XGBoost
Part 1: Overview of boosting

By: Noureddin Sadawi, PhD

University of London

Wisdom of the crowd

- In 1785 The French mathematician Marquis de Condorcet published his article titled:
  - 'Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.’
  - 'Essay on the application of analysis to the probability of decisions rendered by plurality of votes.’
- His idea was that the collective decision/view/opinion of several individuals can be more correct than that of a single expert.
- This concept is known as the ‘wisdom of the crowd’.

Example: Email classification

- It is easy to devise simple individual rules to decide whether an email is spam or not spam.
- Example: if the title contains ‘you have won’ or if it’s an empty message with a picture attached or whether it’s an email from a trusted domain (such as london.ac.uk).
- These rules are not powerful enough individually (we will call each of them a weak learner).
- It would be useful to use the weak learners collectively (i.e. combine their predictions).

Weak learner and boosting

Weak learner:

An algorithm which finds rules that are more than 50% accurate (i.e. better than random guessing).

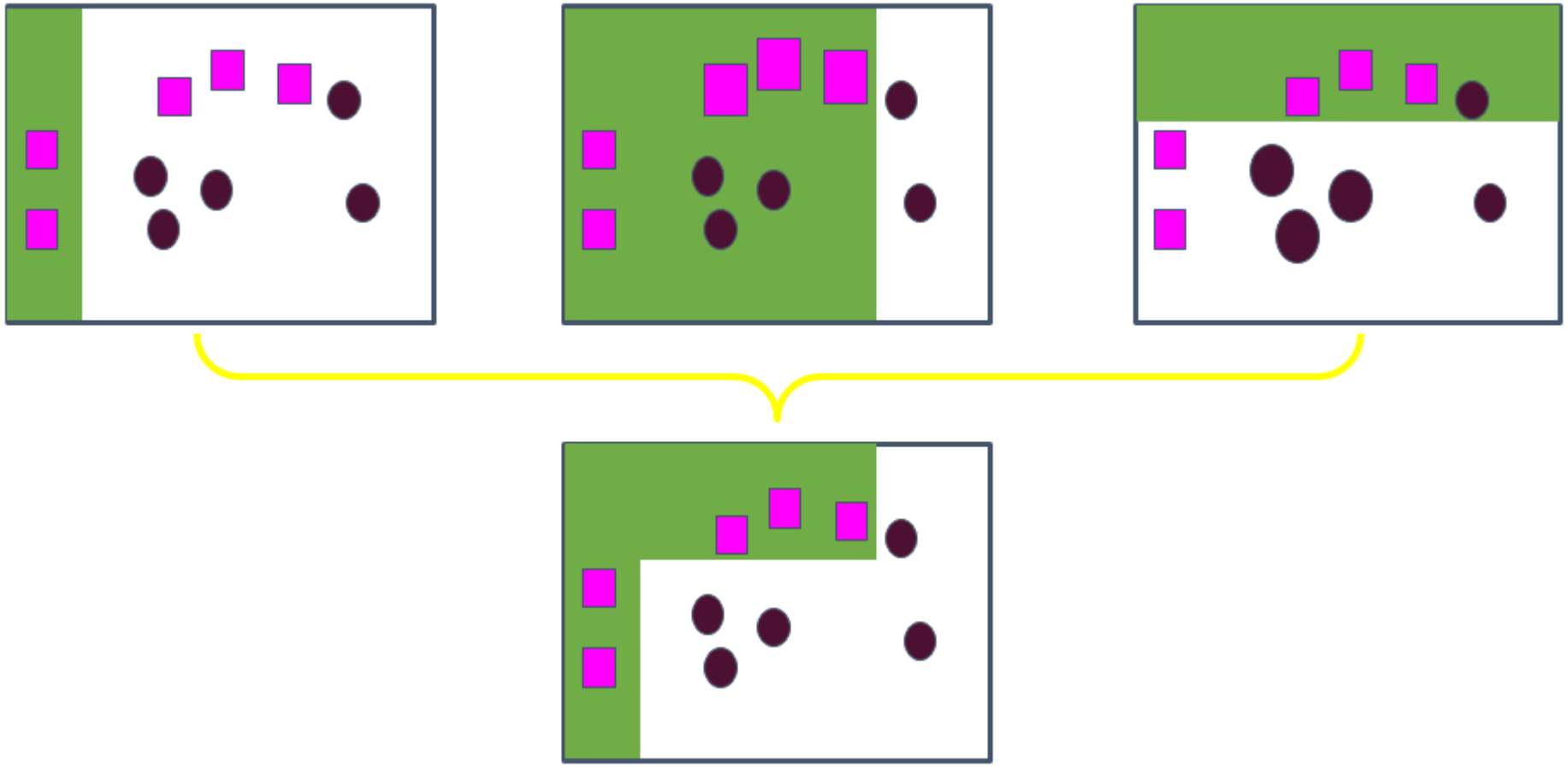
Boosting:

A method that combines weak learners to form a more accurate learner (e.g. classifier).

Boosting algorithm

- Step 1: The initial weak learner is trained on the data (initially it assigns equal weight to each data point).
- Step 2: Incorrectly predicted data points (by the base learner) are identified:
  - In the next iteration, more attention is paid to the incorrect predictions (e.g. by assigning them higher weights).
- Step 3: step 2 is repeated until a stopping condition is satisfied.
- The final model is a combination of all generated models.
- One way to combine is to use a weighted majority vote of the generated models.

Visual illustration



AdaBoost

- The core idea in Adaboost (Adaptive Boosting) is to iteratively train a weak learner on a distribution D on the training set.
- At each iteration more attention is paid to incorrectly classified examples (by assigning them larger weights).
- Larger weights mean the weak learner will focus mostly on those examples.

The final classifier is a weighted majority vote of the T base classifiers where α_t is the weight assigned to model h_t

$$F(x) = \sum_{t=1}^{T} \alpha_t h_t$$

Gradient boosting

- In Gradient boosting the overall model improves sequentially with each iteration (i.e. it makes sure that the current learner is better than the one before).
- Gradient boosting does not increment the weights of misclassified examples.
- Instead, it tries to optimise **the loss function** of the previous learner by adding a new learner.
- In other words, it views boosting as an optimisation problem.

Gradient boosting

- The gradient descent algorithm is used for optimisation.
- Each new model takes a step in the direction that minimises prediction error.
- The central idea here is to correct the errors in the predictions of the previous learner(s).
- It has three main components:
  - loss function that needs to be optimised
  - weak learner
  - a method to optimise the loss function.