Knowledge-Guided Additive Modeling For Supervised Regression

Yann Claes[1] **Vân Anh Huynh-Thu**[1] **Pierre Geurts**[1]

Abstract

Several hybrid approaches, incorporating prior domain knowledge within machine learning (ML), have recently been introduced to improve generalization and robustness. However, such hybrid methods were mostly tested on dynamical systems, with only limited study of the influence of each model component on global performance and parameter identification. In this work, we assess the performance of hybrid modeling on standard regression problems: we compare, on synthetic problems, several approaches for training such hybrid models, focusing on model-agnostic methods that additively combine a parametric physical term with an ML term. We also introduce a new hybrid approach based on partial dependence functions. Experiments are carried out with different types of ML models, including tree-based models and neural networks.

1. Introduction

Recently, hybrid approaches have been introduced to incorporate prior domain knowledge within machine learning (ML) models to improve generalization and robustness of purely data-driven ML approaches (Daw et al., 2017; De Bézenac et al., 2019; Yin et al., 2021b). Their success has been shown empirically on a range of synthetic and real-world problems (Yin et al., 2021b; Ayed et al., 2019; Mehta et al., 2021; Donà et al., 2022). However, these models were mostly evaluated on dynamical problems, using neural networks as ML models, leaving aside other methods.

In this work, we empirically study the benefits of hybrid methods against data-driven methods on standard *static* regression problems (as opposed to dynamical problems). We focus on hybrid models that additively combine a parametric physical term with an ML term, which can be of any type, and relate differences in terms of prediction and parameter identification performance. We also introduce a new hybrid approach based on partial dependence functions.

Related works. Hybrid models combining additively an algebraic term h_k with an ML model h_a have emerged in various domains, massively relying on neural networks and applied to dynamical problems (Yin et al., 2021a; Takeishi & Kalousis, 2021; Wehenkel et al., 2022; Donà et al., 2022). In a more standard regression setting, Zhang et al. (2019) combined random forests with a linear parametric term. Elements of discussion about the well-posedness of the additivity hypothesis have been introduced in previous works. Yin et al. (2021b) show the existence and uniqueness of an optimal pair (h_k, h_a), when the contributions of h_a are constrained to be minimal. Donà et al. (2022) demonstrate the convergence of an algorithm alternating between the optimization of h_k and the optimization of h_a, without however any guarantee about convergence points.

2. Problem statement

Let us define a regression problem, with $y \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^d$, with $d \in \mathbb{N}_+$, drawn from a distribution $p(\mathbf{x}, y)$ such that $y = f(\mathbf{x}) + \varepsilon$ with $f : \mathbb{R}^d \mapsto \mathbb{R}$ the partially known generating function and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ the noise term. We focus on problems such that $f(\mathbf{x})$ can be decomposed as:

Hypothesis 1 (H1, Additivity).

$$y = f_k(\mathbf{x}_k) + f_a(\mathbf{x}) + \varepsilon,$$

where $\mathbf{x}_k$ is a subset of $K \leq d$ input variables. We assume partial knowledge through some known algebraic function $h_k^{\theta_k}(\mathbf{x}_k) \in \mathcal{H}_k$ with tunable parameters θ_k, such that for the optimal parameters θ_k^* we have $h_k^{\theta_k^*} = f_k$. The residual term $f_a(\mathbf{x})$ is unknown and is approximated in this work through an ML component $h_a^{\theta_a} \in \mathcal{H}_a$, with parameters θ_a[1]. The final model $h \in \mathcal{H}$ is denoted $h(\mathbf{x}) = h_k^{\theta_k}(\mathbf{x}_k) + h_a^{\theta_a}(\mathbf{x})$, with the function space $\mathcal{H}$ defined as $\mathcal{H}_k + \mathcal{H}_a$. H1 is common when model-based methods and ML models are

[1] Department of Electrical Engineering and Computer Science, University of Liège, 4000, Belgium. Correspondence to: Yann Claes <y.claes@uliege.be>, Vân Anh Huynh-Thu <vahuynh@uliege.be>, Pierre Geurts <p.geurts@uliege.be>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023. Copyright 2023 by the author(s).

[1] In the following, $h_k^{\theta_k}$ and $h_a^{\theta_a}$ will sometimes be denoted simply as h_k and h_a to lighten the notations.

combined (Takeishi & Kalousis, 2021; Yin et al., 2021a; Donà et al., 2022; Wehenkel et al., 2022).

Given a learning sample of N input-output pairs $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, drawn from $p(\mathbf{x}, y)$, we seek to identify a function $h = h_k^{\theta_k} + h_a^{\theta_a}$, i.e. parameters θ_k and θ_a, that minimizes the following two distances:

$$d(h, y) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}\{(h(\mathbf{x}) - y)^2\}, \tag{1}$$

$$d_k(h_k^{\theta_k}, f_k) = \mathbb{E}_{\mathbf{x}_k\sim p(\mathbf{x}_k)}\{(h_k^{\theta_k}(\mathbf{x}_k) - f_k(\mathbf{x}_k))^2\}. \tag{2}$$

The first distance measures the standard generalization error of the global model h. The hope is that taking h_k into account will help learning a better global model than fitting directly h_a on y. The second distance d_k measures how well the tuned h_k approximates f_k. An alternative to d_k is the distance between the estimated and optimal parameters $\hat{\theta}_k$ and θ_k^* (e.g., $||\hat{\theta}_k - \theta_k^*||^2$). d_k however has the advantage not to require θ_k^* to be fully identifiable, i.e. there can exist several sets of parameters θ_k^* such that $h_k^{\theta_k^*} = f_k$.

Minimizing the distance in (2) is expected to be challenging and sometimes even ill-posed. Indeed, if h_a is too powerful, it could capture f entirely and leave little room for the estimation of f_k. Finding the right balance between h_k and h_a is thus very challenging, if not impossible, using only guidance of the learning sample LS. Unlike (1), (2) cannot be estimated from a sample of input-output pairs and hence cannot be explicitly used to guide model training.

In the following, we will discuss the optimality of the hybrid methods under two additional assumptions:

Hypothesis 2 (H2, Disjoint features). Let $\mathbf{x}_a$ be a subset of features disjoint from $\mathbf{x}_k$ ($\mathbf{x}_k \cap \mathbf{x}_a = \emptyset$). There exists a function $f_a^r(\mathbf{x}_a)$ such that $f_a(\mathbf{x}) = f_a^r(\mathbf{x}_a)$ for all $\mathbf{x}$.

Hypothesis 3 (H3, Independence). Features in $\mathbf{x}_k$ are independent from features in $\mathbf{x}_a$ ($\mathbf{x}_k \perp\!\!\!\perp \mathbf{x}_a$).

H2 makes the problem easier as f_k now captures all the dependence of y on $\mathbf{x}_k$. In the absence of H3, it might be hard to distinguish real contributions from $\mathbf{x}_k$ to f from those due to correlations with features not in $\mathbf{x}_k$.

3. Methods

We focus on model-agnostic approaches, i.e. that can be applied with any algebraic function h_k and any type of ML model h_a. For both terms, we only assume access to training functions, respectively denoted fit^{h_k}, $\text{fit}^{h_k+\gamma}$, and fit^{h_a}, that can estimate each model parameters, respectively θ_k, (θ_k, γ) and θ_a, so as to minimize the mean squared error (MSE) over LS (see below for the meaning of γ), where parametric methods rely on gradient descent.

3.1. Sequential training of h_k and h_a

This baseline approach first fits h_k on the observed output y, then fits h_a on the resulting residuals. More specifically, we train h_k on y by introducing a constant $\gamma \in \mathbb{R}$, such that $(\hat{\theta}_k, \hat{\gamma}) = \text{fit}^{h_k+\gamma}(LS)$. Afterwards, we fit h_a on the output residuals: $\hat{\theta}_a = \text{fit}^{h_a}\{(\mathbf{x}_i, y_i - h_k^{\hat{\theta}_k}(\mathbf{x}_i) - \hat{\gamma})\}_{i=1}^N$.

Let $\hat{\mathcal{F}}_k$ be the set of all functions $\hat{f}_k$ mapping $\mathbf{x}_k \in \mathcal{X}_k$ to some value $y \in \mathbb{R}$, i.e. $\hat{\mathcal{F}}_k = \{\hat{f}_k : \mathcal{X}_k \mapsto \mathbb{R}\}$. Under H2 and H3, it can be shown that $\hat{f}_k^* = \arg\min_{\hat{f}_k \in \hat{\mathcal{F}}_k} d(\hat{f}_k, y)$ is such that $\hat{f}_k^*(\mathbf{x}_k) = f_k(\mathbf{x}_k) + C$, for every $\mathbf{x}_k \in \mathcal{X}_k$, with $C = \mathbb{E}_{\mathbf{x}_a}\{f_a^r(\mathbf{x}_a)\}$. Hence, this approach is sound at least asymptotically. Note however that even under H2 and H3, we have no guarantee that this approach produces the best estimator for a finite sample size, as $f_a^r(\mathbf{x}_a) + \epsilon$ acts as a pure additive noise term.

3.2. Alternate training of h_k and h_a

Donà et al. (2022) proposed a hybrid additive approach that alternates between a training step (i.e., one epoch) on h_k and one on h_a (using neural networks for h_a). We include this approach in our comparison, but also investigate it with random forests (Breiman, 2001) and tree gradient boosting (Friedman, 2002). $\hat{\theta}_k$ is initialized by (fully) fitting $h_k^{\theta_k} + \gamma$ on y. Then, we alternate a single epoch on $h_k^{\theta_k} + \gamma$ with a single epoch for neural networks (as in Donà et al., 2022) or a complete fit of h_a for tree-based models.

While some theoretical results are provided by Donà et al. (2022), convergence of the alternate method towards the optimal solution is not guaranteed in general. Despite an initialization favoring h_k, it is unclear whether a too expressive h_a will not dominate h_k and finding the right balance between these two terms, e.g. by regularizing further h_a, is challenging. Under H2 and H3 however, the population version of the algorithm produces an optimal solution. Indeed, h_k will be initialized as the true f_k, as shown previously, making the residuals $y - h_k$ at the first iteration, as well as h_a, independent of $\mathbf{x}_k$. The h_k will thus remain unchanged (and optimal) at subsequent iterations.

3.3. Partial Dependence-based training of h_k and h_a

We propose a novel approach relying on partial dependence (PD) plots (Friedman, 2001) to produce a proxy dataset depending only on $\mathbf{x}_k$ to fit h_k. PD measures how some features impact the prediction of a model, on average. Let $\mathbf{x}_k$ be the subset of interest and $\mathbf{x}_{-k}$ its complement, with $\mathbf{x}_k \cup \mathbf{x}_{-k} = \mathbf{x}$, then the PD of a function $f(\mathbf{x})$ on $\mathbf{x}_k$ is:

$$PD(f, \mathbf{x}_k) = \mathbb{E}_{\mathbf{x}_{-k}}\left[f(\mathbf{x}_k, \mathbf{x}_{-k})\right]. \tag{3}$$

Algorithm 1 Partial Dependence Optimization

```
**Input:** LS = $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$
$\hat{\theta}_a \leftarrow \text{fit}^{h_a}(\text{LS})$
$(\hat{\theta}_k, \hat{\gamma}) \leftarrow \text{fit}^{h_k+\gamma}(\{(\mathbf{x}_{k,i}, \widehat{PD}(h_a^{\hat{\theta}_a}, \mathbf{x}_{k,i}; \text{LS}))\}_{i=1}^N)$
**for** $n = 1$ **to** $N_{repeats}$ **do**
  $\hat{\theta}_a \leftarrow \text{fit}^{h_a}(\{(\mathbf{x}_i, y_i - h_k^{\hat{\theta}_k}(\mathbf{x}_{k,i}) - \hat{\gamma})\}_{i=1}^N)$
  $(\hat{\theta}_k, \hat{\gamma}) \leftarrow$
      $\text{fit}^{h_k+\gamma}(\{(\mathbf{x}_{k,i}, h_k^{\hat{\theta}_k}(\mathbf{x}_{k,i}) + \hat{\gamma} + \widehat{PD}(h_a^{\hat{\theta}_a}, \mathbf{x}_{k,i}; \text{LS}))\}_{i=1}^N)$
**end for**
$\hat{\theta}_a \leftarrow \text{fit}^{h_a}(\{(\mathbf{x}_i, y_i - h_k^{\hat{\theta}_k}(\mathbf{x}_{k,i}) - \hat{\gamma})\}_{i=1}^N)$
```

Under H1 and H2, Friedman (2001) shows that the PD of $f(x) = f_k(\mathbf{x}_k) + f_a^r(\mathbf{x}_a)$ is:

$$PD(f, \mathbf{x}_k) = f_k(\mathbf{x}_k) + C, \text{ with } C = E_{\mathbf{x}_a}\{f_a^r(\mathbf{x}_a)\}. \quad (4)$$

The idea of our method is to first fit any sufficiently expressive ML model $h_a(\mathbf{x})$ on LS and to compute its PD w.r.t. $\mathbf{x}_k$ to obtain a first approximation of $f_k(\mathbf{x}_k)$ (up to a constant). Although computing the actual PD of a function using (3) requires in principle access to the input distribution, an approximation can be estimated from LS as follows:

$$\widehat{PD}(h_a, \mathbf{x}_k; LS) = \frac{1}{N}\sum_{i=1}^{N} h_a(\mathbf{x}_k, \mathbf{x}_{i,-k}). \quad (5)$$

A new dataset of pairs $(\mathbf{x}_k, \widehat{PD}(h_a, \mathbf{x}_k; LS))$ can be built to fit h_k. In our experiments, we consider only the $\mathbf{x}_k$ values in the learning sample but $\widehat{PD}(h_a, \mathbf{x}_k; LS)$ could also be estimated at other points $\mathbf{x}_k$ to artificially increase the size of the proxy dataset. In practice, optimizing θ_k only once on the PD of h_a could leave residual dependence of $\mathbf{x}_k$ on the resulting $y - h_k^{\hat{\theta}_k}(\mathbf{x}_k) - \hat{\gamma}$. We thus repeat the sequence of fitting h_a on the latter residuals, then fitting h_k on the obtained $\widehat{PD}(h_a^{\hat{\theta}_a}, \mathbf{x}_k; LS) + h_k^{\hat{\theta}_k}(\mathbf{x}_k) + \hat{\gamma}$, with $\hat{\theta}_k$ and $\hat{\theta}_a$ the current optimized parameter vectors (see Algorithm 1).

The main advantage of this approach over the alternate one is to avoid domination of h_a over h_k. Unlike the two previous approaches, this one is also sound even if H3 is not satisfied as it is not a requirement for (4) to hold. One drawback is that it requires h_a to capture well the dependence of f on $\mathbf{x}_k$ so that its PD is a good approximation of f_k. The hope is that even if it is not the case at the first iteration, fitting h_k, that contains the right inductive bias, will make the estimates better and better over the iterations.

4. Experiments

We investigate the performance of all methods on simulated regression datasets, through estimates of (1) and (2) on a test set TS, respectively denoted $\hat{d}(h, y; TS)$ and $\hat{d}_k(h_k^{\theta_k}, f_k; TS)$. We also report $\text{rMAE}(\theta_k^*, \theta_k)$, the relative mean absolute distance between θ_k^* and θ_k (lower is better for all measures). For the hybrid approaches, we use as h_a either a multilayer perceptron (MLP), gradient boosting with decision trees (GB) or random forests (RF). We compare these hybrid models to a standard data-driven model that uses only h_a. We also compare fitting h_a with and without input filtering. Filtering consists in removing $\mathbf{x}_k$ from the set of inputs $\mathbf{x}$ fed to the ML model h_a, to fully exploit H2. This allows us to verify convergence claims about h_k in Section 3.2. Architectures (e.g. for MLP, the number of layer and neurons) are kept fixed across training methods to allow a fair comparison between them.

Table 1. Results on Friedman problem. We report the mean and standard deviation of $\hat{d}$ and $\hat{d}_k$ over the 10 test sets (TS). "$f_k \rightarrow h_a$" is the ideal approach that fits h_a on $y - f_k(\mathbf{x}_k)$.

		$\hat{d}(h, y; TS)$		$\hat{d}_k(h_k^{\theta_k}, f_k; TS)$	
	Method	Unfiltered	Filtered	Unfiltered	Filtered
MLP	$f_k \rightarrow h_a$	1.58 ± 0.33	1.23 ± 0.10	-	
	Sequential	1.54 ± 0.31	1.43 ± 0.13	0.18 ± 0.16	
	Alternate	1.43 ± 0.09	1.32 ± 0.09	0.10 ± 0.09	0.02 ± 0.02
	PD-based	1.54 ± 0.12	1.38 ± 0.09	0.06 ± 0.07	
	h_a only	2.62 ± 0.75		-	
GB	$f_k \rightarrow h_a$	1.73 ± 0.09	1.75 ± 0.12	-	
	Sequential	1.74 ± 0.11	1.81 ± 0.14	0.18 ± 0.16	
	Alternate	1.79 ± 0.11	1.78 ± 0.15	0.91 ± 1.45	0.06 ± 0.06
	PD-based	1.77 ± 0.13	1.78 ± 0.12	0.03 ± 0.02	
	h_a only	3.43 ± 0.94		-	
RF	$f_k \rightarrow h_a$	2.03 ± 0.18	1.96 ± 0.17	-	
	Sequential	2.11 ± 0.23	2.05 ± 0.24	0.18 ± 0.16	
	Alternate	2.03 ± 0.19	1.98 ± 0.17	0.04 ± 0.03	0.04 ± 0.04
	PD-based	2.16 ± 0.27	2.09 ± 0.26	0.16 ± 0.15	
	h_a only	5.58 ± 1.91		-	

4.1. Friedman problem (H2 and H3 satisfied)

We consider the following synthetic regression problem:

$$y = \theta_0 \sin(\theta_1 x_0 x_1) + \theta_2 (x_2 - \theta_3)^2 + \theta_4 x_3 + \theta_5 x_4 + \varepsilon,$$

where $\mathbf{x} \sim \mathcal{U}(0,1)$ and $\varepsilon \sim \mathcal{N}(0,1)$ (Friedman et al., 1983). We generate 10 different datasets using 10 different sets of values for $\theta_0, \ldots, \theta_5$. For the hybrid approaches, we use the first term as prior knowledge, i.e. $f_k = \theta_0 \sin(\theta_1 x_0 x_1)$.

We see in Table 1 that all hybrid training schemes outperform their data-driven counterpart. They come very close to the ideal $f_k \rightarrow h_a$ method, and are sometimes even slightly better, probably due to slight overfitting issues. Sequential fitting of h_k and h_a performs as well as the alternate or PD-based approaches, as H2 and H3 are satisfied for this problem (see Section 3.1). Filtering generally improves performance of hybrid schemes as H2 is verified. PD-based optimization yields good approximations of f_k (as shown by a low $\hat{d}_k$). The alternate approach follows closely whereas the sequential one ends up last, which can be expected as fitting h_k only on y induces a higher noise level centered around $\mathbb{E}_{\mathbf{x}_a}\{f_a^r(\mathbf{x}_a)\}$, while the other approaches benefit from reduced perturbations through h_a estimation, as explained in Section 3.1. Filtering vastly decreases $\hat{d}_k$ for alternate approaches, supporting claims introduced in Section 3.2, while this measure remains unimpaired for sequential and PD-based training by construction.

4.2. Correlated input features (H3 not satisfied)

Correlated linear model. Let $y = \beta_0 x_0 + \beta_1 x_1 + \varepsilon$, with $\beta_0 = -0.5, \beta_1 = 1, \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, and $\varepsilon \sim \mathcal{N}(0.5^2, 1)$. We use as known term $f_k(\mathbf{x}_k) = \beta_0 x_0$. Regressing y on x_0 yields the least-squares solution (Greene, 2003):

$$\mathbb{E}\left[\hat{\beta}_0\right] = \beta_0 + \frac{\operatorname{cov}(x_0, x_1)}{\operatorname{var}(x_0)}\beta_1. \tag{6}$$

We set $\operatorname{cov}(x_0, x_1) = 2.25$ and $\operatorname{var}(x_0) = 2$ so that (6) reverses the sign of β_0. The sequential approach should hence yield parameter estimates of β_0 close to (6) while we expect the others to correct for this bias.

From Table 2, we observe that, contrary to the PD-based approach, the sequential and alternate methods return very bad estimations of β_0, as H3 is no longer verified. Filtering corrects the bias for the alternate approach but degrades the MSE performance for the sequential method as it removes the ability to compensate for the h_k misfit.

Table 2. Results for the correlated linear problem. We report $\hat{d}$ and $\operatorname{rMAE}(\beta_0^*, \hat{\beta}_0)$, over 10 different datasets.

	$\hat{d}(h, y; TS)$		$\operatorname{rMAE}(\beta_0^*, \hat{\beta}_0)$	
Method	Unfiltered	Filtered	Unfiltered	Filtered
Seq. (MLP)	0.30 ± 0.03	0.74 ± 0.09	224.14 ± 13.48	
Alt. (MLP)	0.30 ± 0.02	0.31 ± 0.04	186.65 ± 21.31	15.53 ± 13.57
PD (MLP)	0.30 ± 0.03	0.29 ± 0.02	26.47 ± 17.32	
Seq. (GB)	0.59 ± 0.06	1.38 ± 0.11	224.14 ± 13.48	
Alt. (GB)	0.57 ± 0.06	0.60 ± 0.09	148.75 ± 67.35	24.58 ± 12.20
PD. (GB)	0.56 ± 0.05	0.64 ± 0.13	36.05 ± 17.50	
Seq. (RF)	0.53 ± 0.05	0.90 ± 0.07	224.14 ± 13.48	
Alt. (RF)	0.43 ± 0.04	0.42 ± 0.04	111.04 ± 52.78	45.38 ± 22.39
PD (RF)	0.41 ± 0.03	0.43 ± 0.04	57.47 ± 15.55	

Correlated Friedman problem. The structure is identical to Section 4.1 but with correlated inputs drawn from a multivariate normal distribution where $\mu_i = 0.5$ and $\operatorname{var}(x_i) = 0.75, \forall i$, and $\operatorname{cov}(x_i, x_j) = \pm 0.3, \forall i \neq j$ (the covariance sign being chosen randomly). Inputs are then scaled to be roughly in $[-1, 1]$. Here again, we use $f_k = \theta_0 \sin(\theta_1 x_0 x_1)$.

As in Section 4.1, Table 3 shows that hybrid models outperform their data-driven equivalents. PD-based methods usually yield more robust h_k estimations in the general unfiltered case, but struggle to line up with the alternate scheme in terms of predictive performance, except for GB-related models. For RF, this can be explained by a worse h_k estimation while for MLP we assume that it is due to h_a: in the alternate approach, it is optimized one epoch at a time, interleaved with one step on h_k, whereas that of PD-based methods is fully optimized. Sequential and alternate approaches undergo stronger h_k misparameterization without filtering since H3 is not met, but the latter mitigates this w.r.t. the former, as was already observed in Section 4.1. Input filtering degrades predictive performance for the sequential methods as they cannot counterbalance a poor h_k.

Table 3. Results for the correlated Friedman problem.

		$\hat{d}(h, y; TS)$		$\hat{d}_k(h_k^{\theta_k}, f_k; TS)$	
	Method	Unfiltered	Filtered	Unfiltered	Filtered
MLP	$f_k \rightarrow h_a$	1.64 ± 0.23	1.51 ± 0.17	-	
	Sequential	2.07 ± 0.40	2.68 ± 1.38	1.35 ± 1.42	
	Alternate	1.95 ± 0.33	1.62 ± 0.24	0.49 ± 0.44	0.14 ± 0.19
	PD-based	2.24 ± 0.31	1.78 ± 0.30	0.17 ± 0.23	
	h_a only	2.77 ± 0.73		-	
GB	$f_k \rightarrow h_a$	2.58 ± 0.45	2.53 ± 0.44	-	
	Sequential	2.90 ± 0.39	3.91 ± 1.49	1.35 ± 1.42	
	Alternating	2.67 ± 0.38	2.62 ± 0.43	0.51 ± 0.53	0.22 ± 0.25
	PD-based	2.54 ± 0.35	2.47 ± 0.36	0.03 ± 0.02	
	h_a only	4.49 ± 0.66		-	
RF	$f_k \rightarrow h_a$	3.02 ± 0.45	2.93 ± 0.45	-	
	Sequential	3.78 ± 0.78	4.04 ± 1.30	1.35 ± 1.42	
	Alternating	3.06 ± 0.39	2.99 ± 0.38	0.14 ± 0.16	0.15 ± 0.18
	PD-based	3.24 ± 0.38	3.16 ± 0.37	0.27 ± 0.20	
	h_a only	6.70 ± 1.47		-	

4.3. Overlapping additive structure (H2 not satisfied)

Let $y = \beta x_0^2 + \sin(\gamma x_0) + \delta x_1 + \varepsilon$, with $\beta = 0.2, \gamma = 1.5, \delta = 1$, $\varepsilon \sim \mathcal{N}(0, 0.5^2)$ and $\mathbf{x}$ sampled as in the correlated linear problem. We define $f_k(\mathbf{x}_k) = \beta x_0^2$ and $f_a(\mathbf{x}) = \sin(\gamma x_0) + \delta x_1 + \varepsilon$. Hence, H2 does not hold. Even with $\hat{\beta} = \beta^*$, h_a still needs to compensate for $\sin(\gamma x_0)$. Filtering is thus expected to degrade MSE performance for all hybrid approaches as $h_a(x_1)$ will never compensate this gap, which is observed in Table 4. Results for RF are not shown for the sake of space, but are similar to GB.

Table 4. Results for the overlapping problem.

	$\hat{d}(h, y; TS)$		$\hat{d}(h, y; TS)$	
	Unfiltered	Filtered	Unfiltered	Filtered
Method	MLP		GB	
$f_k \rightarrow h_a$	0.35 ± 0.02	0.54 ± 0.04	0.51 ± 0.04	1.00 ± 0.12
Sequential	0.35 ± 0.01	0.59 ± 0.05	0.55 ± 0.07	1.07 ± 0.11
Alternate	0.35 ± 0.02	0.56 ± 0.05	0.54 ± 0.09	1.01 ± 0.11
PD-based	0.34 ± 0.02	0.56 ± 0.05	0.53 ± 0.05	0.99 ± 0.12
h_a only	0.37 ± 0.02		0.55 ± 0.07	

5. Conclusion

We study several hybrid methods on supervised regression problems modeled in an additive way, using neural networks models and tree-based approaches. We empirically show that trends observed for neural networks also apply for the non-parametric tree-based approaches, both in terms of predictive performance as in the estimation of the algebraic known function. We introduce claims related to the convergence of multiple hybrid approaches, under mild hypotheses, and verify their soundness on illustrative experiments. We present a new hybrid approach leveraging partial dependence and show its competitivity against sequential and alternate optimization schemes. We highlight its benefits in estimating the parametric prior and show that it alleviates both the risk of the ML term to dominate the known term and the need for assuming independent input features sets. As future work, we will apply our method to real problems, investigate further the theoretical properties of the PD-based approach and extend it to dynamical problems.

Software and Data

Our Python implementations of the hybrid methods are available at `https://github.com/yannclaes/kg-regression`.

Broader Impact Statement

Machine learning models can occasionally show undesired properties and behave badly when encountering states they had not been prepared/designed for. In this line of thought, our contributions aim at enhancing the combined effects of domain knowledge, through first-principles models, with machine learning models, which ought to make the latter more interpretable, less data-expensive and eventually more robust to perturbations and reliable for real-life applications, thereby reducing risks of failure.

Acknowledgements

This work was supported by Service Public de Wallonie Recherche under Grant No. 2010235 - ARIAC by DIGITALWALLONIA4.AI. Computational resources have been provided by the Consortium des Equipements de Calcul Intensif (CECI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11 and by the Walloon Region.

References

Ayed, I., de Bézenac, E., Pajot, A., Brajard, J., and Gallinari, P. Learning dynamical systems from partial observations. *Machine Learning and the Physical Sciences: Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Breiman, L. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Daw, A., Karpatne, A., Watkins, W., Read, J., and Kumar, V. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

De Bézenac, E., Pajot, A., and Gallinari, P. Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124009, 2019.

Donà, J., Déchelle, M., Levy, M., and Gallinari, P. Constrained physical-statistics models for dynamical system identification and prediction. In *ICLR 2022-The Tenth International Conference on Learning Representations*, 2022.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

Friedman, J. H. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

Friedman, J. H., Grosse, E., and Stuetzle, W. Multidimensional additive spline approximation. *SIAM Journal on Scientific and Statistical Computing*, 4(2):291–301, 1983.

Greene, W. H. *Econometric analysis*. Pearson Education India, 2003.

Mehta, V., Char, I., Neiswanger, W., Chung, Y., Nelson, A., Boyer, M., Kolemen, E., and Schneider, J. Neural dynamical systems: Balancing structure and flexibility in physical prediction. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 3735–3742. IEEE, 2021.

Takeishi, N. and Kalousis, A. Physics-integrated variational autoencoders for robust and interpretable generative modeling. *Advances in Neural Information Processing Systems*, 34:14809–14821, 2021.

Wehenkel, A., Behrmann, J., Hsu, H., Sapiro, G., Louppe, G., and Jacobsen, J.-H. Improving generalization with physical equations. *Machine Learning and the Physical Sciences: Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Yin, Y., Ayed, I., de Bézenac, E., Baskiotis, N., and Gallinari, P. Leads: Learning dynamical systems that generalize across environments. *Advances in Neural Information Processing Systems*, 34:7561–7573, 2021a.

Yin, Y., Le Guen, V., Dona, J., de Bézenac, E., Ayed, I., Thome, N., and Gallinari, P. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124012, 2021b.

Zhang, H., Nettleton, D., and Zhu, Z. Regression-enhanced random forests. *arXiv preprint arXiv:1904.10416*, 2019.