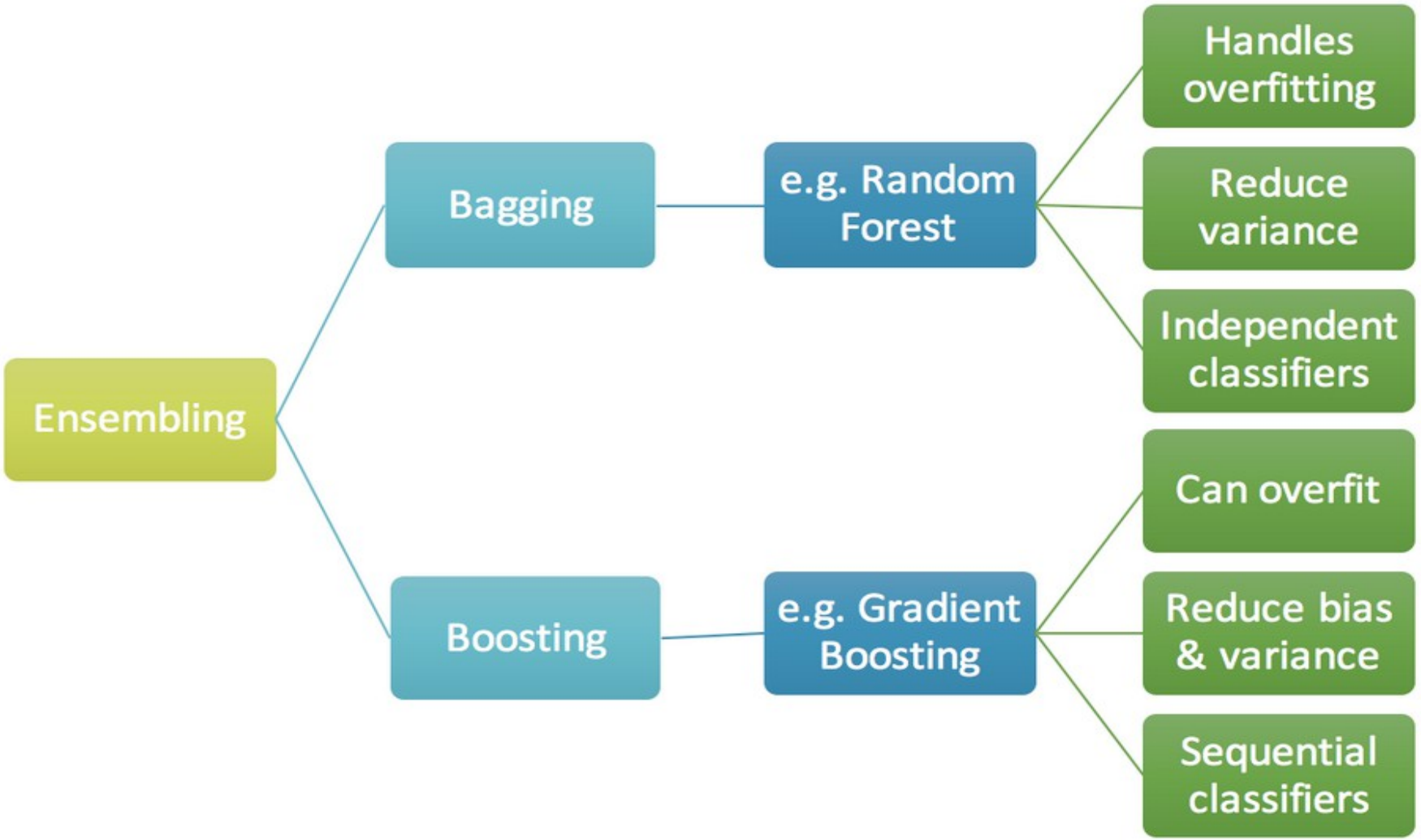
Boosting

Mikhail Lipkovich
06/26/2018

Ensembling


Ensembling
Bagging
e.g. Random Forest
Handles overfitting
Reduce variance
Independent classifiers
Boosting
e.g. Gradient Boosting
Can overfit
Reduce bias & variance
Sequential classifiers

ADABoost classifier

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

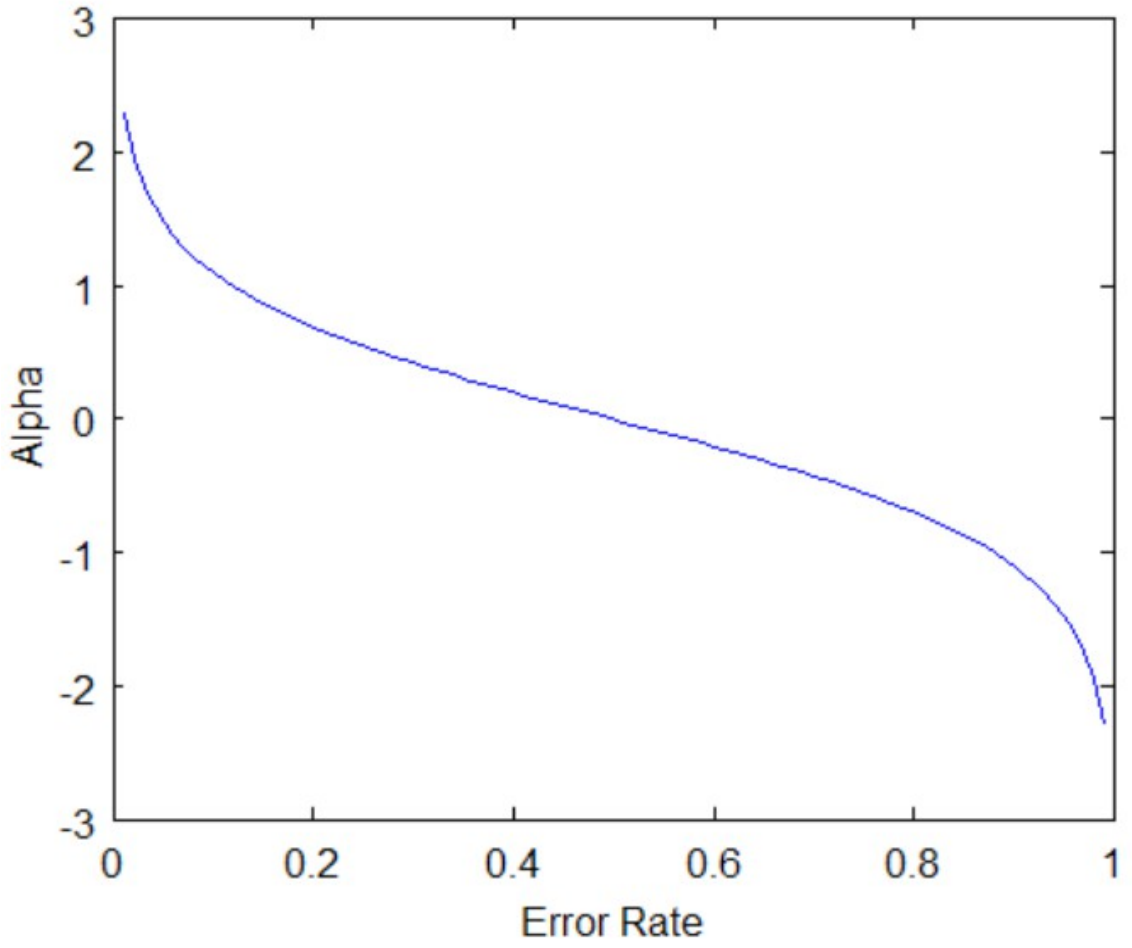
- T is amount of 'weak' classifiers
- h_t(x) is the output of 'weak' classifier 't'
- alpha_t is the weight applied to classifier 't'
- H(x) is the final classifier

ADABoost output weights

$$\alpha_t = \frac{1}{2} ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

- e_t is the proportion of misclassified samples by classifier 't'

ADABoost output weights



ADABoost samples weights

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

- 'i' is the training example number
- D_t(i) is the probability of training example 'i' appear in the training set for classifier 't'
- Z_t is the sum of all D_t

ADABoost algorithm

0. Set t=1
1. Initialize all D_t with ones
2. Train 'weak' classifier h_t(x)
3. Calculate its weight alpha_t
4. Calculate distribution D_{t+1}
5. Set t = t+1
6. Repeat steps 2-5 until get T classifiers
7. Resulting classifier is H(x)

Gradient Boosting

- Roughly speaking on each iteration we build a classifier which models error from the previous classifier
- There are several algorithms (GBM, XGBT)
- XGBoost is the most popular implementation with several optimizations