Boosting: a jump into the realm of slow learning

Filippo Biscarini (CNR, Milan, Italy)

filippo.biscarini@cnr.it

Combining predictors

Ensemble methods

- Like bagging and random forest, boosting is an **ensemble method**
- Multiple **(many!) models** are fitted to (many) **different versions of the data**
- Predictors (classifiers) are then **combined by averaging/voting** (majority vote)
- **Combining** predictors **improves accuracy** (usually at the expense of interpretability)

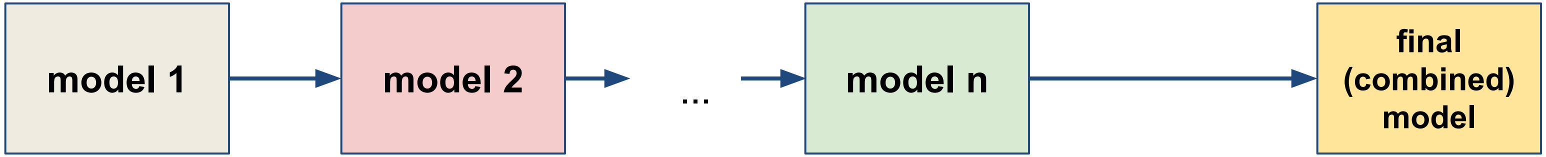
Ensemble methods

- Like bagging and random forest, boosting is an **ensemble method**
- Multiple **(many!) models** are fitted to (many) **different versions of the data**
- Predictors (classifiers) are then **combined by averaging/voting** (majority vote)
- **Combining** predictors **improves accuracy** (usually at t
interpretability)

Ensemble methods (deep learning, gradient boosting) dominate Kaggle competitions!

Boosting

- Bagging and Random Forest work on **bootstrapped copies** of the original data (and reduced samples of the features for RF)
- Bagging and RF work by building **many parallel** and independent **models**
- **Boosting** creates many **sequential models**: each model is built on information from the previous model



Boosting - origin

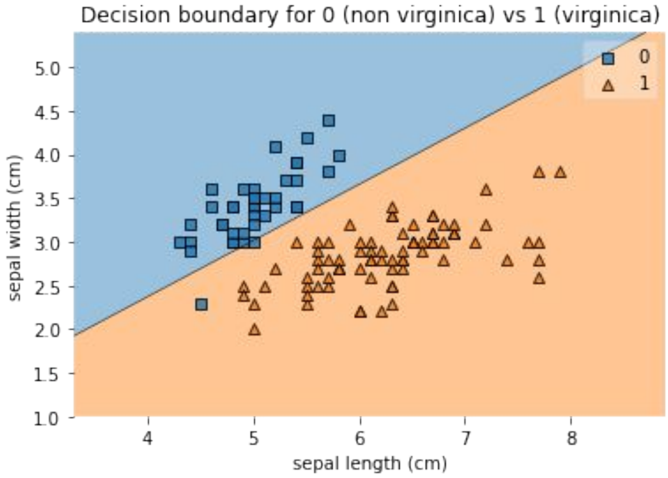
- Kearns and Valiant (1988, 1989): **"Can a set of weak learners create a single strong learner?"**
- **weak learner**: classifier that is only slightly correlated with the true classification (can label examples **just a bit better than random guessing**)
- Robert Schapire 1990 “The strength of weak learnability” → **yes, a set of weak learners can create a strong learner!**
- development of **boosting** (Breiman, 1998)

Boosting - intuition

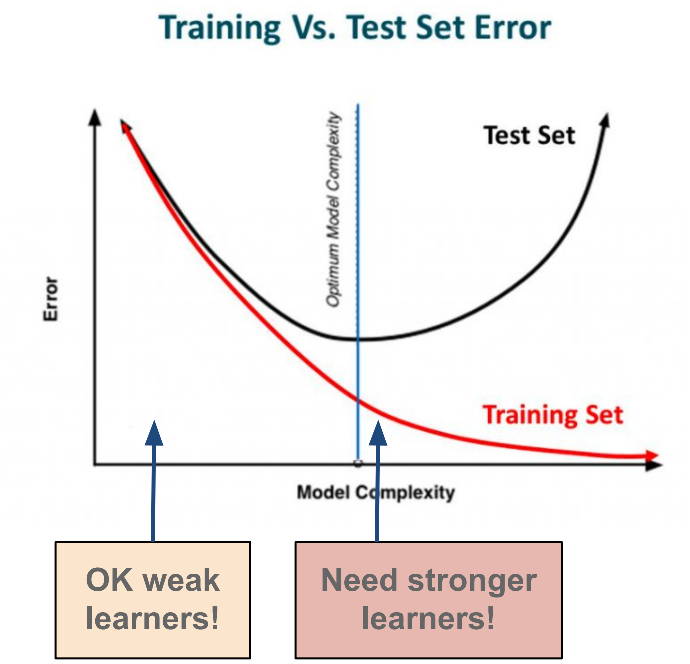
- case/controls: **5** completely **independent classifiers**, each with **70% accuracy**
- **majority vote**: if >= 3 out of 5 classifiers say “case”, we conclude it’s a case
- **combined accuracy** = **83.7%** (from the binomial CDF)
- with 101 (independent) classifiers → accuracy = 99.9%
- (if 101 models seems a lot, the Netflix Prize in 2009 was won by a combination of 107 predictors)

Simple (weak) learners

- e.g. multiple linear regression, logistic regression (with simple features)
- Tend to be very good (**high accuracy**)
- **Low variance → fast learning**
- But **bias is high** ...



Simple (weak) learners

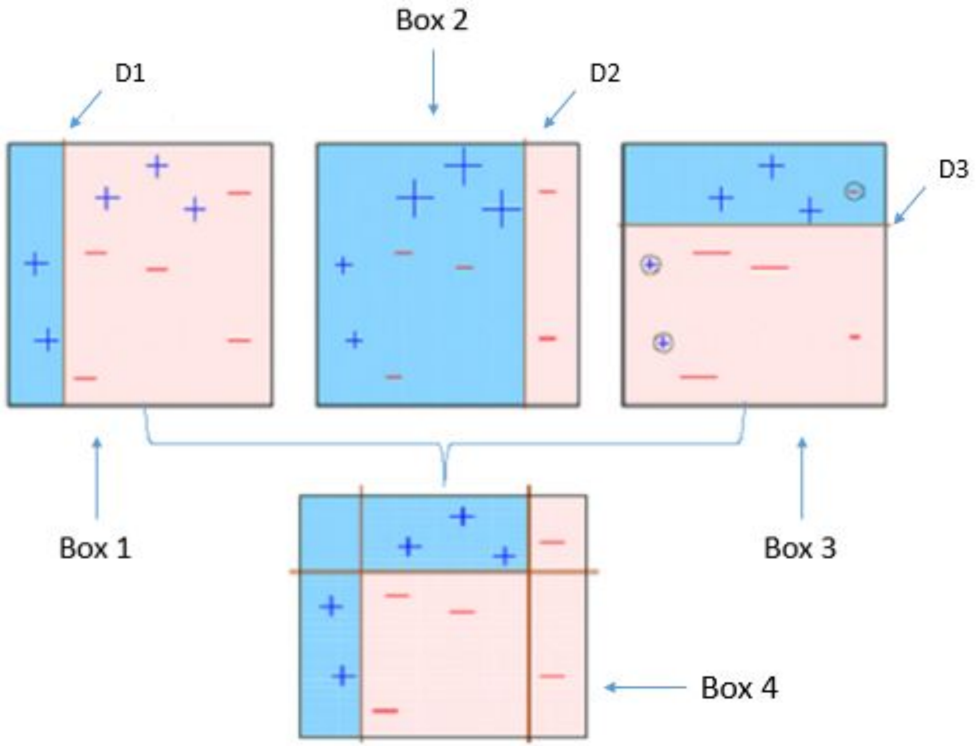


1. Add more features, depth, complexity to the “weak” learner
2. **Combine multiple weak learners**

The boosting way

- Can a set of weak learners be combined to create a stronger learner? [Kearns and Valiant, 1988]
- Yes! (Schapire, 1990) → **boosting**!
- Boosting can be used for both regression and classification problems

The boosting way

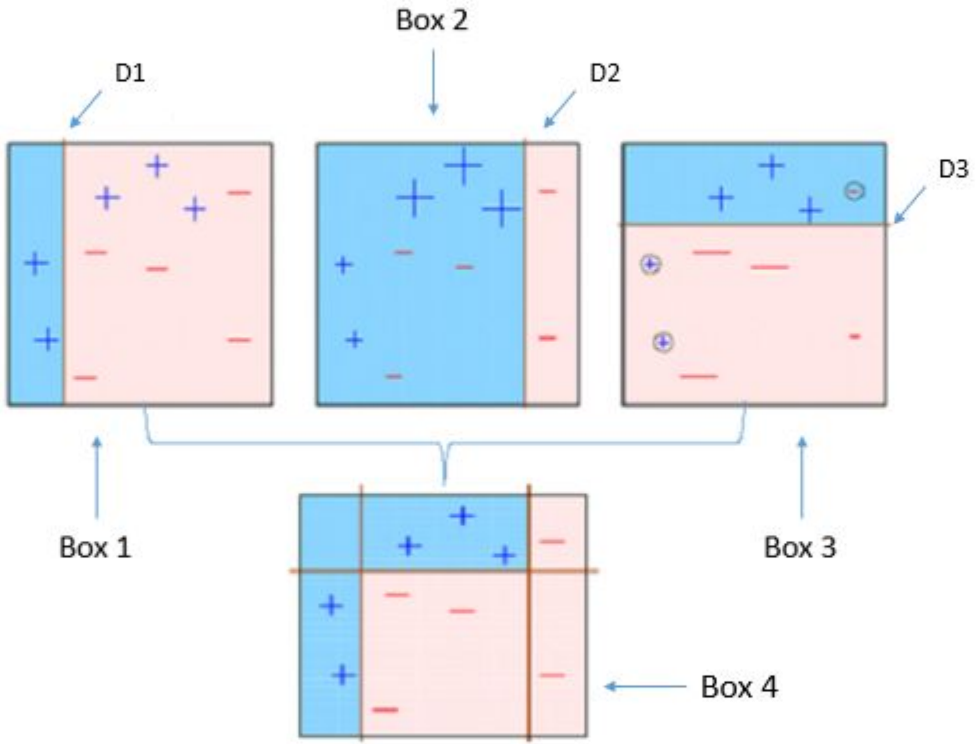


From: https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5

4 classification trees

- D1: first split makes three mistakes → errors will have a higher weight in the next model
- D2: makes three mistakes - higher weight in D3
- D3: classifies correctly the three mistakes from D2, no more mistakes
- D4: all models combined, perfect classification

The boosting way



From: https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5

4 classification trees

- D1: first split makes three mistakes → errors will have a higher weight in the next model
- D2: makes three mistakes - higher weight in D3
- D3: classifies correctly the three mistakes from D2, no more mistakes
- D4: all models combined, perfect classification

There is one obvious risk: can you guess which?

Boosting algorithm - a regression tree example

1. Initialize: $\hat{f}(x) = 0 \quad r_i = y_i$

2. b in 1, 2, … B

   a. Fit a regression tree with d (few) splits to the training data (X, r): $\rightarrow \hat{f}^b(x)$

   b. Update the classifier by a shrunken version of the new model:

   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

   c. Update residuals: $r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$

3. Final boosted model:

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

Boosting learns slowly

- in the <u>classification example</u>, we saw that sequential boosting models started from the **classification results of the previous model(s)**
- in the <u>regression example</u>, we saw that sequential boosting models started from **residuals from the previous model(s)**
- in both cases, individual sequential models are rather “shallow” (weak)
- rather than fitting a single large and complicated model to the data, multiple simpler models are fitted sequentially → **slow learning**

Boosting - hyperparameters

- number of trees, **B**: unlike bagging and RF, boosting can overfit (again, slowly!) if B is too large (trees are sequential, not parallel!)
- shrinkage parameter **lambda**: small positive number, if too slow many trees are needed
- model depth, **d**: e.g. n. of splits in trees (small d usually works well)

Boosting

- demonstration 10.boosting

→ 10.boosting.ipynb