Boosting interaction tree stumps for modeling interactions

Michael Lau, Tamara Schikowski, Holger Schwender

Suggested Citation:
Lau, M., Schikowski, T., & Schwender, H. (2025). Boosting interaction tree stumps for modeling interactions. Computational Statistics & Data Analysis, 213, Article 108247.
https://doi.org/10.1016/j.csda.2025.108247

Computational Statistics and Data Analysis 213 (2026) 108247

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Boosting interaction tree stumps for modeling interactions

Michael Lau [a,b,c,*], Tamara Schikowski [b,d], Holger Schwender [a]

[a] Mathematical Institute, Faculty of Mathematics and Natural Sciences, Heinrich Heine University, Universitätsstr. 1, 40225, Düsseldorf, Germany
[b] IUF – Leibniz Research Institute for Environmental Medicine, Auf'm Hennekamp 50, 40225, Düsseldorf, Germany
[c] eBay Inc., 2025 Hamilton Avenue, San José, 95125, CA, USA
[d] School of Public Health, Bielefeld University, Universitätsstr. 25, 33615, Bielefeld, Germany

ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.14593699, https://doi.org/10.24432/C5NK6X, https://codalab.lisn.upsaclay.fr/competitions/7363

Keywords:
Linear interaction models
Penalized regression
Machine learning
Polygenic risk scores

ABSTRACT

Incorporating interaction effects is essential for accurately modeling complex underlying relationships in many applications. Often, not only strong predictive performance is desired, but also the interpretability of the resulting model. This need is evident in areas such as epidemiology, in which uncovering the interplay of biological mechanisms is critical for understanding complex diseases. Classical linear models, frequently used for constructing genetic risk scores, fail to capture interaction effects autonomously, while modern machine learning methods such as gradient boosting often produce black-box models that lack interpretability. Existing linear interaction models are largely limited to consider two-way interactions. To address these limitations, a novel statistical learning method, BITS (Boosting Interaction Tree Stumps), is introduced to construct linear models while autonomously detecting and incorporating interaction effects. BITS uses gradient boosting on interaction tree stumps, i.e., decision trees with a single split, where in BITS this split can possibly occur on an interaction term. A branch-and-bound approach is employed in BITS to discard weakly predictive terms. For high-dimensional data, a hybrid search strategy combining greedy and exhaustive approaches is proposed. Regularization techniques are integrated to prevent overfitting and the inclusion of spurious interaction effects. Simulation studies and real data applications demonstrate that BITS produces interpretable models with strong predictive performance. Moreover, in the simulation study, BITS primarily identifies truly influential terms.

1. Introduction

In genetic epidemiology and statistical genetics, statistical models are constructed and studied for investigating how genetics influence the manifestation of complex diseases. Often, genetic/polygenic risk scores (GRS) are constructed that summarize parts of the genome with respect to a considered disease outcome (Dudbridge, 2013; Lau et al., 2023). The input variables for constructing GRS are usually SNPs (single nucleotide polymorphisms) that are single base-pair alterations in the DNA. SNPs are coded as $\{0, 1, 2\}$, as they count the occurrences of the respective minor allele (i.e., the genetic variant that occurs less often in the reference population) in humans, which are diploid organisms that carry two complete chromosome sets.

For constructing GRS, generalized linear regression models or regularized variants are popular (see, e.g., Mavaddat et al., 2019; Privé et al., 2019), as they are easy to fit and interpret. The underlying assumption is that genetic loci influence the disease risk

* Corresponding author at: Mathematical Institute, Faculty of Mathematics and Natural Sciences, Heinrich Heine University, Universitätsstr. 1, 40225, Düsseldorf, Germany.
E-mail address: michael.lau@hhu.de (M. Lau).

https://doi.org/10.1016/j.csda.2025.108247
Received 10 May 2024; Received in revised form 14 June 2025; Accepted 7 July 2025

Available online 16 July 2025

independently on the considered phenotype scale. However, it is well known that genetic loci can interact with each other (Che and Motsinger-Reif, 2013). Moreover, genetic risk factors can also interact with environmental risk factors with regard to the considered phenotype (Ottman, 1996). Thus, procedures that construct more sophisticated models and can take interactions into account such as random forests produce more accurate models (Lau et al., 2022). However, these models are usually black boxes. Compared to linear models, they lose most of their interpretability.

Recently, statistical learning methods have been proposed that try to achieve both a high predictive ability and a high interpretability. These methods either lack scalability to high-dimensional problems, interpretability due to still fitting too complex models, or the ability to properly model the underlying relationships due to modeling assumptions that might not be true.

In this work, a novel statistical learning method is proposed that employs gradient boosting for fitting interpretable linear models with modeling flexibility by allowing the terms to be interactions between input variables. Potential interactions are identified in each boosting iteration by searching for the main or interaction effect that leads to the best model. This base learning method is called interaction tree stumps, as only a single term is fitted. The total model complexity is controlled by penalizing both long interactions and complex models consisting of many terms.

The structure of this paper is as follows. Before introducing the proposed methodology in detail in Section 3, related approaches are discussed in Section 2. In Section 4, BITS and related methods are evaluated in simulations considering multiple realistic data scenarios. In Section 5, two real-world data applications involving a genetic data set and a chemical data set are performed, in which BITS and related methods are also applied. Concluding discussions and remarks are provided in Section 6.

2. Related work

Decision trees are among the most frequently used statistical learning concepts and recursively partition the space of input variables (Breiman et al., 1984). At each inner node, the space is divided based on the value of, usually, a single input variable (multivariate splits have been, nonetheless, also proposed, see, e.g., Murthy et al. (1994)). However, single decision trees tend to be unstable, i.e., small training data set modifications might lead to large changes in the resulting model (Bertsimas and Digalakis, 2023). Hence, decision tree ensembles such as random forests (Breiman, 2001) or linear combinations of decision trees obtained using gradient boosting (Friedman, 2001) have been proposed that improve stability and often yield higher predictive performances. These decision tree ensembles are, however, no longer inherently interpretable, especially if deeper decision trees are fitted.

Many interpretability-focused statistical learning methods have been proposed in the past. Friedman (1991), e.g., proposed MARS (multivariate adaptive regression splines) that constructs model of the form

$$F(\boldsymbol{X}) = \beta_0 + \beta_1 m_1(\boldsymbol{X}) + \ldots + \beta_B m_B(\boldsymbol{X}),$$

where $\boldsymbol{X}$ is a p-dimensional random vector of input variables and m_i, $i \in \{1, \ldots, B\}$, are hinge functions $\max(0, \pm(X_j - c))$ or products of hinge functions with $c \in \mathbb{R}$ being a constant and X_j a selected input variable. This model is fitted in a greedy fashion finding the terms that minimize the error and pruning the resulting model to avoid overfitting. By also considering products of hinge functions, MARS can model interactions between input variables.

Logic regression (Ruczinski et al., 2003) is a statistical learning procedure specifically tailored to SNP data. Logic regression assumes that all input variables are binary, e.g., SNPs divided into dominant and recessive modes of inheritance. Logic regression builds models of the form

$$F(\boldsymbol{X}) = \beta_0 + \beta_1 L_1(\boldsymbol{X}) + \ldots + \beta_B L_B(\boldsymbol{X}),$$

where L_i, $i \in \{1, \ldots, B\}$, are logic expressions of $\boldsymbol{X}$ using Boolean operators $\wedge$ (and), $\vee$ (or), and ^{c} (negation). These logic expressions are identified through a global stochastic search employing simulated annealing (Kirkpatrick et al., 1983). Logic regression is able to model each possible prediction function for binary input data. However, the identified logic expressions might be overcomplicated and hard to interpret. Moreover, continuous covariables can only be taken additively into account, i.e., interactions with the binary input variables cannot be modeled.

Rule-based methods identify decision rules such as $X_{73} > 0 \wedge X_{42} \leq 1$. Usually, the decision rules are extracted from fitted decision tree ensembles. An established rule-based method is RuleFit (Friedman and Popescu, 2008) which constructs a boosted decision tree ensemble to generate the decision rules and employs the lasso (Tibshirani, 1996) to fit a linear model consisting of the most important identified rules and marginal terms. More recently, alternative rule-based methods that employ random forests for generating decision rules have been proposed (Meinshausen, 2010; Bénard et al., 2021; Boruah et al., 2023).

A similar class of methods fits regression trunk models, which are linear combinations of main effects and decision rules designed to capture interaction effects. These decision rules are derived from a regression trunk, i.e., a small regression tree. A specific method within this class is STIMA (simultaneous threshold interaction modeling algorithm, Dusseldorp et al., 2010). STIMA begins with fitting an ordinary linear regression model and iteratively adds interaction terms extracted from an expanding regression trunk. Subsequently, a cross-validation-based procedure prunes the resulting regression trunk to retain the most predictive interaction terms.

Lately, there has been a focus on methods that try to identify pairwise hierarchical interactions (Bien et al., 2013; Lim and Hastie, 2015; Yan and Bien, 2017; She et al., 2018; Wu et al., 2018; Hazimeh and Mazumder, 2020; Zhang et al., 2023). An interaction effect $\delta_{i,j}$ between input variables X_i and X_j is said to be strongly hierarchical if for the corresponding main effects β_i and β_j it holds that

$$\delta_{i,j} \neq 0 \;\Rightarrow\; \beta_i \neq 0 \wedge \beta_j \neq 0$$

(Bien et al., 2013). Similarly, weak hierarchy is defined by

$$\delta_{i,j} \neq 0 \;\Rightarrow\; \beta_i \neq 0 \vee \beta_j \neq 0.$$

Many approaches enforce hierarchy through regularization. An established approach is glinternet (Lim and Hastie, 2015) that employs a specific group lasso regularization that overlaps between main and interaction effects. Lim and Hastie (2015) also considered an interaction detection algorithm based on boosting decision trees of depth two. This alternative approach, however, was not able to compete with glinternet in their experiments.

Recently, methods have been proposed that do not rely on the hierarchy assumption, as it might be too restrictive (Yu et al., 2019; Wang et al., 2021). For identifying interactions in the (for pairwise interactions) quadratically growing search space, Yu et al. (2019) replace the hierarchy assumption with the reluctance principle, which simply states that main effects should be preferred over interaction effects if the induced predictive performance is equal. Yu et al. (2019) proposed the sprinter algorithm for identifying reluctant interactions, which first fits a full model of main effects and searches all pairwise interactions for the highest absolute correlation with the residual of the initial model. Subsequently, a regularized model is fitted using all main effects and the identified interactions.

Note that these approaches and the discussed approaches based on hierarchy can only take pairwise interactions into account. Higher order interactions are so far not considered due to computational challenges with the polynomially increasing number of possible interaction terms.

Another recent class of interaction detection methods is given by procedures that employ the concept of pure interactions (Lengerich et al., 2020; Sun et al., 2022). An interaction effect is said to be pure if the outcome variance cannot be explained by any subset of the interacting input variables (Lengerich et al., 2020). In this context, functional ANOVA (analysis of variance) decompositions are used that are constructed recursively and yield an additive model consisting of marginal and interaction effects (Hooker, 2004).

Interaction detection methods that also enable the performance of statistical inference on the identified terms have been recently proposed as well (Suzumura et al., 2017, 2021; Das et al., 2022). Suzumura et al. (2017, 2021) study an algorithm based on orthogonal matching pursuit that chooses in each iteration the term that yields the highest absolute inner product with the current residual. Since screening all possible interactions in each iteration could be computationally infeasible, the interaction terms are gathered in a tree structure and, through an upper bound of the considered score for descendants in the tree, the interaction term search can be locally terminated if this upper bound is below the highest computed score of the current iteration.

3. Boosting interaction tree stumps

In the following, a data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ consisting of n iid observations from the joint distribution of $(\boldsymbol{X}, Y)$ is considered, where $\boldsymbol{X}$ is a p-dimensional random vector. First, it is assumed that Y is a continuous outcome. In Section 3.8, it is discussed how the proposed methodology can be generalized to other types of outcome such as binary outcomes.

3.1. Interaction tree stumps

Gradient boosting (Friedman, 2001) and modern variants such as XGBoost (Chen and Guestrin, 2016) or LightGBM (Ke et al., 2017) produce state-of-the-art prediction models. However, due to constructing decision tree ensembles, the resulting model is no longer human-readable. For fitting interpretable models via boosting, decision tree stumps, i.e., decision trees of depth one that only contain one split, can be fitted in each boosting iteration instead of deep decision trees. Fig. 1**a** shows two ordinary decision tree stumps that split on two different input variables. Such two stumps are summed up in boosted models, yielding in this case the model

$$\hat{Y} = (0.25 + 0.09) + (0.55 - 0.25) \cdot X_1 + (-0.12 - 0.09) \cdot X_2.$$

However, stumps that split on single input variable can only capture marginal effects. In Fig. 1**b**, a decision tree of depth two is depicted that splits on the same two input variables as the stumps in Fig. 1**a**. Such deeper decision trees are able to also capture interaction effects. However, sums of these deeper trees are no longer easily interpretable. Moreover, the decision tree from Fig. 1**b** must also implicitly include the marginal effect of X_1, as this decision tree induces the model

$$\hat{Y} = 0.25 + (0.75 - 0.25) \cdot X_1 + (0.25 - 0.75) \cdot X_1 \cdot X_2.$$

This might be undesirable if the variables contained in the interaction do not exhibit any marginal effects. Furthermore, ordinary decision tree induction procedures employ greedy algorithms that might miss interaction terms if the marginal effects of the contained variables are negligible (see, e.g., Lau et al., 2024).

To overcome these drawbacks, *interaction tree stumps* are proposed that, as ordinary tree stumps, contain exactly one split, but this split could potentially be performed on an interaction term, directly revealing interaction effects. An exemplary interaction tree stump, that was fitted to the same data as the trees in Fig. 1**a–b**, is illustrated in Fig. 1**c**. This stump splits on the interaction term $X_1 \cdot X_2^c$, where X_2^c denotes the negation of X_2. This interaction term is binary, since the individual input variables X_1 and X_2 are binary. As for conventional decision trees, the leaves of this stump contain direct predictions of the outcome. This interaction tree stump captures the isolated interaction effect from the decision tree of depth two in Fig. 1**b**.

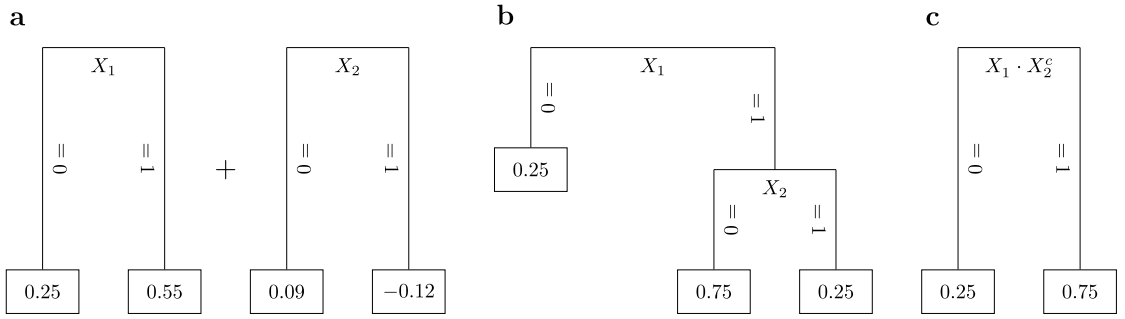

Fig. 1. Different types of decision trees fitted to the same data set. In **a**, two decision tree stumps splitting on the marginal effects of X_1 and X_2, respectively, are shown. In **b**, a decision tree of depth two is shown that splits on X_1 and X_2 and captures their interaction. In **c**, an interaction tree stump is presented that splits on the interaction of X_1 and X_2^c and captures the same interaction effect as the decision tree in **b**. The superscript ^{c} in X_2^c denotes the negation of X_2.

To enable the modeling of all possible interaction effects, i.e., the exploration of all possible decision tree branches, interactions of both the original predictors (e.g., in the above example, X_1 and X_2) and their negations (X_1^c and X_2^c) need to be considered (see Section 3.1.2).

3.1.1. Modeling linear effects

SNPs can exhibit different MOIs (modes of inheritance), i.e., different ways of influencing the outcome (Scherer et al., 2021). A SNP could exhibit an additive MOI such that the effect of the presence of minor alleles on both chromosomes is doubled compared to the presence of one minor allele. In this case, the SNP variable can be directly used in linear models. Moreover, a dominant MOI is also possible such that the effect is present if at least one minor allele is present and two minor alleles do not modify the effect over one minor allele. Here, the SNP variable can be coded as $\mathbb{1}(\text{SNP} > 0)$. A recessive MOI is another possibility, in which the effect is only present if minor alleles are present on both chromosomes, i.e., the SNP would be coded as $\mathbb{1}(\text{SNP} = 2)$.

Decision trees can directly model a dominant or recessive MOI by splitting on $\text{SNP} > 0$ or $\text{SNP} > 1$, respectively, which is equivalent to using the corresponding variable codings $\mathbb{1}(\text{SNP} > 0)$ or $\mathbb{1}(\text{SNP} = 2)$ in linear models. However, for modeling the additive MOI, a conventional decision tree would need to perform two consecutive splits on the same SNP. To also properly consider the additive MOI and to generalize interaction tree stumps to continuous input variables, simple linear regression models

$$\mathbb{E}[Y \mid T(\boldsymbol{X})] = \alpha_0 + \alpha_1 T(\boldsymbol{X})$$

are, therefore, considered, where the term $T(\boldsymbol{X})$ is a function that either consists of a single input variable corresponding to a main effect, i.e., $T(\boldsymbol{X}) = X_j$, or consists of multiple input variables corresponding to an interaction effect, i.e., $T(\boldsymbol{X}) = X_{j_1} \cdot \ldots \cdot X_{j_l}$. For binary input variables, decision tree stumps and simple linear regression models are thus equivalent.

For properly including SNPs as input variables in interaction tree stumps, the MOI of each SNP is identified prior to model fitting. Similarly to Scherer et al. (2021) and Petersen et al. (2012), this is done by evaluating for each SNP which MOI is most plausible. More precisely, for each SNP and each MOI, likelihood-ratio tests are performed testing the association of the respective SNP with the outcome using the considered MOI. The MOI that yields the minimum p-value is used for potentially transforming the SNP for the model fitting and prediction steps.

3.1.2. Variable negations

SNPs are coded based on the minor and major allele definition. The minor/major allele is the allele that is present in less/more than 50% of the reference population. Thus, the encoding of the input variable is not associated with the outcome or the corresponding effect direction, i.e., the minor allele is not necessarily also the risk-increasing allele, so that the encoding of the input variable only depends on its marginal distribution. By considering the complement/negated version $\text{SNP}^c := 2 - \text{SNP}$, the allele definitions are swapped. Hence, in the fitting of interaction tree stumps, negations of input variables are also considered, which removes the importance of including the correct SNP coding beforehand. Negations can be generalized to upper bounded variables X_j by considering $X_j^c = \max(X_j) - X_j$ so that a binary 0-1-variable X_j (such as a SNP with a dominant or recessive MOI) is negated by $X_j^c = 1 - X_j$.

Negations are non-trivial in interaction terms due to inducing a shift on participating variables. Consider, e.g., the model $m(\boldsymbol{X}) = \alpha_1 X_1 X_2^c = \alpha_1 X_1 - \alpha_1 X_1 X_2$ for binary input variables X_1 and X_2. As can be seen on the right-hand side, the negation X_2^c creates a marginal shift by $\alpha_1 X_1$. Thus, theoretically, the interaction $X_1 X_2$ is considered in this model, but only with an additional consideration of the marginal term X_1. Therefore, the need to include additional (marginal) terms could be avoided and a simpler model could be obtained by not only considering the input variables themselves, but also their negation.

3.1.3. Fitting interaction tree stumps

Interaction tree stump fitting is a discrete optimization problem, which has the goal to identify an ideal set of input variables. As in ordinary gradient boosting, the score/error of the base learner to be minimized is the empirical L_2 loss, i.e., the MSE (mean squared error)

$$S(m_T) := \frac{1}{n}\sum_{i=1}^{n}(m_T(\boldsymbol{x}_i) - y_i)^2,$$

where the stump m_T is based on the term T and given by

$$m_T(\boldsymbol{X}) = \hat{\alpha}_0 + \hat{\alpha}_1 T(\boldsymbol{X})$$

with $\hat{\alpha}_0$ and $\hat{\alpha}_1$ being the ordinary least squares estimates. Note that if the considered outcome type is not continuous, the optimization objective of the base learner of gradient boosting or interaction tree stumps, respectively, remains to be the L_2 loss, as the base learner receives the quantitative gradients of the boosting loss function (that incorporates the type of outcome) as target variable.

If negations are also considered, the exact number of possible interaction terms that need to be considered in the optimization process is given by

$$\sum_{i=1}^{k}\binom{2p}{i} - \sum_{i=2}^{k} p\binom{2(p-1)}{i-2}, \tag{1}$$

where k is the maximum considered interaction order. In the first sum, all subsets from p input variables and their p negations are gathered, and in the second sum, implausible interaction terms are removed such as $X_1 \cdot X_1^c$. This number scales in the magnitude of $\mathcal{O}\left((2p)^k\right)$.

For $p = 50$ and $k = 3$, e.g., the number of possible terms is given by 161,800. For this setting, it might be feasible to perform a complete search over all possible interaction terms to guarantee that the empirically optimal stump is identified.

When considering, e.g., $p = 1000$ and $k = 3$, the number of possible terms increases to 1,331,336,000. Here, it might no longer be computationally feasible to perform a complete search over all interaction terms. Therefore, in this scenario, a hybrid between a greedy search and a complete search is conducted for fitting interaction tree stumps that keeps evaluating terms based on the best evaluated term thus far until a maximum iteration number is reached (see Section 3.4.2). A pure greedy search would execute a maximum of $2kp$ iterations. The drawback of a greedy search, however, is that it might miss important interaction terms due to corresponding marginal effects being negligible or masking the interaction effect.

More details on fitting interaction tree stumps are provided in Section 3.4.

3.2. BITS model

In BITS (**B**oosting **I**nteraction **T**ree **S**tumps), interaction tree stumps are used as base learner trained in gradient boosting. More precisely, in each boosting iteration, the gradient of the current loss with respect to the next model is computed and the negative gradient is used as outcome in a new interaction tree stump. This new submodel, i.e., this stump, is added to the total model. Basically, gradient boosting performs gradient descent in the space of base learners and is based on the fact that the negative gradient points in the direction of steepest descent when minimizing a convex function such as the mean squared error for regression tasks or the negative binomial log-likelihood for binary outcomes.

The resulting model is given by

$$F(\boldsymbol{X}) = \rho_0 + \eta\rho_1 m_1(\boldsymbol{X}) + \ldots + \eta\rho_B m_B(\boldsymbol{X}),$$

where B is the number of boosting iterations, η is the learning rate considered in gradient descent, $\rho_j \in \mathbb{R}$, $j \in \{0, \ldots, B\}$, are the boosting coefficients, and m_j, $j \in \{1, \ldots, B\}$, are the trained base models. This BITS model is a sum of linear models, since the interaction tree stumps are linear models (linear in dependence of potential interaction terms). Hence, this model can be transformed into a linear model

$$F(\boldsymbol{X}) = \hat{\beta}_0 + \hat{\beta}_1 T_1(\boldsymbol{X}) + \ldots + \hat{\beta}_{B^*} T_{B^*}(\boldsymbol{X}), \tag{2}$$

where T_j, $j \in \{1, \ldots, B^*\}$, are the identified terms and $B^* \leq B$ is the number of uniquely identified terms.

The resulting BITS model in Eq. (2) is inherently interpretable, as the types of effects, i.e., main effect or interaction effect, and their magnitudes are directly revealed. Moreover, for interaction effects, it can also be directly derived in which way the involved predictors interact with each other. For example, the term $\hat{\beta}_1 T_1(\boldsymbol{X}) = -2.42 \cdot X_2^c X_7 X_{32}^c$ would have a decreasing effect on the response, only if $X_2 = 0$, $X_7 = 1$, and $X_{32} = 0$ for binary predictors X_2, X_7, and X_{32}. For continuous predictors X_3 and X_5, the term $\hat{\beta}_2 T_2(\boldsymbol{X}) = 1.89 \cdot X_3 X_5^c$ could be interpreted as an interaction effect that increases the effect of X_3 with decreasing values of X_5.

In Section 3.6, the complete algorithms for fitting interaction tree stumps and BITS models along with a theoretical time complexity analysis are presented.

3.3. Controlling model complexity

BITS aims at fitting multiple different stumps. Thus, simply restricting the stumps to contain exactly the same number of input variables might not lead to ideal models, since, e.g., the true model might be given by

$$\mathbb{E}[Y \mid \boldsymbol{X}] = \beta_0 + \beta_1 X_2 + \beta_2 X_1 X_4^c$$

so that one boosting iteration should ideally find X_2 while another boosting iteration should yield $X_1 X_4^c$. Therefore, it must be ensured that each boosting iteration can yield a model with a different model complexity. For small to moderately large data sets, simply setting the maximum number of variables without any further restrictions is not a practicable solution, since generally more complex models are preferred by statistical learning algorithms for minimizing the training loss.

Thus, in BITS, a penalizing approach similar to the one used in the lasso (Tibshirani, 1996) or the cost-complexity penalty in pruning decision trees (Breiman et al., 1984) is employed. For fitting interactions tree stumps, the score being minimized is thus adjusted by adding a penalty for terms that include more variables. Instead of the original, unadjusted score $S(m_T)$ for the model m_T, the adjusted score

$$S^*(m_T) = S(m_T) + \gamma ||m_T||_0 \tag{3}$$

is minimized, where $||m_T||_0$ is the number of variables contained in the term T of the corresponding model m_T and $\gamma \geq 0$ is the penalty parameter. Ideally, this penalty should lead to stopping adding unnecessary variables to the term, while uncovering exactly those responsible for the variation in the outcome. In Section 3.7, it is discussed how plausible candidate values for γ can be computed.

Penalizing interactions of higher order and preferring main effects, if the induced prediction performances are (nearly) equal, is also in line with the reluctance principle for main and interaction effects that was proposed by Yu et al. (2019) for detecting interactions (see Section 2).

The complexity of BITS models is not only regularized by considering the adjusted score S^* from Eq. (3), but also by pruning the resulting BITS model from Eq. (2) to the important terms. This pruning is achieved by employing a lasso regularization, i.e., fitting a model

$$\min_{(\beta_0, \boldsymbol{\beta})} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{B^*} \beta_j T_j(\boldsymbol{x}_i) \right)^2 + \lambda ||\boldsymbol{\beta}||_1 \right\},$$

where $\boldsymbol{\beta} = \begin{pmatrix} \beta_1 & \dots & \beta_{B^*} \end{pmatrix}^T$ is the vector of regression coefficients and λ is the regularization parameter or term inclusion penalty. In BITS, a relaxed lasso fit (Meinshausen, 2007; Hastie et al., 2020) is used that separates the problems of term selection and coefficient estimation by considering predictions

$$\hat{Y}_{\text{relaxed}} = \kappa \hat{Y}_{\text{lasso}} + (1 - \kappa) \hat{Y}_{\text{OLS}},$$

where $\hat{Y}_{\text{lasso}}$ is the ordinary lasso estimate, $\hat{Y}_{\text{OLS}}$ is the ordinary least squares estimate computed using only the variables selected by the lasso, and $\kappa \in [0, 1]$ is a hyperparameter controlling the balance between these two estimates. The relaxed fit is designed for obtaining an optimal term selection while also yielding a strong predictive performance.

3.4. Detailed interaction tree stump fitting procedure

For fitting interaction tree stumps, the objective to be optimized is the (penalized) mean squared error of simple linear regression models, since interaction tree stumps consider regression tasks. Minimizing the mean squared error of a simple linear regression model with respect to the variable is equivalent to maximizing the absolute empirical correlation, i.e.,

$$\underset{j \in \{1,\dots,p\}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{\alpha}_0 - \hat{\alpha}_1 x_{i,j} \right)^2 = \underset{j \in \{1,\dots,p\}}{\arg\max} \frac{\left| \sum_{i=1}^{n} x_{i,j} y_i - n \cdot \overline{x}_{\cdot,j} \cdot \overline{y} \right|}{\sqrt{\sum_{i=1}^{n} \left(x_{i,j} - \overline{x}_{\cdot,j} \right)^2} \sqrt{\sum_{i=1}^{n} \left(y_i - \overline{y} \right)^2}},$$

where $x_{\cdot,j}$ is the n-dimensional vector of observations for input variable X_j and $\overline{x}_{\cdot,j}$ is its arithmetic mean. Thus, estimates of the regression coefficients $\hat{\alpha}_0$ and $\hat{\alpha}_1$ do not need to be determined to evaluate the performance of a certain input variable. See Appendix A for more details on the derivation of the fitting objective in interaction tree stumps.

Since the outcome vector y is fixed in the fitting procedure of interaction tree stumps and it is assumed, without loss of generality, that y is centered, i.e., $\overline{y} = 0$, the objective is to maximize

$$\frac{\left| \frac{1}{n} \sum_{i=1}^{n} T(\boldsymbol{x}_i) y_i \right|}{\sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(T(\boldsymbol{x}_i) - \overline{T(\boldsymbol{x})} \right)^2}} - \gamma ||T||_0, \tag{4}$$

for fitting interaction tree stumps, where the optimal term T is identified that exhibits a maximum interaction order of k.

3.4.1. Complete branch-and-bound search

To reduce the computational burden of a complete search, a branch-and-bound strategy can be employed that discards interaction terms that cannot lead to a better objective value than the currently best value without fully evaluating these terms.

Similar to Suzumura et al. (2017), it is proposed to structure the search space as a tree (see Fig. 2). The root node consists of the empty model and the children of each node are created by adding one input variable (or its negation) to the term in the considered node. Fig. 2 illustrates the search space structure for $p = 4$ input variables and a maximum interaction order of $k = 4$. For simplicity,

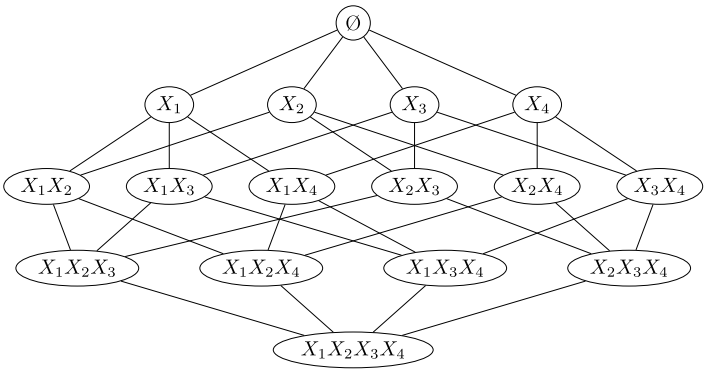

Fig. 2. Interaction tree stump search space for $p = 4$ input variables and a maximum interaction order of $k = 4$. Variable negations are not considered for simplicity.

variable negations are not included in this illustration that can be also interpreted as a Hasse diagram. The purpose of this structure is that at a considered node with term T an upper bound for the score of all possible descendant nodes is computed so that it can be concluded that no descendant node can yield a better term if this upper bound is smaller than the current maximum observed score. In this case, the search can, therefore, be locally terminated and descendants of the considered node can be discarded.

For applying a branch-and-bound search, Suzumura et al. (2017) consider the objective $|\sum_{i=1}^{n} T(\boldsymbol{x}_i)y_i|$ and assume $X_j \in [0,1]$ for every $j \in \{1,\ldots,p\}$. It is further assumed by Suzumura et al. (2017) that a term $T_1(\boldsymbol{x}) = x_{j_1} \cdot \ldots \cdot x_{j_l}$ was just evaluated and the search continues by evaluating terms that include T_1, i.e., terms $T_1 \cdot T_2$, where the term T_2 consists of input variables that are not included in T_1 so that further input variables are added to T_1. In this situation, it holds that

$$\left|\sum_{i=1}^{n} T_1(\boldsymbol{x}_i)T_2(\boldsymbol{x}_i)y_i\right| \leq \max\left\{\sum_{i:\,y_i>0} T_1(\boldsymbol{x}_i)y_i, \quad -\sum_{i:\,y_i<0} T_1(\boldsymbol{x}_i)y_i\right\}, \tag{5}$$

since $x_i \cdot x_j \leq x_i$ and $x_i \cdot x_j \leq x_j$ for $x_i, x_j \in [0,1]$. However, optimizing this score instead of the adjusted score (4), i.e., ignoring the standard deviation of the term $T_1 \cdot T_2$, might lead to identifying or neglecting terms not due to their correlation with Y, but due to their scale (see Appendix B). Hence, in the following, an upper bound for the adjusted score (4) is derived that incorporates the standard deviation of the term.

To keep the notation concise,

$$\langle T, y\rangle := \sum_{i=1}^{n} T(\boldsymbol{x}_i)y_i$$

denotes the inner product between an evaluated term T and y,

$$U(T,y) := \max\left\{\sum_{i:\,y_i>0} T(\boldsymbol{x}_i)y_i, \quad -\sum_{i:\,y_i<0} T(\boldsymbol{x}_i)y_i\right\}$$

denotes the upper bound from Eq. (5), and

$$\mathrm{sd}(T) := \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(T(\boldsymbol{x}_i) - \overline{T(\boldsymbol{x})}\right)^2}$$

denotes the empirical standard deviation of an evaluated term T.

Since the individual boosting iterations do not modify the input variables, but only the outcome, it is proposed for the training of the interaction tree stumps to compute the interaction features and corresponding standard deviations of all eligible interaction terms as an initialization step for BITS. These standard deviations can then be used in the inequality

$$\frac{\frac{1}{n}\left|\langle T_1 \cdot T_2, y\rangle\right|}{\mathrm{sd}(T_1 \cdot T_2)} - \gamma||T_1 \cdot T_2||_0 \leq \max_{\mathfrak{T}\in \mathrm{Comp}_{\leq}(T_1,k)}\left\{\frac{\frac{1}{n}U(T_1,y)}{\mathrm{sd}(\mathfrak{T})} - \gamma||\mathfrak{T}||_0\right\} \tag{6}$$

to obtain an upper bound for the optimization objective (4), where

$$\mathrm{Comp}_{\leq}(T_1,k) := \left\{\text{Terms } \mathfrak{T} \;\middle|\; ||\mathfrak{T}||_0 \leq k \text{ and } T_1 \subset \mathfrak{T}\right\}$$

is the set of all terms with a maximum interaction order of k that contain T_1. If this upper bound is less than the best score that has been observed so far, all descendants of T_1 can be discarded and do not have to be evaluated.

To efficiently calculate this maximum, the initialization step also computes for each term/node the minimum standard deviation of descendant nodes for each possible interaction order, since

$$\max_{\mathfrak{T}\in \mathrm{Comp}_{\leq}(T_1,k)} \left\{ \frac{\frac{1}{n}U(T_1,y)}{\mathrm{sd}(\mathfrak{T})} - \gamma ||\mathfrak{T}||_0 \right\} = \max_{\ell:\ ||T_1||_0<\ell\leq k} \left\{ \frac{\frac{1}{n}U(T_1,y)}{\min\limits_{\mathfrak{T}\in \mathrm{Comp}_{=}(T_1,\ell)} \mathrm{sd}(\mathfrak{T})} - \gamma \cdot \ell \right\}, \tag{7}$$

where

$$\mathrm{Comp}_{=}(T_1,\ell) := \left\{ \text{Terms } \mathfrak{T} \ \middle|\ ||\mathfrak{T}||_0 = \ell \text{ and } T_1 \subset \mathfrak{T} \right\}$$

is the set of all terms with an interaction order equal to ℓ that contain T_1.

Moreover, it also holds that

$$\frac{\frac{1}{n}\left|\langle T_1 \cdot T_2, y\rangle\right|}{\mathrm{sd}(T_1 \cdot T_2)} - \gamma ||T_1 \cdot T_2||_0 \leq \frac{\frac{1}{n}U(T_1,y)}{\mathrm{sd}(T_1 \cdot T_2)} - \gamma ||T_1 \cdot T_2||_0, \tag{8}$$

where the upper bound only depends on pre-computed quantities. Therefore, if the upper bound in (6) is not lower than the best score observed so far, the evaluation of the single term $T_1 \cdot T_2$ might still be skipped, if the best score is greater than the upper bound in (8), as, in this case, $T_1 \cdot T_2$ cannot yield a better score than the best term that was evaluated so far.

Due to discarding terms if a corresponding upper bound is less than the best score observed so far, the overall search procedure acts in a greedy fashion by first evaluating direct descendants of terms that exhibit the highest scores. This is done to achieve a relatively high score early on in the search procedure to discard as many terms as possible.

In Appendix C, it is empirically evaluated in a simulation study in which scenarios how many terms can be discarded by this search approach.

If a term $T_1 \cdot T_2$ cannot be discarded and has to be fully evaluated, only the absolute inner product $|\langle T_1 \cdot T_2, y\rangle|$ between the term $T_1 \cdot T_2$ and y has to be computed, since the interaction feature itself and the corresponding standard deviation were already computed in the initialization step. Moreover, since the features do not change over boosting iterations or for different values of the interaction length penalty γ, the once initialized search space can be reused for the complete fitting and validation procedure of BITS models.

3.4.2. Limiting computational resources

If the complete search space considering all possible interactions of up to a maximum interaction order k cannot be traversed and the proposed branch-and-bound technique cannot prune off enough terms, a hybrid between a greedy search and a complete search is conducted in BITS, in which the number of iterations in the search, and hence, the number of evaluated terms, is limited.

First, this search screens all p variables for the optimal score. Next, the search evaluates all descendants of the (marginal) variable with this optimal score that are given by all two-way interactions consisting of this and another of the $p-1$ remaining input variables. The search continues by selecting the best unevaluated term so far and evaluating all descendants of this terms that are, again, given by all terms with incremented interaction order that contain this term. The search continues until the maximum number of iterations is reached or the complete search space has been evaluated.

Using the tree search space structure from Fig. 2, the idea is to evaluate all children nodes of a considered node, where the considered node is chosen so that it exhibits the maximum observed score among all evaluated nodes whose children have not been evaluated yet. Therefore, this kind of search carries out a greedy search, but does not terminate unless a maximum number of iterations has been reached or the complete search space has been evaluated. Also in contrast to a conventional greedy search, the search is also allowed to go back one level in the search space. As an example, consider the situation in which the two-way interactions X_1X_2, X_1X_3, and X_1X_4 have been evaluated, since their parent node X_1 exhibits the largest score among all p input variables. If now X_2 yields a higher score than any of these two-way interactions, the next step in the search would not be to evaluate three-way interactions, but to evaluate all (non-evaluated) children of X_2, i.e., X_2X_3 and X_2X_4 (as X_1X_2 was already evaluated).

Since the complete search space cannot be traversed in this case, the term initialization step, that would compute all interaction features with corresponding empirical standard deviations, is not performed. Thus, the branch-and-bound approach cannot be employed in the hybrid search, as the empirical standard deviations of descendant terms that are mandatory for the computation of the upper bound (6) are unknown.

3.5. Dismissing spurious interactions

Due to locally searching for single terms, it might happen that if, e.g., two variables have strong main effects, their interaction—despite the interaction length penalty—might be included in the model before including the variables as main effects. Thus, whenever a term $\prod_j X_j$, i.e., a stump $\alpha_0 + \alpha_1 \prod_j X_j$, consisting of more than one variable is identified as the best term to include, its linear counterpart $\alpha_0 + \sum_j \alpha_j X_j$ is evaluated as well. If this counterpart consisting of multiple terms yields a better score than the single interaction term, the individual terms comprising this counterpart are added to the model to prevent including spurious interactions.

If an interaction consisting of more than two variables is identified, all of its possible partitionings are investigated. For example, when the term $X_1X_2X_3$, i.e., the stump $\alpha_0 + \alpha_1 X_1X_2X_3$, has been identified, the models

$$\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3,$$

$$\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 X_3,$$

$$\alpha_0 + \alpha_1 X_2 + \alpha_2 X_1 X_3,$$

$$\alpha_0 + \alpha_1 X_3 + \alpha_2 X_1 X_2$$

are evaluated. This adjustment procedure is carried out to avoid including interaction effects that are in fact not present, but lead to a lower error, as the effects of variables participating in the interaction term are not included in the model so far.

Vice versa, if only the interaction between several features influences the outcome, the interaction term should be preferred to the individual variables, and hence, be chosen, since the considered interaction term carries all important information in this case.

3.6. BITS algorithm

Algorithm 1: Fitting interaction tree stumps.

```
1  function fitInteractionTreeStump(Training data D, interaction length penalty γ, maximum interaction length k, maximum number of search iterations I,
     search space S):
2      Normalize all input variables X_j to [0, 1] in D; Center the outcome Y in D
3      T_best = ∅; ρ_best = 0
4      s = Empty stack; Push empty term T_best to s
5      i = 0
6      while i < I and |s| > 0 do
7          T_old = Get and remove top element from s
8          ρ_upper = Score upper bound of T_old
9          if ρ_upper ≤ ρ_best then
10             Continue to next while loop iteration
11         end
12         T_new = Empty list; P_new = Empty list
13         for Every modification T_new of T_old do
14             ρ_tmp upper = Temporary upper bound for the score of T_new using Eq. (8) and S
15             if ρ_tmp upper > ρ_best then
16                 ρ_new = Score of T_new using Eq. (4) and S
17                 Append T_new to T_new; Append ρ_new to P_new
18                 ρ_upper = Upper bound for the score of T_new using Eq. (7) and S
19                 Increment i
20                 if i = I then
21                     Break the for loop
22                 end
23             end
24             if ||T_new||_0 < k then
25                 Push T_new to s
26             end
27         end
28         ρ_new = max(P_new)
29         T_new = Term in T_new with score ρ_new
30         if ρ_new > ρ_best then
31             T_best = T_new; ρ_best = ρ_new
32         end
33     end
34     Revert normalizing all input variables X_j in D; Revert centering Y in D
35     m_best = Fit {Ŷ = α̂_0 + α̂_1 T_best(X)}
36     m_best = Screen additive combinations of participating terms of T_best (see Section 3.5)
37     return m_best
38 end
```

In the following, the algorithm of fitting BITS models is summarized and its computational complexity is derived.

In Algorithm 1, the algorithm for fitting interaction tree stumps is presented in pseudocode. The body of the inner for loop, which is executed for every modification of a considered term, i.e., for every addition of a variable (or its negation), is executed for at most I times, where I is the maximum number of search iterations. In this for loop, all the computationally intensive calculations of fitting interaction tree stumps are carried out.

For a complete search, computing the temporary upper bound in Line 14 of Algorithm 1 amounts to a complexity of $\mathcal{O}(1)$, since the search space with all considered interaction features and corresponding empirical standard deviations has already been initialized and the score of an ancestor term is already known. For a hybrid search, the interaction features and corresponding empirical standard deviations have to be computed first, which amounts to a complexity of $\mathcal{O}(nk)$. Computing the score of the currently investigated model in Line 16 amounts to a complexity of $\mathcal{O}(n)$, since only the inner product between the new term and the outcome has to be computed. The upper bound in Line 18 is computed in $\mathcal{O}(k)$, because the minimum standard deviations of descendant terms were already determined in the search space initialization, the score was already computed, and the maximum is taken over at most k term

Algorithm 2: Boosting interaction tree stumps.

function BITS(*Training data D, boosting iterations B, learning rate η, interaction length penalty γ, maximum interaction length k, maximum number of search iterations I*):
  $S = \emptyset$
  if *Complete search ($I \geq$ Size of the search space using Eq. (1))* **then**
    S = Initialize search space by computing all interaction features and corresponding empirical standard deviations
  end
  $F_0(\boldsymbol{x}) = \arg\min_{y' \in \mathbb{R}} \sum_{i=1}^{n} L(y', y_i)$
  for $b \in \{1, \ldots, B\}$ **do**
    $\nabla = \begin{bmatrix} \frac{\partial}{\partial F(\boldsymbol{x}_1)} L(F(\boldsymbol{x}_1), y_1)\big|_{F(\boldsymbol{x}_1)=F_{b-1}(\boldsymbol{x}_1)} \\ \vdots \\ \frac{\partial}{\partial F(\boldsymbol{x}_n)} L(F(\boldsymbol{x}_n), y_n)\big|_{F(\boldsymbol{x}_n)=F_{b-1}(\boldsymbol{x}_n)} \end{bmatrix}$
    $m_b = \texttt{fitInteractionTreeStump}(\{\boldsymbol{x}_i, -\nabla_i\}_{i=1}^{n}, \gamma, k, I, S)$
    $\rho_b = \arg\min_{\rho > 0} \sum_{i=1}^{n} L(F_{b-1}(\boldsymbol{x}_i) + \rho m_b(\boldsymbol{x}_i), y_i)$
    $F_b(\boldsymbol{x}) = F_{b-1}(\boldsymbol{x}) + \eta \cdot \rho_b m_b(\boldsymbol{x})$
  end
  F = Fit a (generalized) linear model $\left\{ g(\hat{Y}) = \hat{\beta}_0 + \sum_{j=1}^{B^{**}} \hat{\beta}_j T_j(\boldsymbol{X}) \right\}$ consisting of the B^{**} most influential identified terms T_j using the relaxed lasso and an appropriate link function g
  return F
end

lengths. Therefore, the complexity of fitting an interaction tree stump is given by $\mathcal{O}(I(n+k))$ for the complete search and $\mathcal{O}(Ink)$ for the hybrid search.

In Algorithm 2, the procedure of fitting BITS models is presented. The search space initialization in case of a complete search in Line 4 amounts to a complexity of $\mathcal{O}\left((2p)^k nk\right)$, since the size $|S|$ of the search space scales, as discussed in Section 3.1.3, in the magnitude of $\mathcal{O}\left((2p)^k\right)$ and the interaction features are computed in $\mathcal{O}(nk)$ steps. The line search in Line 10 has a closed-form solution for continuous outcomes. Therefore, the complexity of the line search is in this case given by $\mathcal{O}(n)$. Similarly, computing the gradient in Line 8 and computing the model update in Line 11 also amount to $\mathcal{O}(n)$.

As discussed by Efron et al. (2004), fitting lasso models by the LARS (least angle regression) algorithm amounts to a complexity of $\mathcal{O}(np^2 + p^3)$. In BITS, the lasso is applied to at most B terms, which translates the complexity of the lasso fit to $\mathcal{O}(B^2 n + B^3)$.

Hence, if a complete search is performed, and therefore, the number I of search iterations is chosen as the size of the search space, i.e., $I = |S| \in \mathcal{O}((2p)^k)$, the total complexity of BITS is given by

$$\mathcal{O}\Big((2p)^k nk + B(2p)^k(n+k) + B^2 n + B^3\Big) = \mathcal{O}\Big((2p)^k(nk + B(n+k)) + B^2 n + B^3\Big).$$

If a hybrid search between a greedy search and a complete search is performed, the computational complexity of BITS is given by

$$\mathcal{O}\left(BInk + B^2 n + B^3\right).$$

If a greedy search is performed, and therefore, the number I of search iterations is chosen as $I \propto kp$, the computational complexity of BITS, hence, becomes

$$\mathcal{O}\left(Bnpk^2 + B^2 n + B^3\right).$$

Thus, assuming that the number B of boosting iterations and the maximum interaction order k are fixed, the complexity of BITS employing a complete search is polynomial in p and the complexity of BITS employing a greedy search is linear in p.

Alternatively, if it is assumed that the number of predictive terms is not fixed, but increases linearly with the total number of input variables, i.e., if it is assumed that B scales linearly with p, the computational complexity of BITS is given by $\mathcal{O}\left((2p)^k(nk + p(n+k))\right)$, if a complete search is used, and $\mathcal{O}\left(np^2k^2 + p^3\right)$, if a greedy search is employed.

3.7. Computing a γ-path

Lasso models are usually fitted for an entire path of potential values for the term inclusion penalty λ. Similarly, a path of reasonable values for the interaction length penalty γ can be determined and evaluated so that γ and λ can be jointly optimized in an efficient way in BITS. First, the maximum value $\gamma_{\max}$ is computed by deriving the minimum value that would lead to an empty model, i.e., that fulfills

$$S(m_0) = S(m_1) + \gamma_{\max}||m_1||_0 = S(m_1) + \gamma_{\max},$$

where m_1 is the optimal model among all models consisting of a single variable and the optimal model m_0 not containing any variables is given by the average $m_0 = \frac{1}{n}\sum_{i=1}^{n} y_i$. The minimum value $\gamma_{\min}$ can, e.g., be chosen as $0.001 \cdot \gamma_{\max}$. The complete γ-path is then given by equidistant steps on the logarithmic scale between $\gamma_{\min}$ and $\gamma_{\max}$.

As described in Section 3.4, computed interaction features and corresponding standard deviations can be reused for quickly determining the complete γ-path.

For tuning the hyperparameters of BITS, the combined γ- and λ-paths can be determined using training data and evaluated using independent validation data. Besides using single training-validation data splits, k-fold cross-validation may also be employed to choose the (γ, λ) pair that minimizes the cross-validation error or to promote model parsimony and make use of the one-standard-error rule by selecting the simplest model (largest γ and its corresponding largest λ) whose induced error is within one standard error of the minimum error.

3.8. Generalized linear models

BITS can also be generalized from linear models to generalized linear models, and therefore, be applied to other than continuous outcomes. For this, the loss employed in gradient boosting has to be chosen appropriately and the model pruning with the relaxed lasso has to be performed using an appropriate link function.

If a continuous outcome is considered, the linear model

$$\mathbb{E}[Y \mid \boldsymbol{X}] = F(\boldsymbol{X}) = \rho_0 + \eta \sum_{j=1}^{B} \rho_j m_j(\boldsymbol{X})$$

should be fitted. The squared error $L(F(\boldsymbol{x}), y) = (F(\boldsymbol{x}) - y)^2/2$ is usually chosen as the corresponding loss function, which leads to the residuals

$$\frac{\partial}{\partial F(\boldsymbol{x})} L(F(\boldsymbol{x}), y) = F(\boldsymbol{x}) - y$$

for computing the gradient.

Another example is a binary outcome, in which case the logistic model

$$\text{logit}(\mathbb{E}[Y \mid \boldsymbol{X}]) = \text{logit}(\mathbb{P}(Y = 1 \mid \boldsymbol{X})) := \log\left(\frac{\mathbb{P}(Y = 1 \mid \boldsymbol{X})}{1 - \mathbb{P}(Y = 1 \mid \boldsymbol{X})}\right) = F(\boldsymbol{X}) = \rho_0 + \eta \sum_{j=1}^{B} \rho_j m_j(\boldsymbol{X})$$

should be fitted (Friedman, 2001; Friedman et al., 2000). In this case, the negative binomial log-likelihood is used as loss function which leads to

$$\begin{aligned} L(F(\boldsymbol{x}), y) &= -\log\left(\left[\text{logit}^{-1}(F(\boldsymbol{x}))\right]^{y} \cdot \left[1 - \text{logit}^{-1}(F(\boldsymbol{x}))\right]^{1-y}\right) \\ &= \log(1 + \exp(F(\boldsymbol{x}))) - y \cdot F(\boldsymbol{x}) \end{aligned}$$

for the true value $y \in \{0, 1\}$ of the outcome. The gradient can then be calculated using

$$\frac{\partial}{\partial F(\boldsymbol{x})} L(F(\boldsymbol{x}), y) = \text{logit}^{-1}(F(\boldsymbol{x})) - y.$$

Other types of outcome with distributions that belong to the exponential family can be also fitted in BITS, as they can be fitted with gradient boosting procedures in general (Bühlmann and Hothorn, 2007).

4. Simulation study

In the following, BITS is applied to different simulation scenarios to empirically investigate its applicability. Moreover, comparable approaches that also fit interpretable models involving interactions between input variables are applied to compare BITS to these methods. More precisely, glinternet (Lim and Hastie, 2015), sprinter (Yu et al., 2019), MARS (Friedman, 1991; Milborrow, 2021), RuleFit (Friedman and Popescu, 2008; Fokkema, 2020), and logic regression (Ruczinski et al., 2003) are considered in this comparison. In addition, random forests (Breiman, 2001; Wright and Ziegler, 2017) is also applied as a machine learning method that induces state-of-the-art prediction models, which are black-box models, and thus, do not provide an (easy) interpretation. All analyses are carried out using R version 4.0.3 (R Core Team, 2020).

4.1. Simulation setup

Since BITS is motivated by applications to genetic association studies in which a selection of genetic variants from, e.g., different genes or pathways is analyzed, the input variables are simulated as SNPs that can take the values 0, 1, or 2, counting the numbers of minor alleles.

The application to data from SNPs that are pruned based on linkage disequilibrium, i.e., correlation structures between SNPs, is of particular interest, as this is common practice in genetic epidemiology (see, e.g., Calus and Vandenplas, 2018; Hüls and Czamara, 2020). Therefore, 50 SNPs are simulated as independent variables. A minor allele frequency of 0.25 is used for all SNPs. In the following, a dominant mode of inheritance is denoted by $\text{SNP}^D := \mathbb{1}(\text{SNP} > 0)$, and a recessive mode of inheritance is denoted by $\text{SNP}^R := \mathbb{1}(\text{SNP} = 2)$.

For the application of glinternet, sprinter, and logic regression, the binary input variables SNP^D and SNP^R are used so that these methods can also identify the correct modes of inheritance. For MARS, RuleFit, and random forests, the raw variables $\text{SNP} \in \{0, 1, 2\}$ are used as input variables, as these tree-based methods are directly able to capture dominant and recessive modes of inheritance using discrete splits (see Section 3.1.1).

In addition, a continuous environmental variable E is simulated by drawing from a rectified normal distribution with a mean of 20 and a standard deviation of 10. This is achieved by first sampling E' from the Gaussian distribution $\mathcal{N}(20, 10^2)$ and subsequently truncating negative values to zero, i.e., $E = \max(0, E')$. This ensures that the values of E are non-negative, which is usually the case for exposures to environmental risk factors such as air pollution.

Four different simulation scenarios are considered. First, a linear model

$$\mathbb{E}[Y \mid \mathbf{SNP}] = \text{SNP}_1 + \text{SNP}_2 + \text{SNP}_3^D + \text{SNP}_4^D + \text{SNP}_5^R + \text{SNP}_6^R \tag{9}$$

without any interaction effects is considered to investigate whether BITS identifies spurious interactions and fits overly complex models in this scenario or if BITS correctly mainly identifies main effects.

Next, two simulation scenarios involving interactions are considered that were also used by Lau et al. (2024) to evaluate, besides other methods, RuleFit, logic regression, and random forests. In the first of these scenarios, a model

$$\mathbb{E}[Y \mid \mathbf{SNP}] = \left[\sqrt{\log(1.5)} \cdot \text{SNP}_1^D + \sqrt{\log(2)} \cdot \text{SNP}_2^D \left(1 - \text{SNP}_3^D\right)\right]^2, \tag{10}$$

is employed in which a two-way interaction and a three-way interaction have an effect on the outcome (on the considered scale) due to the quadratic function. In the other scenario, a model

$$\mathbb{E}[Y \mid \mathbf{SNP}, E] = \log(2) \cdot \text{SNP}_1^D + \log(4) \cdot \text{SNP}_2^D \left(1 - \text{SNP}_3^D\right) \frac{E}{20}, \tag{11}$$

is considered in which a two-way SNP interaction is involved in a gene–environment interaction with E.

Lastly, a more complex model

$$\begin{aligned}\mathbb{E}[Y \mid \mathbf{SNP}, E] = {} & \text{SNP}_1^D + 1.5 \cdot \text{SNP}_2^D \frac{E}{1.349 \cdot 10} - \text{SNP}_3^R - \text{SNP}_4 \\ & + 2 \cdot \text{SNP}_5^D \text{SNP}_6 \frac{E}{1.349 \cdot 10} - 2 \cdot \text{SNP}_7^D \left(1 - \text{SNP}_8^D\right) \text{SNP}_9\end{aligned} \tag{12}$$

is investigated that consists of main effects, two-way interactions, three-way interactions, different modes of inheritance, and different effect directions. Here, E is standardized to have an (approximate) interquartile range of one, as the effect sizes of gene–environment interactions are often specified per interquartile range of the considered environmental risk factor (see, e.g., Hüls et al., 2017).

For every simulation scenario, nine different simulation settings are considered. More precisely, the total training sample size n is varied between 500, 1000, and 2000, corresponding to small, medium, and large data sets. For deriving precise prediction performances, independent test data sets with a sample size of 10,000 are utilized. Furthermore, the signal-to-noise ratio (SNR) is varied between 0.5, 1, and 2 by adding random noise from the Gaussian distribution $\mathcal{N}(0, \mathbb{V}\text{ar}(\text{Signal})/\text{SNR})$ to the prediction functions in the four simulation scenarios.

For every simulation setting, 100 independent replications are carried out, i.e., 100 independent data sets are randomly generated. For tuning the hyperparameters of the statistical learning methods considered in the comparison, the training data set of each replication is randomly divided into 75% training data and 25% validation data. For evaluating the performance of the different statistical learning methods, the models are trained with these procedures again on the complete training data set (including the validation data) using the optimized hyperparameter settings.

4.2. Hyperparameter optimization

The hyperparameters of all considered methods are optimized to yield the lowest MSE on the validation data sets.

A complete term search is conducted by BITS in all simulation settings. The number of boosting iterations is fixed to 50, the learning rate to 0.1, and the maximum interaction order to 3. Moreover, a grid of 50 different values for the interaction length penalty γ and 100 different values for the lasso regularization parameter λ is investigated.

Fig. 3 shows exemplarily for the complex simulation scenario with the medium setting of $n = 1000$ and $\text{SNR} = 1$ the validation data error for different values of the two hyperparameters. As depicted in this figure, a value of about $\exp(-2)$ seems to be optimal for γ, which is in the middle of the considered γ-path. For λ, smaller values seem to be preferred such that BITS already seems to identify mostly predictive terms that do not have to be pruned.

For all methods that make use of lasso, i.e., BITS, glinternet, sprinter, and RuleFit, the penalty parameter λ is tuned using the corresponding validation data set in each replication to enable fair comparisons between the methods. For all evaluated methods, the set of tuned hyperparameters with corresponding descriptions and considered values can be found in Appendix D.

4.3. Predictive performance

The predictive performance of all fitted models F is assessed on independent test data using the coefficient of determination R^2. This metric can be determined on a data set $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, n$, by

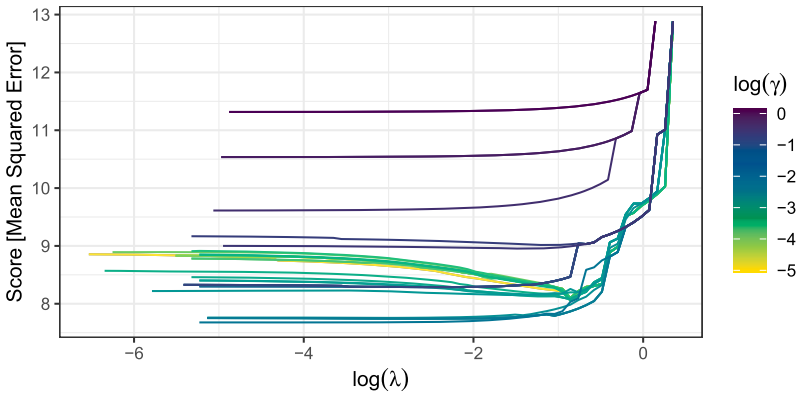

Fig. 3. Validation data error in the complex simulation scenario considering the setting with $n = 1000$ and SNR = 1 for BITS and different values of the interaction length penalty γ and the term inclusion penalty λ. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

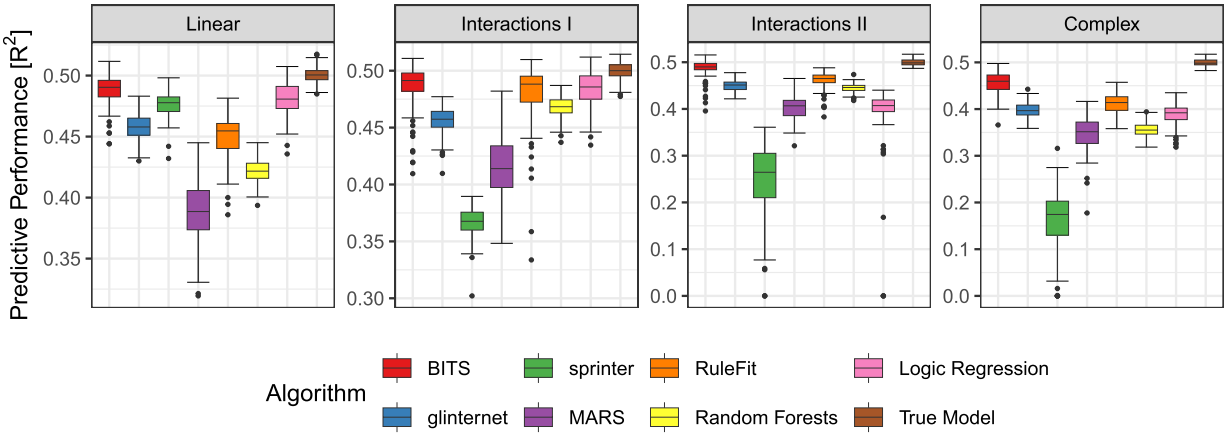

Fig. 4. Predictive performance of BITS and the comparable procedures in the simulation study considering the setting with $n = 1000$ and SNR = 1.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - F(\boldsymbol{x}_i))^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} = 1 - \frac{\text{MSE}}{\widehat{\mathbb{V}\text{ar}}(y)},$$

where $\widehat{\mathbb{V}\text{ar}}(y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2$ is the empirical variance of $y_1, \ldots, y_n$. While R^2 is equivalent to the MSE, as it is a rescaled (and inverted) modification of the MSE, it has the advantage that it provides a standardized interpretation. $R^2 = 0$ implies that the predictions of the fitted model are on average no better than simply taking the mean over the values of the outcome as constant prediction, and $R^2 = 1$ corresponds to perfect predictions with no error.

In Fig. 4, the predictive performances of BITS and the compared methods are displayed for the four considered simulation scenarios and the medium setting with $n = 1000$ and SNR = 1. The predictive performances in all other considered simulation settings can be found in Appendix E.

For the linear scenario, sprinter is expected to yield the highest predictive power, as it follows the reluctance principle for interactions. Therefore, its most important goal is to prefer main effects if all else is equal. BITS, however, performs slightly better than sprinter and logic regression in this linear scenario, suggesting that BITS is also applicable to scenarios, in which no interaction effects between the input variables are present. The other considered approaches lead to inferior predictive performances, indicating that they fit overly complex models.

In the simple interaction scenarios, BITS outperforms the other considered methods and yields performances close to the true underlying model for a high SNR.

In the complex simulation scenario, BITS also induces relatively high R^2 values. However, even for a high SNR and a large sample size, the true model cannot be fully attained due to the complexity of this scenario. In the complex scenario, glinternet, RuleFit, and logic regression yield similar, but slightly lower performances than BITS.

4.4. Term identification performance

In the following, it is investigated how often correct or wrong terms are identified depending on the simulation scenario and the number of identified terms. For this purpose, the lasso penalty λ is varied in BITS and glinternet to obtain optimal models for various

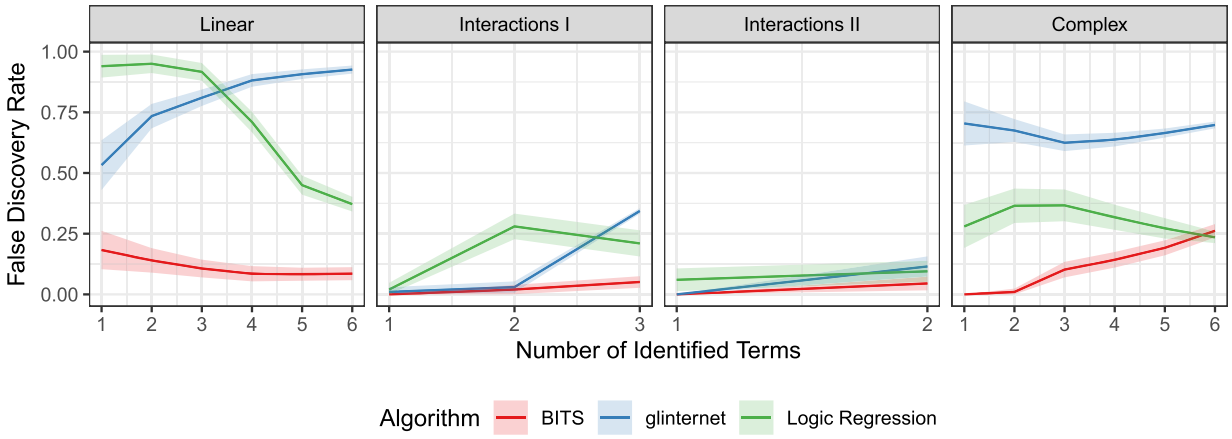

Fig. 5. False discovery rate of identified terms with asymptotic 95% confidence intervals in the simulation study considering the setting with $n = 1000$ and SNR = 1 for BITS, glinternet, and logic regression.

numbers of identified terms. In logic regression, the optimal number of variables that minimizes the validation data error is used for every number of trees. For determining whether an identified term is influential, the mode of inheritance and variable negations are not considered. It is thus only investigated if the correct input variable are identified in a term. Moreover, the environmental covariable E is not considered, as glinternet is not able to capture three-way interactions and logic regression is not able to capture interactions with continuous input variables. Thus, it is sufficient to identify the SNPs participating in a gene–environment interaction.

Fig. 5 shows the false discovery rate (FDR), i.e., the fraction of identified terms that are not included in the true underlying model, for the medium setting with $n = 1000$ and SNR = 1 in the four simulation scenarios. For all scenarios and numbers of identified terms, BITS leads to the lowest FDR, except for the complex scenario and six identified terms, in which logic regression yields a slightly better FDR.

For the linear scenario, the FDRs of BITS and logic regression decrease with the number of identified terms. This might be unintuitive, since the number of remaining (strongly) influential terms decreases with an increasing number of identified terms, as in general important terms are detected first. However, this phenomenon is presumably caused by BITS and logic regression trying to include multiple input variables into one interaction term if only one term should be identified. For BITS, this behavior does not seem to be severe and is presumably alleviated by the adjustment for spurious interactions that was designed for these situations (see Section 3.5). For the other considered simulation scenarios, the FDR of BITS increases, as expected, with the number of identified terms.

The FDRs for all other considered simulation settings can be found in Appendix F and show analogous results to the simulation setting discussed above. In addition, the identification of terms is also analyzed in more detail in Appendix F, including the comparison of false discovery and false negative rates of the optimal models, the number of identified terms, and how often main effects and interaction effects are correctly detected. As can be seen in this analysis, BITS tends to identify slightly more terms than logic regression, but also leads to detecting more terms from the true underlying models. Moreover, BITS leads to low error rates in identifying interaction effects, but detects slightly more false main effects than logic regression.

5. Real data application

BITS and the procedures considered for comparison are also applied to two real data sets, on the one hand, to a medium-dimensional genetic data set from an epidemiological cohort study, and on the other hand, to a high-dimensional toxicological data set that was used in the NeurIPS 2003 feature selection challenge.

5.1. SALIA study

The SALIA study (**S**tudy on the Influence of **A**ir Pollution on **L**ung, **I**nflammation and **A**ging, Schikowski et al., 2005) is a German cohort study, in which 4874 women were initially recruited in the period between 1985 and 1994. In follow-up examinations, epidemiological outcomes such as the status of different diseases and risk factors such as the genetic makeup were recorded. More details on the SALIA study can, e.g., be found in Krämer et al. (2010).

In the evaluation of BITS and the other methods, the outcome of interest is the presence of at least one rheumatic disease and is, therefore, binary. The subset of the data from the SALIA study that is considered in this evaluation consists of data from 517 women with available information on rheumatic diseases as well as genetic information. 123 study participants had (at least) one rheumatic disease so that 394 study participants did not have a rheumatic disease. 77 SNPs from the HLA-DRB1 gene are available, which showed significant associations with the development of rheumatoid arthritis in prior analyses (see, e.g., Clarke and Vyse, 2009). This data set from the SALIA study was also analyzed by Lau et al. (2024).

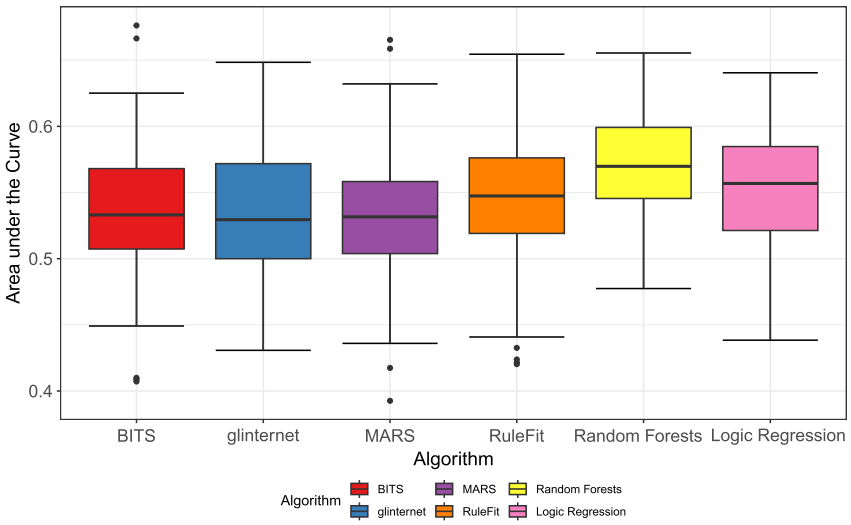

Fig. 6. Predictive performance of BITS and the comparable procedures in the application to data from the SALIA study.

To consider, similar to the simulation study discussed in Section 4, 100 replications, the data set from the SALIA study is 100 times randomly partitioned into training, validation, and test data sets. In each of these replications, 30% of the data set is used as test data, while the remaining 70% of observations are split into 75% training data and 25% validation data. As in the simulation study, the final models are then fitted on the combined training and validation data set using the respective optimal hyperparameter setting of the statistical learning method in the respective iteration. The hyperparameters of all considered procedures are optimized for the highest AUC (area under the receiver operating characteristic curve). In this application, sprinter is not considered, since sprinter is not implemented for non-continuous outcomes.

In Fig. 6, the distribution of the values of the AUC are shown for BITS and the other methods in their application to the SALIA data. BITS, glinternet, MARS, RuleFit, and logic regression all induce similarly high AUCs. The corresponding median AUCs are between 0.53–0.55. Random forests yields a slightly higher median AUC of approximately 0.57. However, random forests is also the only considered method that does not yield (directly) interpretable prediction models. All induced median AUCs are below 0.60, which indicates that the signal in the considered data set is relatively weak.

To interpret the resulting models, the median of each optimal hyperparameter value over the 100 replications was used to fit one model on the complete SALIA data for BITS, glinternet, and logic regression, respectively. BITS yields a model consisting of 18 terms, from which 4 are main effects, 10 are two-way interactions, and 4 are three-way interactions. For the full model displayed in Appendix H, it can be directly seen how predictions are composed. For example, the term $-4.17 \cdot \text{rs41288045}_R$ indicates that the SNP rs41288045 seems to decrease the risk of rheumatic diseases if its minor allele is present on both chromosomes (due to the recessive mode of inheritance as denoted by the subscript $_R$). Moreover, the term $7.90 \cdot \text{rs1060176}_R \cdot \text{rs151025335}_A$ indicates that the simultaneous presence of the minor allele of SNP rs1060176 on both chromosomes seems to increase the risk for rheumatic diseases if the SNP rs151025335 is present on at least one chromosome and this risk is further increased if rs151025335 is present on both chromosomes (due to the additive mode of inheritance as denoted by the subscript $_A$).

glinternet identifies a model consisting of 11 terms. All of these terms are two-way interactions. Logic regression detects a model consisting of 4 terms from which 3 terms are two-way interactions and one term is a three-way interaction. Hence, two-way interactions dominate all three models.

5.2. *Dorothea data set*

The Dorothea data set is a high-dimensional drug discovery data set that was used in the NeurIPS 2003 feature selection challenge (Guyon et al., 2004). It was also used by the authors of glinternet for comparing glinternet to similar procedures (Lim and Hastie, 2015). The data set contains 100,000 binary input variables that represent structural molecular features of chemical compounds. Half of the input variables are artificial noise features that do not influence the binary outcome, which is the binding ability to thrombin (an important enzyme in blood clotting). 1950 observations are available with a fixed split into 800 training observations, 350 validation observations, and 800 test observations. Since the Dorothea data set is part of a data science competition, the labels of the test data set are not public and the test data performance can only be assessed by submitting the test data predictions for the outcome to the competition website.

BITS and glinternet are the only considered procedures that are able to directly handle the complete data set of 100,000 input variables. However, it is not possible for BITS to consider the complete search space of more than 10^{15} terms for a maximum interaction order of $k = 3$. Thus, the hybrid search approach between a greedy and a complete search (see Section 3.4.2) was employed in BITS with a maximum number of search iterations of $4k(2p) = 2{,}400{,}000$, which is four times as many search iterations as would be carried out by a pure greedy search. Moreover, in contrast to the previous experiments with at most $p = 77$ input variables, 100 (instead of

Table 1
Predictive performance of BITS and the comparable procedures in the application to the Dorothea data set.

Method	BITS	glinternet	MARS	RuleFit	Random Forests	Logic Regression
AUC	**0.895**	**0.890**	0.729	0.813	**0.901**	0.799

50) boosting iterations are conducted in the application of BITS to the Dorothea data with $p = 100{,}000$ input variables to enable BITS to detect more (true positive) effects in this high-dimensional data set.

Analogously to Lim and Hastie (2015), glinternet was restricted to consider interactions with the highest 1,000 main effects in the application to the Dorothea data.

MARS, RuleFit, random forests, and logic regression cannot be applied to the whole data set, as they either do not allow that many input variables or the software for applying the respective method crashes when trying to fit models with this vast number of input variables. Thus, for applying MARS, RuleFit, random forests, and logic regression to the Dorothea data, a variable selection using the lasso was performed, resulting in 155 input variables. This selection was used as input variables in the application of these procedures.

In Table 1, the AUCs of BITS and the other methods determined on the Dorothea test data are depicted. BITS, glinternet, and random forests yield the highest AUCs. The superior predictive performance induced by BITS and glinternet compared to the other interpretability-focused methods is presumably partially caused by these methods being the only methods that could directly utilize all input variables.

To analyze the effect of considering all input variables in this application, BITS and glinternet were also evaluated using only the 155 variables selected by the lasso. With this preselection, BITS and glinternet yield similar AUCs of 0.867 and 0.875, respectively, which are slightly lower than the AUCs without preselection. Thus, this preselection leads to a slight performance loss.

BITS yields a model containing 48 terms, from which 38 terms correspond to main effects and 10 terms correspond to effects of two-way interactions. The full model can be found in Appendix I. In contrast, the application of glinternet results in a model consisting of 80 terms, which all correspond to two-way interactions. The application of random forests leads to a model consisting of 2000 trees with a median number of terminal nodes of 119. Hence, BITS yields the best interpretable model among the three best performing methods, as the BITS model contains less terms with fewer interaction terms compared to the glinternet model and the random forests model is not (directly) interpretable.

6. Conclusion and discussion

BITS is a novel statistical learning method for fitting linear models that can autonomously incorporate interaction effects. By penalizing long interactions and dismissing spurious interactions that are only detected because of main effects of variables included in the considered interaction term, BITS adheres to the reluctance principle for interactions, preferring main effects over interaction effects, and thus, simpler models, if the induced predictive performances are (almost) equal.

In a simulation study, it could be seen that BITS is able to construct simple models that do not contain many interaction terms if the underlying model does so as well. However, in more complex scenarios, BITS is able to fit correspondingly complex models involving higher order interaction terms. This was confirmed by the high predictive performance and low false discovery rate yielded by BITS in the simulation study. In the real data applications, BITS also showed comparatively strong predictive performances.

Most methods that fit linear interaction models, e.g., glinternet and sprinter, are restricted to detecting pairwise interactions, i.e., interactions of two variables. In contrast, BITS can detect interactions of up to a specified maximum interaction order, which was set to three in all applications. The maximum interaction order k could in principle be set to higher values. However, the increased computational burden for increasing k has to be taken into account.

The computational complexity of BITS was derived and is even for a complete search considering all possible interaction terms polynomial in the number p of input variables. To further reduce the computational burden, a branch-and-bound technique was proposed that discards interaction terms based on an appropriate upper bound for their scores and the best score so far. For problems with a huge number of input variables, such as the analysis of the Dorothea data, a hybrid between a greedy and a complete search can be employed that scales linearly in p. In Appendix G, actual model fitting and prediction times of BITS and procedures with which BITS was compared are presented and compared. As expected, BITS with a complete search is the most computationally intensive method. However, the hybrid search significantly reduces computation, making BITS faster to fit and evaluate than glinternet, RuleFit, random forests, and logic regression.

As mentioned in Section 3.7, k-fold cross-validation can also be used for hyperparameter tuning in BITS. This cross-validation procedure is expected to require about k times the computational effort of a typical single data split. This is because the training data slightly changes across the k folds, preventing the reuse of initialized search spaces for interaction terms in the complete search (see Section 3.4.1) between the folds. However, parallelizing the k folds could reduce this computational burden if the necessary hardware is available.

Many recent interaction detection methods assume weak or strong hierarchy in interaction terms, which is a reasonable assumption in many applications. BITS does not directly make hierarchy assumptions. Only weak hierarchy is implicitly assumed, if a pure greedy search is employed, as in this case detectable marginal effects are necessary for the identification of interactions. If one wishes to explicitly enforce weak or strong hierarchy, the interaction stump search can be modified so that effects participating in identified terms are carried on in the search. For example, if the search identifies the term X_1X_2 based on X_1, i.e., because X_1 induced a low

model error which lead to investigating terms that include X_1, then both terms X_1X_2 and X_1 are added in the boosting iteration for achieving weak hierarchy. For strong hierarchy, all terms X_1X_2, X_1, and X_2 would be added to the model. In the final step of BITS in which the terms are pruned using the relaxed lasso, marginal terms will be pruned off again if only the interaction terms seem to have an effect. The hierarchy notion can be generalized to interactions of higher order, leading to including all participating effects for strong hierarchy (i.e., considering all subsets of variables of the interaction term). Weak hierarchy for higher interaction orders can be defined recursively by requiring that at least one participating term with one less variable fulfills weak hierarchy. For example, $X_1X_2X_3$ fulfills weak hierarchy if X_3 and X_1X_3 have marginal effects.

Global optimization over integers (here, indices of input variables) seems to be a natural task solved by mixed-integer optimization. However, since, in the context of BITS, the search is performed over products of the form $T(\boldsymbol{X}) = X_{j_1} \cdot \ldots \cdot X_{j_l}$, the optimization objective also consists of products of potentially continuous variables. In mixed-integer optimization, transforming a product between two continuous variables into a linear objective involves piecewise linear approximations of every considered product with a fixed number of bins (Asghari et al., 2022). Therefore, mixed-integer optimization does not seem to be an appropriate solution to the considered optimization task.

Statistical learning methods that aim to fit a highly predictive decision tree, such as optimal decision tree procedures (Bertsimas and Dunn, 2017), also yield well-interpretable models. However, effect sizes or variable importances of individual terms in decision trees cannot be directly accessed. Moreover, statistically testing whether the detected terms actually influence the outcome using independent test data is also no longer straightforward. In contrast to single decision tree procedures, BITS and other methods that fit linear interaction models are able to provide summary statistics such as estimates of effect sizes and potentially confidence intervals and p-values (that can be obtained using independent test data). Such summary statistics are usually published as a result of genome-wide association studies (Uffelmann et al., 2021). However, such publications, typically, contain only information on marginal effects of genetic variants. Therefore, in contrast to conventional linear methods that are limited to marginal effects, BITS can be used to also detect interaction effects and determine corresponding summary statistics.

Statistically testing the importance of detected terms in BITS would require fitting the model to training data and performing the statistical tests on independent test data. This is because BITS must first generate the hypotheses to be tested by identifying influential input variables and interactions between input variables. To avoid using one random data split and trying to leverage the whole data set for both model fitting and testing, one idea for future research might be to employ multisplitting (Meinshausen et al., 2009; Dai et al., 2024), where the complete data set is randomly divided into training and test data sets multiple times and the test results are aggregated using quantiles of the observed p-values.

Code availability

The proposed methodology is implemented in the R software package `BITS` (Lau, 2024), which is publicly available on GitHub. The code for obtaining the empirical results presented in this manuscript is publicly available on Zenodo (Lau, 2025).

CRediT authorship contribution statement

M.L. and H.S. developed BITS and designed the simulation study. M.L. and T.S. conceived the real data application to data from the SALIA cohort study. The simulation study and the real data evaluations were conducted by M.L. M.L. was the major contributor in writing the manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate

The SALIA cohort study was conducted in accordance to the declaration of Helsinki and has been approved by the Ethics Committees of the Ruhr University Bochum and the Heinrich Heine University Düsseldorf. The study coordinators received written informed consent from all participants.

Funding

This work has been supported by the Research Training Group “Biostatistical Methods for High-Dimensional Data in Toxicology” (RTG 2624, Project R3) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation – Project Number 427806116). The SALIA follow-up study 2007–2010 was funded by the German Statutory Accident Insurance (DGUV) Grant No: 617.0-FP266 and the European Community’s Seventh Framework Programme (FP 7/2007-2011) under the grant agreement number 211250.

Acknowledgements

The authors would like to thank Trevor Hastie for insightful discussions. Computational infrastructure and support were provided by the Centre for Information and Media Technology at the Heinrich Heine University Düsseldorf.

Appendix A. Equivalence between minimizing the mean squared error and maximizing the absolute empirical correlation

If a simple linear regression model $\hat{Y} = \hat{\alpha}_0 + \hat{\alpha}_1 X$ is fitted, the coefficient of determination R^2 and the sample correlation coefficient r^2 are equal (see, e.g., Section 6.4, Rencher and Schaalje, 2007). Therefore, assuming y is centered so that $\sum_{i=1}^n y_i = 0$, it holds that

$$\begin{aligned} R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{\alpha}_0 - \hat{\alpha}_1 x_i)^2}{\sum_{i=1}^n y_i^2} &= \frac{(\sum_{i=1}^n x_i y_i)^2}{(\sum_{i=1}^n (x_i - \overline{x})^2)(\sum_{i=1}^n y_i^2)} = r^2 \\ \iff \quad -\text{MSE} = (r^2 - 1)\frac{1}{n}\sum_{i=1}^n y_i^2 &= r^2 \frac{1}{n}\sum_{i=1}^n y_i^2 - \frac{1}{n}\sum_{i=1}^n y_i^2 \\ &= \frac{(\frac{1}{n}\sum_{i=1}^n x_i y_i)^2}{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})^2} + C \end{aligned}$$

for the mean squared error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^n \left(y_i - \hat{\alpha}_0 - \hat{\alpha}_1 x_i\right)^2$$

and a constant C that does not depend on x. Thus, minimizing the mean squared error (with respect to the chosen variable x) is equivalent to maximizing

$$\frac{\left(\frac{1}{n}\sum_{i=1}^n x_i y_i\right)^2}{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})^2},$$

which in turn is equivalent to maximizing its square root

$$\frac{\left|\frac{1}{n}\sum_{i=1}^n x_i y_i\right|}{\sqrt{\frac{1}{n}\sum_{i=1}^n (x_i - \overline{x})^2}}.$$

In the context of BITS, this means that

$$\underset{T:\ ||T||_0 \le k}{\arg\min} \frac{1}{n}\sum_{i=1}^n \left(y_i - m_T(\boldsymbol{x}_i)\right)^2 = \underset{T:\ ||T||_0 \le k}{\arg\max} \frac{\left|\frac{1}{n}\sum_{i=1}^n T(\boldsymbol{x}_i) y_i\right|}{\sqrt{\frac{1}{n}\sum_{i=1}^n \left(T(\boldsymbol{x}_i) - \overline{T(\boldsymbol{x})}\right)^2}},$$

where m_T is the model based on the term T.

Appendix B. Covariance versus correlation

To see that standardization of the terms by their standard deviation is important for identifying the most predictive terms, consider the exemplary data set

$$D = \begin{array}{c} \\ 1 \\ \vdots \\ n \\ n+1 \\ \vdots \\ 2n \\ 2n+1 \end{array} \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \left[\begin{array}{cc|c} 0 & 0 & -1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & -1 \\ \delta & 1 & 1 \\ \vdots & \vdots & \vdots \\ \delta & 1 & 1 \\ 1 & 1 & 0 \end{array}\right] \end{array} \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ y \end{array}$$

for $\delta \in (0, 1)$. The value vectors x_1 and x_2 of both input variables X_1 and X_2 (that can be also interpreted as terms here) are scaled to $[0, 1]$ and the value vector y of the outcome Y is centered. Considering the observations $1, \ldots, 2n$, both x_1 and x_2 are perfectly correlated with y. Taking also observation $2n+1$ into account, the absolute inner products are given by $|\langle x_1, y\rangle| = \delta \cdot n$ and $|\langle x_2, y\rangle| = n$. Hence, if δ is small, the absolute inner product score would clearly prefer x_2 over x_1. Moreover, if δ is small enough, any other term with a non-zero score would be preferred over x_1. The reason for this is the mainly small range of values of x_1 and the outlier $x_{2n+1,1} = 1$.

By considering the correlation objective (4), terms are corrected for their standard deviation. In the above example, the asymptotic empirical standard deviation of x_1 is given by

$$\text{sd}(x_1) = \sqrt{\frac{\delta^2 n + 1}{2n+1} - \left(\frac{\delta n + 1}{2n+1}\right)^2} \xrightarrow{n \to \infty} \frac{\delta}{2}.$$

Therefore, the correlation objective (4) without considering the interaction length penalty would be (asymptotically) equal to 1 for both x_1 and x_2.

Appendix C. Efficiency of the branch-and-bound search

In Section 3.4.1, a branch-and-bound approach for reducing the complete search space of interaction terms was presented. In the following, it is analyzed how many terms can be discarded by this approach.

Fig. C.7 depicts the fraction of discarded interaction terms depending on the boosting iteration, the interaction length penalty γ, and the SNR, where here the complex simulation scenario presented in Eq. (12) and a medium sample size of $n = 1000$ are exemplarily considered. Not surprisingly, the fraction of discarded terms is the largest for high values of γ, since the upper bound for the branch-and-bound approach bounds descendant terms that exhibit a higher number of variables per interaction. For increasing SNRs and decreasing numbers of boosting iterations (which are equivalent in the sense that with increasing boosting iterations the signal in the gradient decreases due to identifying the most important terms early on), the fraction of discarded terms increases. An explanation for this is that a high SNR also induces a higher discrepancy between the score of influential terms and non-influential terms, which leads to more non-influential terms with upper bounds below the best total score so far.

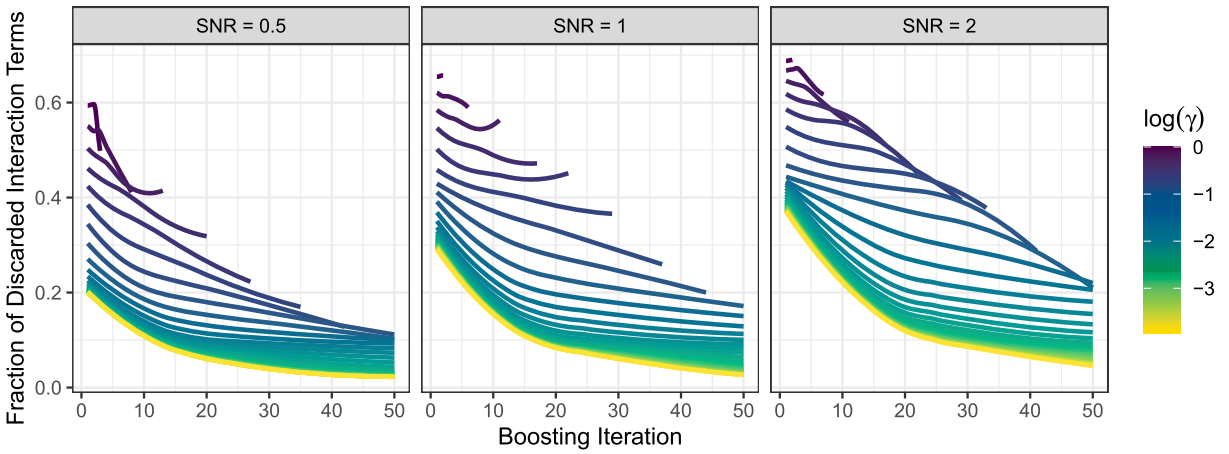

Fig. C.7. Fraction of terms that are discarded by BITS and its branch-and-bound approach in the complex simulation scenario with $n = 1000$ for different SNRs, boosting iterations, and interaction length penalties γ.

For weak signals, the fraction of discarded terms does not seem to be substantially high. However, this fraction is positive in all cases so that at least some terms can be discarded.

Appendix D. Optimized hyperparameters

Table D.2
Optimized hyperparameters with corresponding descriptions. The names of the hyperparameters are the names of the corresponding arguments in the respective `R` packages. For the `min.node.size` parameter of random forests in the `ranger` package, the default setting of 5 observations for regression tasks and 10 observations for probability estimation tasks has been evaluated as well. The design of this table is adapted from Lau et al. (2024).

Method	Software package	Hyperparameter	Description	Considered realizations
BITS	`BITS` (Lau, 2024)	`gamma`	Interaction length penalty	50 log-equidistant values in the γ-path $[\gamma_{\min}, \gamma_{\max}]$
		`lambda`	Term inclusion penalty	100 log-equidistant values in the λ-path $[\lambda_{\min}, \lambda_{\max}]$
		`learning.rate`	Learning rate	0.1
		`boosting.iter`	Number of boosting iterations	50
		`max.vars`	Maximum interaction order	3
glinternet	`glinternet` (Lim and Hastie, 2021)	`lambda`	Term inclusion penalty	50 log-equidistant values in the λ-path $[\lambda_{\min}, \lambda_{\max}]$ (default)
sprinter	`sprintr` (Yu, 2019)	`lambda`	Term inclusion penalty	100 log-equidistant values in the λ-path $[\lambda_{\min}, \lambda_{\max}]$ (default)
MARS	`earth` (Milborrow, 2021)	`degree`	Maximum interaction order	3

(*continued on next page*)

Table D.2 (*continued*)

Method	Software package	Hyperparameter	Description	Considered realizations
RuleFit	`pre` (Fokkema, 2020)	`sampfrac`	Subsample fraction for drawing samples without replacement for each tree	$\{0.5, 0.75, 1.0\}$
		`minbucket`	Minimum number of observations per leaf	$\lfloor \{0.01, 0.05, 0.1\} \cdot N \rfloor$
		`learnrate`	Learning rate	0.1 (same as in BITS)
		`ntrees`	Number of boosting iterations	50 (same as in BITS)
		`penalty.par.val`	Term inclusion penalty	100 log-equidistant values in the λ-path $[\lambda_{\min}, \lambda_{\max}]$ (default)
Random Forests	`ranger` (Wright and Ziegler, 2017)	`mtry`	Number of randomly drawn input variables at each split	$\left\lfloor \{0.5, 1, 2\} \cdot \lfloor \sqrt{p} \rfloor \right\rfloor$
		`min.node.size`	Minimum number of observations per leaf	$\lfloor \{0.01, 0.05, 0.1\} \cdot N \rfloor \cup \left\{ \begin{array}{ll} 5, & \text{regression} \\ 10, & \text{probability} \end{array} \right\}$
		`num.trees`	Number of trees grown	2000
Logic Regression	`LogicReg` (Kooperberg and Ruczinski, 2023)	(`nleaves`, `ntrees`)	Maximum number of (total) leaves and maximum number of trees	$\{(i, j) \in \{1, \ldots, 12\} \times \{1, \ldots, 6\} \mid i \geq j\}$
		`anneal.control`	Simulated annealing cooling schedule	Experimental

Appendix E. Predictive performance

See Fig. E.8.

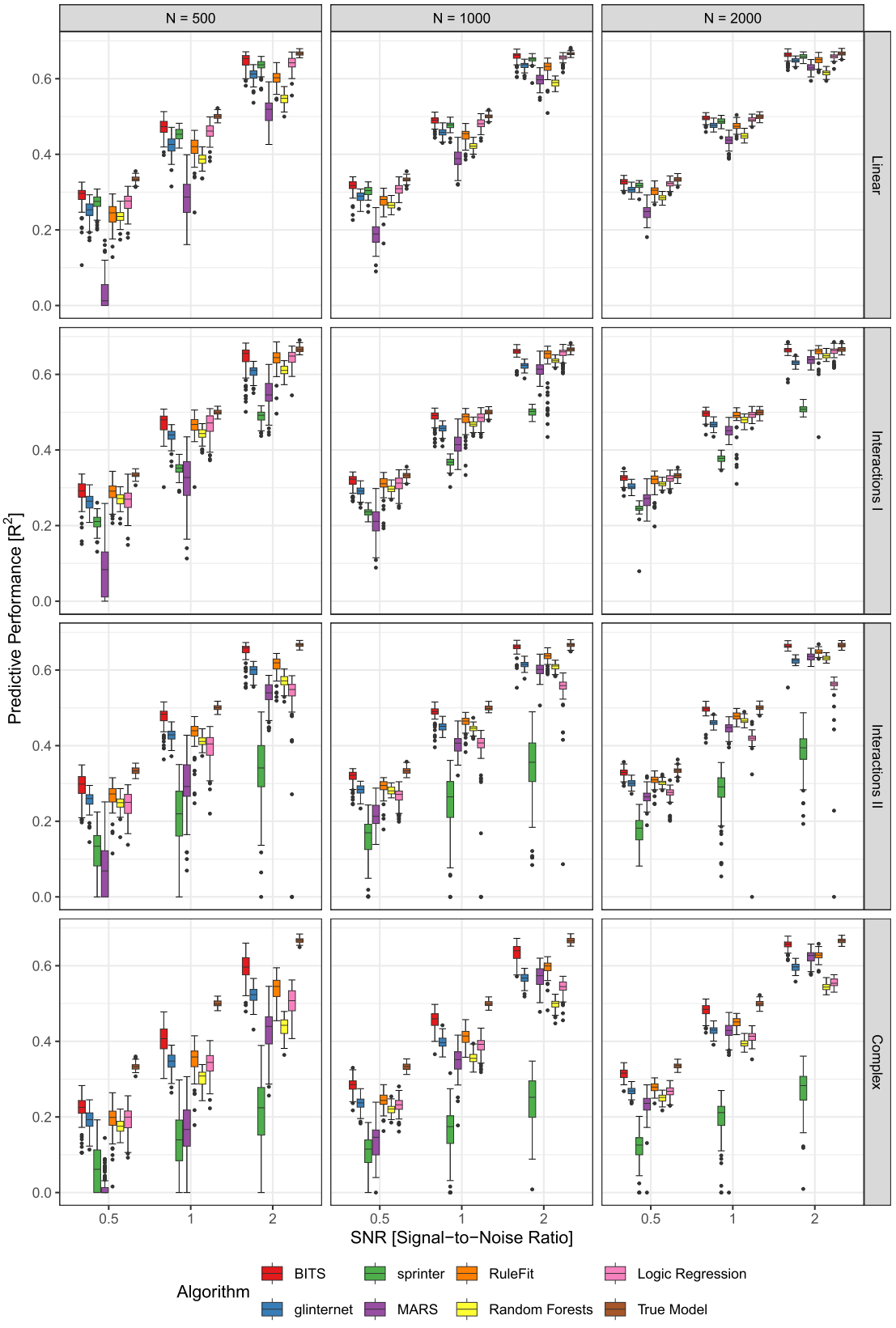

Fig. E.8. Predictive performance of BITS and the comparable procedures in the simulation study considering different simulation scenarios, sample sizes, and SNRs.

Appendix F. Term identification performance

In Fig. F.9, the FDRs for all considered simulation scenarios are presented corresponding to the results depicted in Fig. 5 and discussed in Section 4.4.

F.1. Number of identified terms

Fig. F.10 shows how many terms are identified in the optimal models generated by BITS, glinternet, and logic regression. This figure reveals that glinternet includes the highest number of terms in all scenarios. BITS and logic regression identify fewer terms, thus more closely reflecting the true underlying model sizes (Linear: six terms, Interactions I: three terms, Interactions II: two terms, Complex: six terms; see Section 4.1). BITS tends to identify slightly more terms than logic regression, which may be also caused by the hyperparameter optimization for logic regression being restricted to include at most six terms (see Table D.2). This limit, derived from the hard-coded maximum of five trees in the `LogicReg` software package (Kooperberg and Ruczinski, 2023), was raised from five to six to ensure logic regression could accurately reflect the true underlying models in the simulation study.

F.2. False discovery and false negative rates of optimal models

In Fig. F.11, the corresponding false discovery rates (FDRs) and false negative rates (FNRs) of terms identified by the optimal models are presented. As in Section 4.4, the FDR is calculated as the ratio between the number of falsely identified terms and the number of all identified terms, and the FNR is determined by the ratio between the number of undetected terms in the true model and the number of all terms in the true model. For the linear simulation scenario and the two simple interaction scenarios, BITS induces the lowest FDRs and the lowest FNRs in most settings. By contrast, glinternet leads to relatively high FDRs in all scenarios, since its corresponding optimal models contain more identified terms (as discussed in Appendix F.1). In the complex simulation scenario, BITS induces higher FDRs than logic regression when the SNR is low or the data set is small. However, BITS also induces lower FNRs in these settings, as it misses true terms less frequently.

F.3. Identification of main effects and interaction effects

Considering the complex simulation scenario that comprises both main and interaction effects, it is also investigated how often main effects and interaction effects are correctly detected. For this purpose, FDRs and FNRs are computed by exclusively considering either main effects or interaction effects in the resulting models and the true underlying model. Fig. F.12 depicts the results of this analysis. While glinternet almost exclusively identifies interaction effects, BITS and logic regression detect both main and interaction effects in a balanced manner. BITS yields in all settings the lowest FDRs and FNRs for detecting interaction terms. In the detection of main effects, BITS leads to higher FDRs than logic regression, which explains the slightly higher total FDR of BITS in the complex scenario that could be seen in Appendix F.2. Falsely detecting main effects, however, does not degrade the model interpretability as severely as falsely detecting interaction effects, due to the relative simplicity of the former. Nonetheless, BITS also induces the lowest FNRs for detecting main effects.

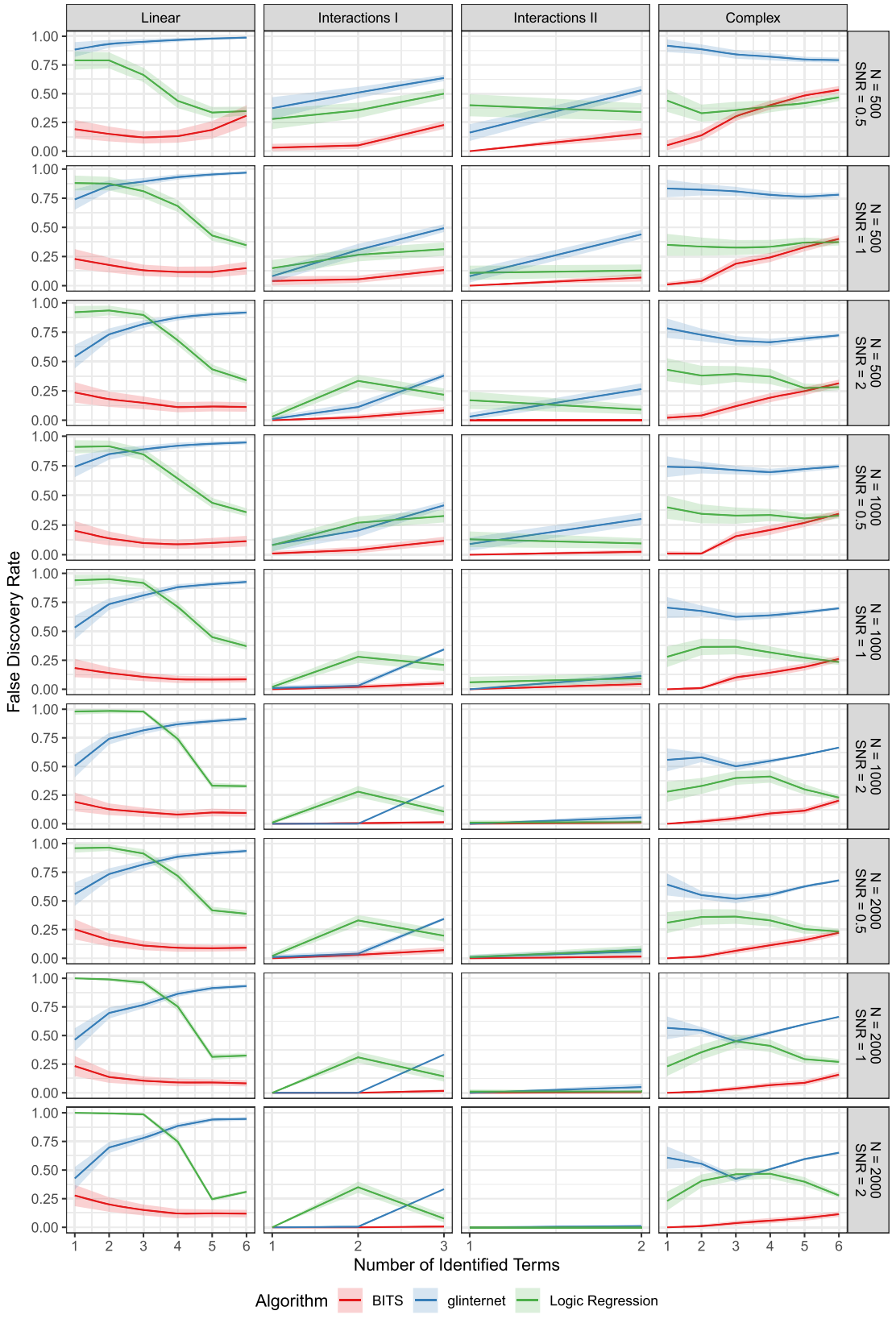

Fig. F.9. False discovery rate of identified terms with asymptotic 95% confidence intervals in the simulation study for BITS, glinternet, and logic regression.

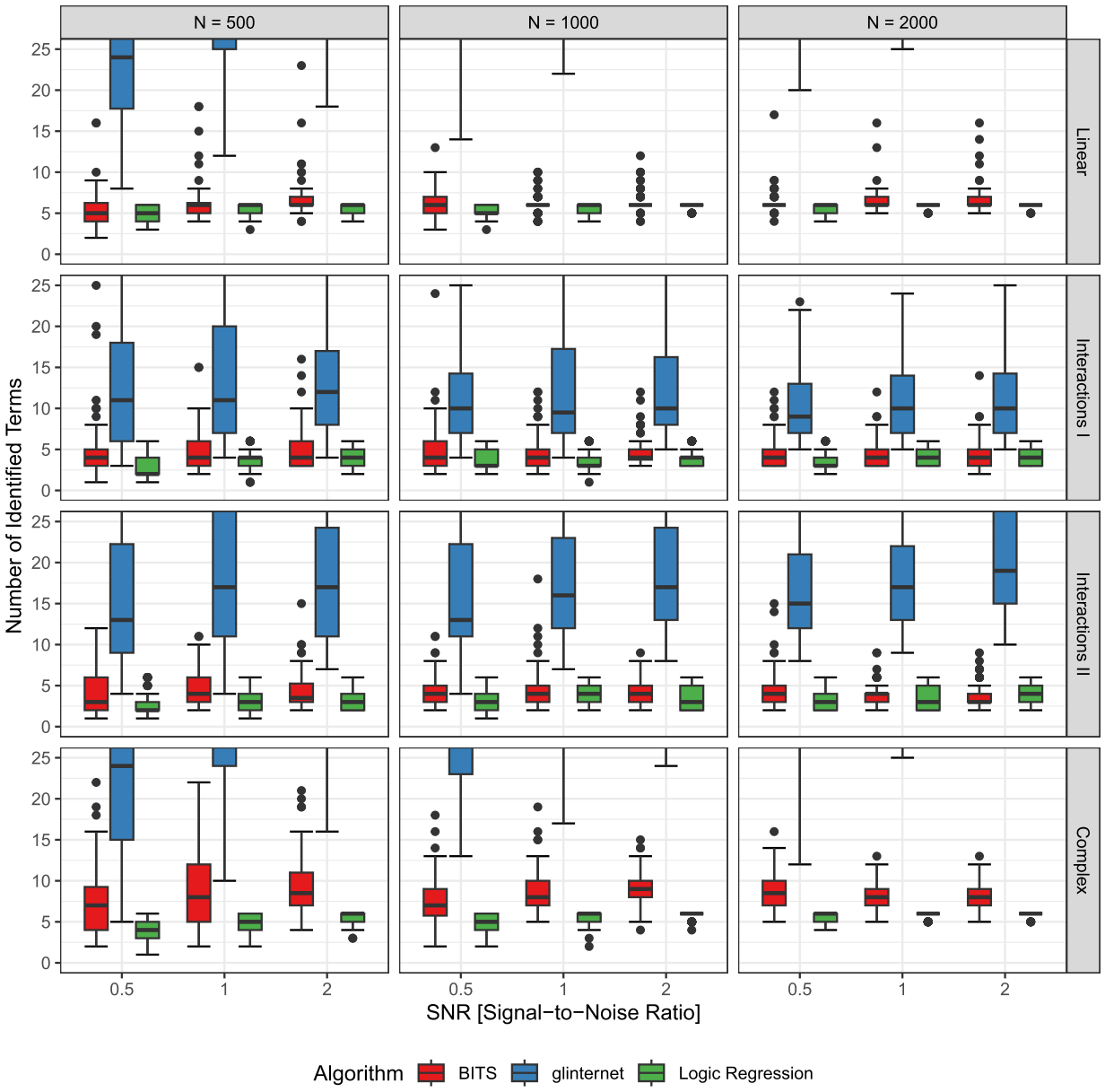

Fig. F.10. Number of identified terms in the optimal (hyperparameter-tuned) models in the simulation study for BITS, glinternet, and logic regression. The y-axis is truncated at 25 terms to facilitate comparison, particularly in the range of identified terms close to the true underlying model size.

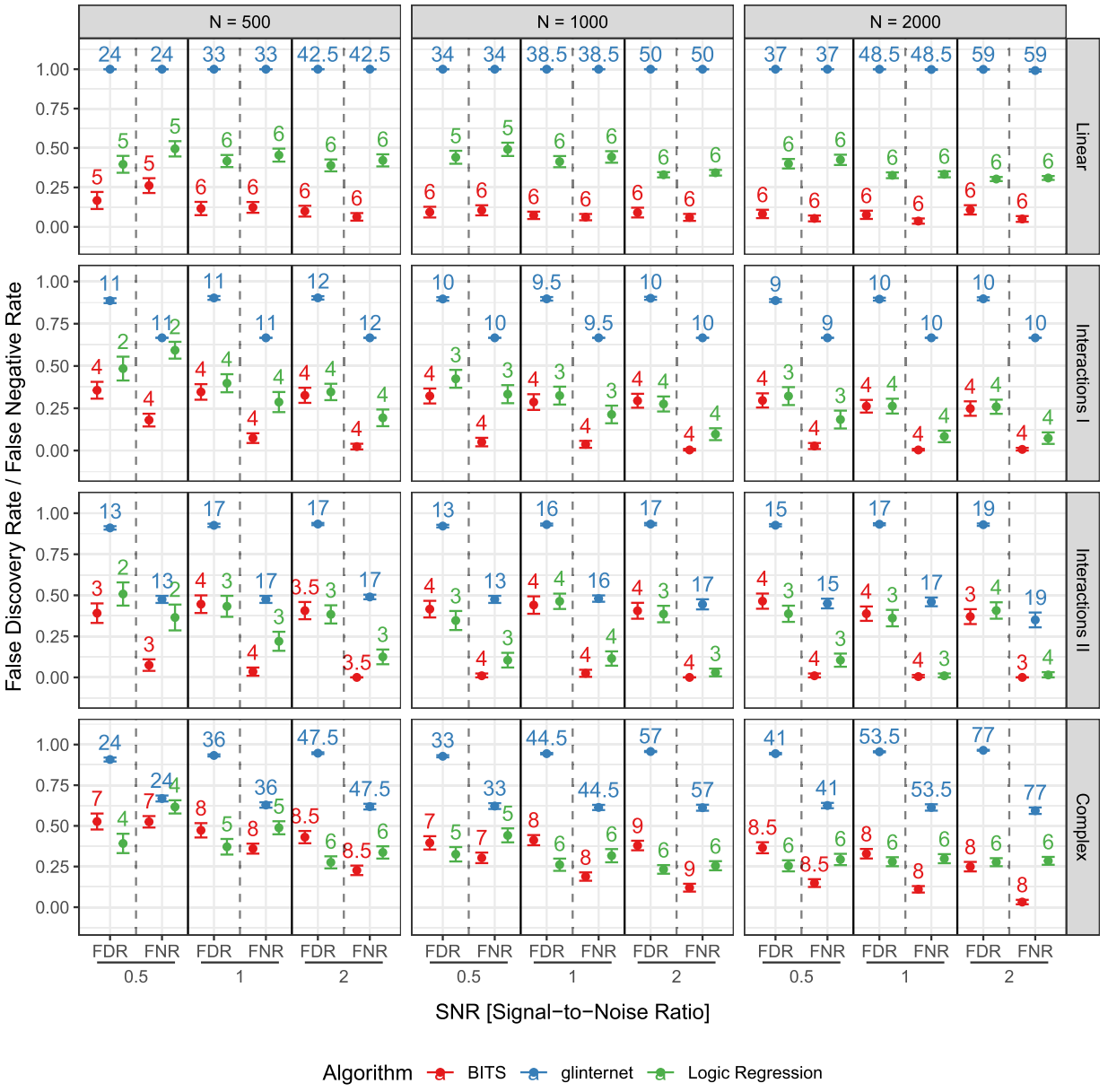

Fig. F.11. False discovery rate and false negative rate of terms identified by the optimal (hyperparameter-tuned) models with asymptotic 95% confidence intervals in the simulation study for BITS, glinternet, and logic regression. The values displayed above the confidence intervals indicate the median number of identified terms in the respective optimal models.

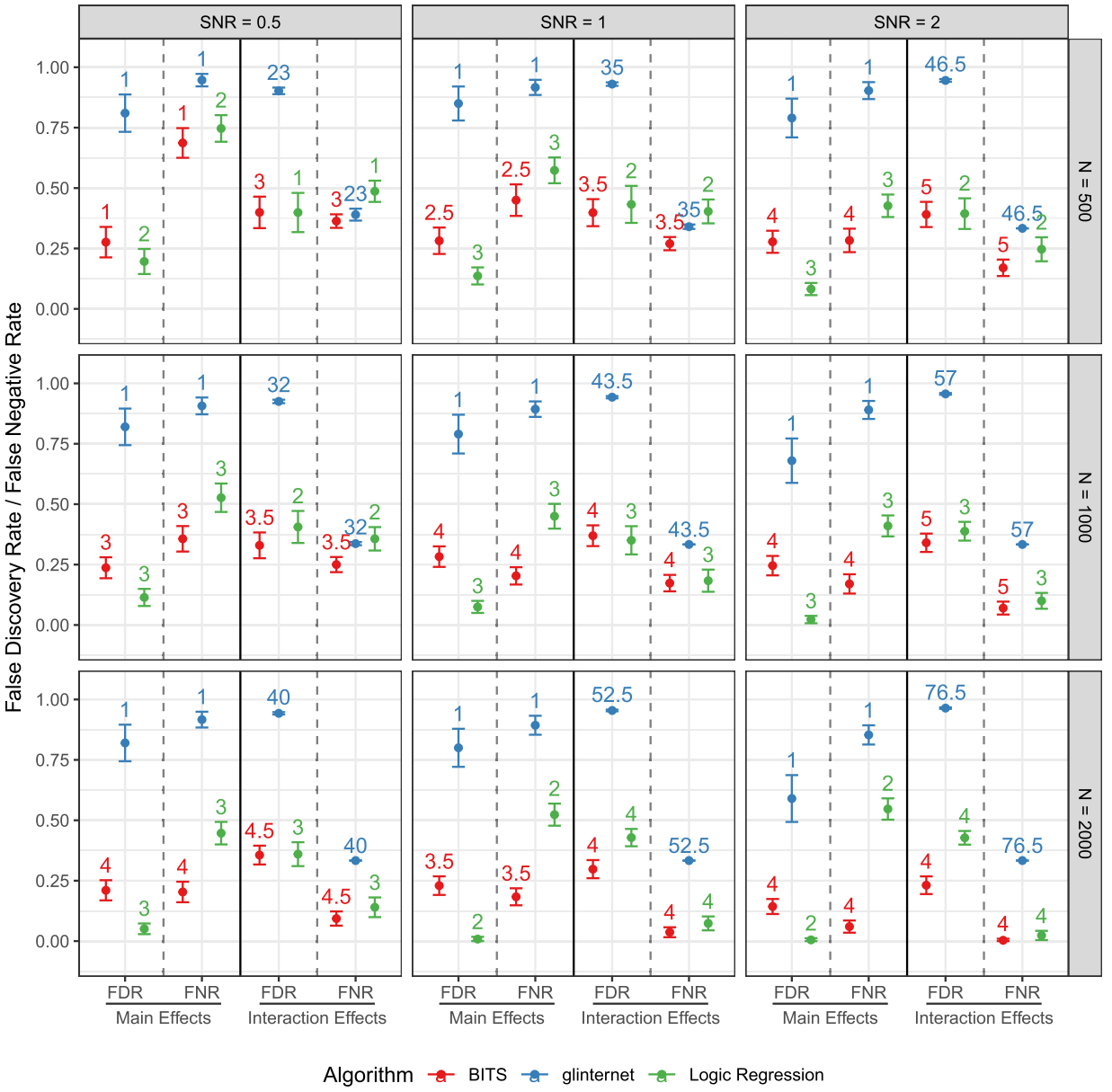

Fig. F.12. False discovery rate and false negative rate of identified main effects and interaction effects by the optimal (hyperparameter-tuned) models with asymptotic 95% confidence intervals in the complex simulation scenario for BITS, glinternet, and logic regression. The values displayed above the confidence intervals indicate the median numbers of identified main effects and interaction effects, respectively, in the corresponding optimal models.

Appendix G. Computation times

For the complex simulation scenario considering a medium sample size of $n = 1000$ and a medium signal-to-noise ratio of SNR = 1, times for model fitting and prediction were collected. The computations were performed using an Intel Xeon Gold 6136 CPU clocked at 3.0 GHz. For this computation time experiment, parallel computing was disabled to gather realistic single model evaluation times.

In Table G.3, the mean model evaluation over 100 repetitions are presented. BITS using a complete search takes the longest time for a full model fitting and prediction cycle due to considering all possible interactions of up to an interaction order of three. However, using a hybrid search, that was also employed in the application to a high-dimensional real data set in Section 5.2, the time reduces considerably from 283.8 s to 8.1 s. The computation times required by glinternet, RuleFit, random forests, and logic regression are between the computation times of BITS using a hybrid and a complete search. In this experiment, sprinter and MARS are the fastest methods. These two methods, however, yielded inferior predictive performances in the simulation study.

Table G.3
Mean model evaluation times in seconds for the complex simulation scenario with $n = 1000$ and SNR = 1.

Method	Time
BITS Complete	283.8
BITS Hybrid	8.1
glinternet	114.2
sprinter	0.7
MARS	2.7
RuleFit	21.2
Random Forests	25.2
Logic Regression	64.1

Appendix H. SALIA model

In the following model constructed by BITS, the subscripts $_A$, $_D$, and $_R$ denote, as defined in Section 3.1.1, additive, dominant, and recessive modes of inheritance, respectively. The superscript ^{c} denotes variable negations, as defined in Section 3.1.2. This model fitted in the application to the SALIA data discussed in Section 5.1 is given by

$$
\begin{aligned}
\operatorname{logit}(\mathbb{P}(Y=1)) = \; & -1.85 \\
& +0.41 \cdot \mathrm{rs142402365}_D \\
& -4.17 \cdot \mathrm{rs41288045}_R \\
& +0.60 \cdot \mathrm{rs9269815}_D \\
& +0.67 \cdot \mathrm{rs17885382}_D \qquad \text{Main effects} \\
& +7.90 \cdot \mathrm{rs1060176}_R \cdot \mathrm{rs151025335}_A \\
& +7.42 \cdot \mathrm{rs1064701}^c_D \cdot \mathrm{rs113505515}_D \\
& -4.58 \cdot \mathrm{rs41287215}_A \cdot \mathrm{rs149121504}_D \\
& +1.41 \cdot \mathrm{rs142402365}_D \cdot \mathrm{rs9269910}^c_A \\
& -0.93 \cdot \mathrm{rs9269799}^c_D \cdot \mathrm{rs74626234}^c_R \\
& +1.62 \cdot \mathrm{rs112796209}^c_R \cdot \mathrm{rs113322920}_R \\
& +0.13 \cdot \mathrm{rs9269831}_A \cdot \mathrm{rs9270143}^c_D \\
& -0.36 \cdot \mathrm{rs35789108}_D \cdot \mathrm{rs17884945}^c_R \\
& +2.28 \cdot \mathrm{rs149121504}_D \cdot \mathrm{rs34578704}_D \\
& +1.14 \cdot \mathrm{rs9270143}^c_D \cdot \mathrm{rs188759419}_A \qquad \text{Two-way interaction effects} \\
& +7.12 \cdot \mathrm{rs1064701}^c_D \cdot \mathrm{rs112353158}_D \cdot \mathrm{rs9270303}_D \\
& +2.08 \cdot \mathrm{rs9269693}_R \cdot \mathrm{rs35789108}^c_D \cdot \mathrm{rs145244672}_D \\
& +0.48 \cdot \mathrm{rs35789108}_D \cdot \mathrm{rs36101847}_R \cdot \mathrm{rs17884945}_R \\
& +5.65 \cdot \mathrm{rs35789108}_D \cdot \mathrm{rs17884945}_R \cdot \mathrm{rs17885382}_D. \qquad \text{Three-way interaction effects}
\end{aligned}
$$

Appendix I. Dorothea model

The model fitted by BITS in the application to the Dorothea data discussed in Section 5.2 is given by

$$
\begin{aligned}
\operatorname{logit}(\mathbb{P}(Y=1)) = \; & 2.50 \\
& +0.25 \cdot X_{412} + 0.37 \cdot X_{3015} + 1.77 \cdot X_{9696} \\
& -0.37 \cdot X_{12129} + 0.43 \cdot X_{13350} + 0.39 \cdot X_{16665} \\
& +1.12 \cdot X_{17739} + 1.73 \cdot X_{22774} + 0.42 \cdot X_{24989} \\
& +0.08 \cdot X_{28052} + 1.24 \cdot X_{29053} + 0.98 \cdot X_{33512} \\
& +0.85 \cdot X_{33530} + 0.05 \cdot X_{38083} - 0.58 \cdot X_{38862} \\
& +0.26 \cdot X_{45340} + 0.72 \cdot X_{50526} + 0.90 \cdot X_{52876} \\
& +0.21 \cdot X_{56636} + 0.11 \cdot X_{58689} + 0.19 \cdot X_{59960} \\
& +0.68 \cdot X_{66320} + 0.26 \cdot X_{69371} + 0.55 \cdot X_{69396} \\
& +0.04 \cdot X_{74997} + 0.68 \cdot X_{79103} + 0.54 \cdot X_{81380} \\
& +0.59 \cdot X_{87608} + 0.23 \cdot X_{89451} + 0.55 \cdot X_{91895} \\
& +0.40 \cdot X_{92539} + 0.26 \cdot X_{92586} + 0.99 \cdot X_{92702} \\
& +0.18 \cdot X_{93780} + 0.97 \cdot X_{94471} + 0.92 \cdot X_{94726} \\
& +0.52 \cdot X_{95598} + 0.13 \cdot X_{96078} \qquad \text{Main effects} \\
& -0.44 \cdot X^c_{2526} \cdot X^c_{26175} - 0.08 \cdot X^c_{2526} \cdot X^c_{30728} \\
& -0.34 \cdot X^c_{3164} \cdot X^c_{98301} - 1.59 \cdot X^c_{4296} \cdot X^c_{8632} \\
& +1.71 \cdot X_{8505} \cdot X_{99504} - 0.65 \cdot X^c_{15130} \cdot X^c_{87815} \\
& -0.30 \cdot X^c_{25762} \cdot X^c_{98301} - 0.60 \cdot X^c_{26175} \cdot X^c_{98301} \\
& +1.31 \cdot X_{41734} \cdot X_{92539} - 0.42 \cdot X^c_{75364} \cdot X^c_{98301}. \qquad \text{Two-way interaction effects}
\end{aligned}
$$

Data availability

The code for generating the simulated data sets is publicly available on Zenodo (Lau, 2025). The Dorothea data is available at the UCI Machine Learning Repository (https://doi.org/10.24432/C5NK6X). The website for assessing the test data performance for the Dorothea data (https://codalab.lisn.upsaclay.fr/competitions/7363) was accessed in July 2023.

References

Asghari, M., Fathollahi-Fard, A.M., Mirzapour Al-e-hashem, S.M.J., Dulebenets, M.A., 2022. Transformation and linearization techniques in optimization: a state-of-the-art survey. Mathematics 10. https://doi.org/10.3390/math10020283.

Bénard, C., Biau, G., da Veiga, S., Scornet, E., 2021. Interpretable random forests via rule extraction. In: Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. PMLR, pp. 937–945. https://proceedings.mlr.press/v130/benard21a.html.

Bertsimas, D., Digalakis Jr, V., 2023. Improving stability in decision tree models. https://doi.org/10.48550/arXiv.2305.17299.

Bertsimas, D., Dunn, J., 2017. Optimal classification trees. Mach. Learn. 106, 1039–1082. https://doi.org/10.1007/s10994-017-5633-9.

Bien, J., Taylor, J., Tibshirani, R., 2013. A lasso for hierarchical interactions. Ann. Stat. 41, 1111–1141. https://doi.org/10.1214/13-AOS1096.

Boruah, A.N., Biswas, S.K., Bandyopadhyay, S., 2023. Transparent rule generator random forest (TRG-RF): an interpretable random forest. Evolv. Syst. 14, 69–83. https://doi.org/10.1007/s12530-022-09434-4.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. https://doi.org/10.1023/A:1010933404324.

Breiman, L., Friedman, J.H., Stone, C.J., Olshen, R.A., 1984. Classification and Regression Trees. CRC Press, Boca Raton, Florida, USA.

Bühlmann, P., Hothorn, T., 2007. Boosting algorithms: regularization, prediction and model fitting. Stat. Sci. 22, 477–505. https://doi.org/10.1214/07-STS242.

Calus, M.P., Vandenplas, J., 2018. SNPrune: an efficient algorithm to prune large SNP array and sequence datasets based on high linkage disequilibrium. Genet. Sel. Evol. 50, 34. https://doi.org/10.1186/s12711-018-0404-z.

Che, R., Motsinger-Reif, A., 2013. Evaluation of genetic risk score models in the presence of interaction and linkage disequilibrium. Front. Genet. 4, 138. https://doi.org/10.3389/fgene.2013.00138.

Chen, T., Guestrin, C., 2016. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, pp. 785–794.

Clarke, A., Vyse, T.J., 2009. Genetics of rheumatic disease. Arthritis Res. Ther. 11, 248. https://doi.org/10.1186/ar2781.

Dai, B., Shen, X., Pan, W., 2024. Significance tests of feature relevance for a black-box learner. IEEE Trans. Neural Netw. Learn. Syst. 35, 1898–1911. https://doi.org/10.1109/TNNLS.2022.3185742.

Das, D., Duy, V.N.L., Hanada, H., Tsuda, K., Takeuchi, I., 2022. Fast and more powerful selective inference for sparse high-order interaction model. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9999–10007.

Dudbridge, F., 2013. Power and predictive accuracy of polygenic risk scores. PLoS Genet. 9, 1–17. https://doi.org/10.1371/journal.pgen.1003348.

Dusseldorp, E., Conversano, C., Van Os, B.J., 2010. Combining an additive and tree-based regression model simultaneously: STIMA. J. Comput. Graph. Stat. 19, 514–530. https://doi.org/10.1198/jcgs.2010.06089.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Ann. Stat. 32, 407–499. https://doi.org/10.1214/009053604000000067.

Fokkema, M., 2020. Fitting prediction rule ensembles with R package pre. J. Stat. Softw. 92, 1–30. https://doi.org/10.18637/jss.v092.i12.

Friedman, J.H., 1991. Multivariate adaptive regression splines. Ann. Stat. 19, 1–67. https://doi.org/10.1214/aos/1176347963.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 29, 1189–1232. https://doi.org/10.1214/aos/1013203451.

Friedman, J.H., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting. Ann. Stat. 28, 337–407. https://doi.org/10.1214/aos/1016218223.

Friedman, J.H., Popescu, B.E., 2008. Predictive learning via rule ensembles. Ann. Appl. Stat. 2, 916–954. https://doi.org/10.1214/07-AOAS148.

Guyon, I., Gunn, S., Ben-Hur, A., Dror, G., 2004. Result analysis of the NIPS 2003 feature selection challenge. In: Advances in Neural Information Processing Systems. MIT Press. https://proceedings.neurips.cc/paper/2004/file/5e751896e527c862bf67251a474b3819-Paper.pdf.

Hastie, T., Tibshirani, R., Tibshirani, R., 2020. Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons. Stat. Sci. 35, 579–592. https://doi.org/10.1214/19-STS733.

Hazimeh, H., Mazumder, R., 2020. Learning hierarchical interactions at scale: a convex optimization approach. In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. PMLR, pp. 1833–1843. https://proceedings.mlr.press/v108/hazimeh20a.html.

Hooker, G., 2004. Discovering additive structure in black box functions. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, pp. 575–580.

Hüls, A., Ickstadt, K., Schikowski, T., Krämer, U., 2017. Detection of gene-environment interactions in the presence of linkage disequilibrium and noise by using genetic risk scores with internal weights from elastic net regression. BMC Genet. 18, 55. https://doi.org/10.1186/s12863-017-0519-1.

Hüls, A., Czamara, D., 2020. Methodological challenges in constructing DNA methylation risk scores. Epigenetics 15, 1–11. https://doi.org/10.1080/15592294.2019.1644879.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., 2017. Lightgbm: a highly efficient gradient boosting decision tree. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680. https://doi.org/10.1126/science.220.4598.671.

Kooperberg, C., Ruczinski, I., 2023. LogicReg: logic regression. https://CRAN.R-project.org/package=LogicReg, R package version 1.6.6.

Krämer, U., Herder, C., Sugiri, D., Strassburger, K., Schikowski, T., Ranft, U., Rathmann, W., 2010. Traffic-related air pollution and incident type 2 diabetes: results from the salia cohort study. Environ. Health Perspect. 118, 1273–1279. https://doi.org/10.1289/ehp.0901689.

Lau, M., 2024. BITS: boosting interaction tree stumps. https://github.com/michlau/BITS. R package version 1.0.0.

Lau, M., 2025. BITS paper code. Zenodo. https://doi.org/10.5281/zenodo.14593699.

Lau, M., Kress, S., Schikowski, T., Schwender, H., 2023. Efficient gene–environment interaction testing through bootstrap aggregating. Sci. Rep. 13, 937. https://doi.org/10.1038/s41598-023-28172-4.

Lau, M., Schikowski, T., Schwender, H., 2024. logicDT: a procedure for identifying response-associated interactions between binary predictors. Mach. Learn. 113, 933–992. https://doi.org/10.1007/s10994-023-06488-6.

Lau, M., Wigmann, C., Kress, S., Schikowski, T., Schwender, H., 2022. Evaluation of tree-based statistical learning methods for constructing genetic risk scores. BMC Bioinform. 23, 97. https://doi.org/10.1186/s12859-022-04634-w.

Lengerich, B., Tan, S., Chang, C.H., Hooker, G., Caruana, R., 2020. Purifying interaction effects with the functional ANOVA: an efficient algorithm for recovering identifiable additive models. In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. PMLR, pp. 2402–2412. https://proceedings.mlr.press/v108/lengerich20a.html.

Lim, M., Hastie, T., 2015. Learning interactions via hierarchical group-lasso regularization. J. Comput. Graph. Stat. 24, 627–654. https://doi.org/10.1080/10618600.2014.938812.

Lim, M., Hastie, T., 2021. glinternet: learning interactions via hierarchical group-lasso regularization. https://CRAN.R-project.org/package=glinternet. R package version 1.0.12.

Mavaddat, N., Michailidou, K., Dennis, J., Lush, M., Fachal, L., Lee, A., Tyrer, J.P., Chen, T.H., Wang, Q., Bolla, M.K., Yang, X., Adank, M.A., Ahearn, T., Aittomäki, K., Allen, J., Andrulis, I.L., Anton-Culver, H., Antonenkova, N.N., Arndt, V., Aronson, K.J., Auer, P.L., Auvinen, P., Barrdahl, M., Beane Freeman, L.E., Beckmann, M.W., Behrens, S., Benitez, J., Bermisheva, M., Bernstein, L., Blomqvist, C., Bogdanova, N.V., Bojesen, S.E., Bonanni, B., Børresen-Dale, A.L., Brauch, H., Bremer, M., Brenner, H., Brentnall, A., Brock, I.W., Brooks-Wilson, A., Brucker, S.Y., Brüning, T., Burwinkel, B., Campa, D., Carter, B.D., Castelao, J.E., Chanock, S.J., Chlebowski, R., Christiansen, H., Clarke, C.L., Collée, J.M., Cordina-Duverger, E., Cornelissen, S., Couch, F.J., Cox, A., Cross, S.S., Czene, K., Daly, M.B., Devilee, P., Dörk, T., Dos-Santos-Silva, I., Dumont, M., Durcan, L., Dwek, M., Eccles, D.M., Ekici, A.B., Eliassen, A.H., Ellberg, C., Engel, C., Eriksson, M., Evans, D.G., Fasching, P.A., Figueroa, J., Fletcher, O., Flyger, H., Försti, A., Fritschi, L., Gabrielson, M., Gago-Dominguez, M., Gapstur, S.M., García-Sáenz, J.A., Gaudet, M.M., Georgoulias, V., Giles, G.G., Gilyazova, I.R., Glendon, G., Goldberg, M.S., Goldgar, D.E., González-Neira, A., Grenaker Alnæs, G.I., Grip, M., Gronwald, J., Grundy, A., Guénel, P., Haeberle, L., Hahnen, E., Haiman, C.A., Håkansson, N., Hamann, U., Hankinson, S.E., Harkness, E.F., Hart, S.N., He, W., Hein, A., Heyworth, J., Hillemanns, P., Hollestelle, A., Hooning, M.J., Hoover, R.N., Hopper, J.L., Howell, A., Huang, G., Humphreys, K., Hunter, D.J., Jakimovska, M., Jakubowska, A., Janni, W., John, E.M., Johnson, N., Jones, M.E., Jukkola-Vuorinen, A., Jung, A., Kaaks, R., Kaczmarek, K., Kataja, V., Keeman, R., Kerin, M.J., Khusnutdinova, E., Kiiski, J.I., Knight, J.A., Ko, Y.D., Kosma, V.M., Koutros, S., Kristensen, V.N., Krüger, U., Kühl, T., Lambrechts, D., Le Marchand, L., Lee, E., Lejbkowicz, F., Lilyquist, J., Lindblom, A., Lindström, S., Lissowska, J., Lo, W.Y., Loibl, S., Long, J., Lubiński, J., Lux, M.P., MacInnis, R.J., Maishman, T., Makalic, E., Maleva Kostovska, I., Mannermaa, A., Manoukian, S., Margolin, S., Martens, J.W.M., Martinez, M.E., Mavroudis, D., McLean, C., Meindl, A., Menon, U., Middha, P., Miller, N., Moreno, F., Mulligan, A.M., Mulot, C., Muñoz-Garzon, V.M., Neuhausen, S.L., Nevanlinna, H., Neven, P., Newman, W.G., Nielsen, S.F., Nordestgaard, B.G., Norman, A., Offit, K., Olson, J.E., Olsson, H., Orr, N., Pankratz, V.S., Park-Simon, T.W., Perez, J.I.A., Pérez-Barrios, C., Peterlongo, P., Peto, J., Pinchev, M., Plaseska-Karanfilska, D., Polley, E.C., Prentice, R., Presneau, N., Prokofyeva, D., Purrington, K., Pylkäs, K., Rack, B., Radice, P., Rau-Murthy, R., Rennert, G., Rennert, H.S., Rhenius, V., Robson, M., Romero, A., Ruddy, K.J., Ruebner, M., Saloustros, E., Sandler, D.P., Sawyer, E.J., Schmidt, D.F., Schmutzler, R.K., Schneeweiss, A., Schoemaker, M.J., Schumacher, F., Schürmann, P., Schwentner, L., Scott, C., Scott, R.J., Seynaeve, C., Shah, M., Sherman, M.E., Shrubsole, M.J., Shu, X.O., Slager, S., Smeets, A., Sohn, C., Soucy, P., Southey, M.C., Spinelli, J.J., Stegmaier, C., Stone, J., Swerdlow, A.J., Tamimi, R.M., Tapper, W.J., Taylor, J.A., Terry, M.B., Thöne, K., Tollenaar, R.A.E.M., Tomlinson, I., Truong, T., Tzardi, M., Ulmer, H.U., Untch, M., Vachon, C.M., van Veen, E.M., Vijai, J., Weinberg, C.R., Wendt, C., Whittemore, A.S., Wildiers, H., Willett, W., Winqvist, R., Wolk, A., Yang, X.R., Yannoukakos, D., Zhang, Y., Zheng, W., Ziogas, A., Dunning, A.M., Thompson, D.J., Chenevix-Trench, G., Chang-Claude, J., Schmidt, M.K., Hall, P., Milne, R.L., Pharoah, P.D.P., Antoniou, A.C., Chatterjee, N., Kraft, P., García-Closas, M., Simard, J., Easton, D.F., 2019. Polygenic risk scores for prediction of breast cancer and breast cancer subtypes. Am. J. Hum. Genet. 104, 21–34. https://doi.org/10.1016/j.ajhg.2018.11.002.

Meinshausen, N., 2007. Relaxed lasso. Comput. Stat. Data Anal. 52, 374–393. https://doi.org/10.1016/j.csda.2006.12.019.

Meinshausen, N., 2010. Node harvest. Ann. Appl. Stat. 4, 2049–2072. https://doi.org/10.1214/10-AOAS367.

Meinshausen, N., Meier, L., Bühlmann, P., 2009. p-values for high-dimensional regression. J. Am. Stat. Assoc. 104, 1671–1681. https://doi.org/10.1198/jasa.2009.tm08647.

Milborrow, S., 2021. Earth: multivariate adaptive regression splines. https://CRAN.R-project.org/package=earth. R package version 5.3.1.

Murthy, S.K., Kasif, S., Salzberg, S., 1994. A system for induction of oblique decision trees. J. Artif. Intell. Res. 2, 1–32. https://doi.org/10.1613/jair.63.

Ottman, R., 1996. Gene–environment interaction: definitions and study design. Prev. Med. 25, 764–770. https://doi.org/10.1006/pmed.1996.0117.

Petersen, A.K., Krumsiek, J., Wägele, B., Theis, F.J., Wichmann, H.E., Gieger, C., Suhre, K., 2012. On the hypothesis-free testing of metabolite ratios in genome-wide and metabolome-wide association studies. BMC Bioinform. 13, 120. https://doi.org/10.1186/1471-2105-13-120.

Privé, F., Aschard, H., Blum, M.G.B., 2019. Efficient implementation of penalized regression for genetic risk prediction. Genetics 212, 65–74. https://doi.org/10.1534/genetics.119.302019.

R Core Team, 2020. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Rencher, A.C., Schaalje, G.B., 2007. Linear Models in Statistics. John Wiley & Sons, Hoboken, New Jersey, USA.

Ruczinski, I., Kooperberg, C., LeBlanc, M., 2003. Logic regression. J. Comput. Graph. Stat. 12, 475–511. https://doi.org/10.1198/1061860032238.

Scherer, N., Sekula, P., Pfaffelhuber, P., Schlosser, P., 2021. pgainsim: an R-package to assess the mode of inheritance for quantitative trait loci in GWAS. Bioinformatics 37, 3061–3063. https://doi.org/10.1093/bioinformatics/btab150.

Schikowski, T., Sugiri, D., Ranft, U., Gehring, U., Heinrich, J., Wichmann, H.E., Krämer, U., 2005. Long-term air pollution exposure and living close to busy roads are associated with COPD in women. Respir. Res. 6, 152. https://doi.org/10.1186/1465-9921-6-152.

She, Y., Wang, Z., Jiang, H., 2018. Group regularized estimation under structural hierarchy. J. Am. Stat. Assoc. 113, 445–454. https://doi.org/10.1080/01621459.2016.1260470.

Sun, X., Wang, Z., Ding, R., Han, S., Zhang, D., 2022. Puregam: learning an inherently pure additive model. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, pp. 1728–1738.

Suzumura, S., Nakagawa, K., Umezu, Y., Tsuda, K., Takeuchi, I., 2017. Selective inference for sparse high-order interaction models. In: Proceedings of the 34th International Conference on Machine Learning. PMLR, pp. 3338–3347. https://proceedings.mlr.press/v70/suzumura17a.html.

Suzumura, S., Nakagawa, K., Umezu, Y., Tsuda, K., Takeuchi, I., 2021. Selective inference for high-order interaction features selected in a stepwise manner. IPSJ Trans. Bioinform. 14, 1–11. https://doi.org/10.2197/ipsjtbio.14.1.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B, Methodol. 58, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Uffelmann, E., Huang, Q.Q., Munung, N.S., De Vries, J., Okada, Y., Martin, A.R., Martin, H.C., Lappalainen, T., Posthuma, D., 2021. Genome-wide association studies. Nat. Rev. Methods Primers 1, 59. https://doi.org/10.1038/s43586-021-00056-9.

Wang, C., Jiang, B., Zhu, L., 2021. Penalized interaction estimation for ultrahigh dimensional quadratic regression. Stat. Sin. 31, 1549–1570. https://doi.org/10.5705/ss.202019.0081. https://www.jstor.org/stable/27034830.

Wright, M.N., Ziegler, A., 2017. ranger: a fast implementation of random forests for high dimensional data in C++ and R. J. Stat. Softw. 77, 1–17. https://doi.org/10.18637/jss.v077.i01.

Wu, M., Huang, J., Ma, S., 2018. Identifying gene-gene interactions using penalized tensor regression. Stat. Med. 37, 598–610. https://doi.org/10.1002/sim.7523.

Yan, X., Bien, J., 2017. Hierarchical sparse modeling: a choice of two group lasso formulations. Stat. Sci. 32, 531–560. https://doi.org/10.1214/17-STS622.

Yu, G., 2019. sprintr: sparse reluctant interaction modeling. https://CRAN.R-project.org/package=sprintr. R package version 0.9.0.

Yu, G., Bien, J., Tibshirani, R., 2019. Reluctant interaction modeling. https://doi.org/10.48550/ARXIV.1907.08414.

Zhang, X., Shi, X., Liu, Y., Liu, X., Ma, S., 2023. A general framework for identifying hierarchical interactions and its application to genomics data. J. Comput. Graph. Stat., 1–11. https://doi.org/10.1080/10618600.2022.2152034.