The phylogenetic tree of boosting has a bushy carriage but a single trunk

Richard Nockk,b,1 and Frank Nielsenc

The phylogenetic tree of boosting has a bushy carriage, with early influencers: Ref. 1 was undoubtedly one for the making and use of popular application packages, some used nowadays in almost every Kaggle competition (2), and in that respect it is an understatement to say that the recent work of Luna et al. (3), embedded in the framework of interpretability, is important.

There is, however, a single trunk to this tree, and it grows from roots 1) in the probably approximately correct (PAC) model of Valiant (4), which led to 2) opening the boosting problem (5)—namely, how to use an oracle returning “weak” classifiers slightly better than random guessing to build arbitrarily accurate “strong” classifiers—followed by 3) an early formal breakthrough (6) and 4) a practical algorithm in this lineage, AdaBoost (7). As Luna et al. (3) point out, refs. 7 and 8 were the first to show how boosting could be formally achieved using decision trees (e.g., Classification and Regression Tree [CART] or C4.5) and linear combinations of classifiers (AdaBoost), respectively.

However, the claim, introduced in the paper’s abstract and developed at length in its body, that the paper discovers previously unknown connections between additive and full interaction models, at the representation and algorithmic levels, is inaccurate, in particular in the context of boosting—as originally designed from Valiant’s PAC model—and even in the context of CART vs. gradient boosted stumps. The unification of greedy boosting for decision trees and additive models in a single master algorithm was early shown in ref. 9, further resulting in a formal boosting algorithm able, for example, to switch the “weak/strong” roles of additive models and decision trees and thereby inducing oblique decision trees, more general than additive tree’s (AddTree’s) models. What ref. 9 did for a single loss (Matsushita’s loss, as used in ref. 8 for their optimal splitting criterion) was later generalized, in ref. 10, to all proper canonical and symmetric losses satisfying mild differentiability assumptions, a set to which belongs the splitting criteria used by CART and C4.5. The formal connection via the loss is simple: The splitting criteria used in decision tree induction like CART or C4.5 and the ones optimized by additive approaches like LogitBoost are dual from a convex optimization standpoint. However, this goes beyond the loss: While the algorithmic approaches look different at the pseudocode level—additive modeling does tune during induction a “memory” (weights) that the greedy induction of decision trees does not—they are in fact equivalent and the trick used to reduce one on to the other exploits the same representational trick as the one depicted in figure 1 of ref. 3.

An important novelty in Luna et al. (3) is the first-order training with weights that adjust the model between additive and full interaction. We have no doubt that it will contribute to spread further boosting. However, boosting being both a trusted companion to interpretability and one of the paper’s clear grounds, it would come as desirable a formal understanding of AddTree in this framework.


*Data61, Commonwealth Scientific and Industrial Research Organisation, Eveleigh NSW 2015, Australia; Research School of Computer Science, Australian National University, Canberra ACT 0200, Australia; and Sony Computer Science Laboratories, Inc., Higashigotanda, 3 Chome-14-13, Tokyo, Japan

Author contributions: R.N. and F.N. wrote the paper.

The authors declare no competing interest.

This open access article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

1To whom correspondence may be addressed. Email: Richard.Nock@data61.csiro.au.

First published April 7, 2020.