Received: 13 May 2023 | Revised: 25 July 2023 | Accepted: 11 October 2023

DOI: 10.1002/sta4.633

ORIGINAL ARTICLE

WILEY

A confidence machine for sparse high-order interaction model

Diptesh Das[1] | Eugene Ndiaye[2] | Ichiro Takeuchi[1,3]

[1]Department of Mechanical System Engineering, Nagoya University, Nagoya, Japan

[2]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA

[3]Data-driven Biomedical Science Team, RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan

Correspondence
Diptesh Das and Ichiro Takeuchi, Department of Mechanical System Engineering, Nagoya University, Nagoya, Japan.
Email: diptesh.das@edu.k.u-tokyo.ac.jp and ichiro.takeuchi@mae.nagoya-u.ac.jp

Funding information
JST Moonshot R&D; JST CREST; MEXT KAKENHI; New Energy and Industrial Technology Development Organization; RIKEN Center for Advanced Intelligence Project

Abstract

In predictive modelling for high-stake decision-making, predictors must be not only accurate but also reliable. Conformal prediction (CP) is a promising approach for obtaining the coverage of prediction results with fewer theoretical assumptions. To obtain the prediction set by so-called full-CP, we need to refit the predictor for all possible values of prediction results, which is only possible for simple predictors. For complex predictors such as random forests (RFs) or neural networks (NNs), split-CP is often employed where the data is split into two parts: one part for fitting and another for computing the prediction set. Unfortunately, because of the reduced sample size, split-CP is inferior to full-CP both in fitting as well as prediction set computation. In this paper, we develop a full-CP of sparse high-order interaction model (SHIM), which is sufficiently flexible as it can take into account high-order interactions among variables. We resolve the computational challenge for full-CP of SHIM by introducing a novel approach called homotopy mining. Through numerical experiments, we demonstrate that SHIM is as accurate as complex predictors such as RF and NN and enjoys the superior statistical power of full-CP.

1 | INTRODUCTION

The uncertainty in data-driven analysis is a major concern, particularly in risk-sensitive automated decision-making problems (for example, in medical diagnosis and criminal justice). Several strategies exist to quantify the uncertainty of a point estimator. For example, the Bayesian approach (Clyde et al., 2021) can provide a strong coverage bound, but requires the assumption on prior distribution. The PAC analysis is another approach that provides bounds on the probability of error (Alquier, 2021; Shawe-Taylor & Williamson, 1997). As another direction, selective inference has been studied for quantifying the uncertainty of data-driven knowledge (Das et al., 2021; Duy et al., 2020; Fithian et al., 2014; Le Duy & Takeuchi, 2021; Lee et al., 2016). The conformal prediction (CP) is one such uncertainty quantification method that is very generic, and it is applicable to almost any point estimators (Shafer & Vovk, 2008; Vovk et al., 2005). The CP method has recently gained significant attention (Barber et al., 2023; Gibbs et al., 2023; Lei & Candès, 2021) as it can provide valid finite sample statistical coverage guarantee at any nominal level as long as data are independently and identically distributed (i.i.d.). The coverage guarantee provided is valid even when the model is misspecified. In this paper we are interested in the *full-CP* where the full data is used to compute the CP set. An alternative to this approach is the *split-CP* in which the data is split into two parts: one part is used for fitting and the remaining part is used for computing the CP set. The essential idea of the full-CP framework can be stated as follows: Given a training set $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and a test input instance x_{n+1} which are both i.i.d., the goal

Most of the work of this article was done while Diptesh Das was affiliated with Department of Mechanical System Engineering, Nagoya University, Japan. However, Diptesh Das is currently affiliatied with Department of Computational Biology and Medical Sciences, University of Tokyo, Japan. Eugene Ndiaye's affiliation has also been changed, and he is currently affiliated with Machine Learning Group, Apple, Paris, France. All correspondence should be directed to current email address of Diptesh Das <diptesh.das@edu.k.u- tokyo.ac.jp> or (Ichiro Takeuchi,<ichiro.takeuchi@mae.nagoya- u.ac.jp>).

of the CP is to construct a $100(1-\alpha)\%$ prediction set that contains the unobserved y_{n+1}. Here, $\alpha \in [0,1]$ represents the coverage level. In other words, we are interested in all possible cases of the augmented dataset in the form of $\mathcal{D}_n \cup (x_{n+1}, \tau)$, such that $y_{n+1} = \tau$ is *typical* for the known data $\{\mathcal{D}_n, x_{n+1}\}$. This typicality is measured by a typicalness function $\pi(\cdot)$, which is also called the *p*-value in analogy to the classical hypothesis testing. In other words, the CP set for x_{n+1} is defined as the set of all τ for which the null hypothesis $H_0 : y_{n+1} = \tau$ is not rejected against the alternative hypothesis $H_1 : y_{n+1} \neq \tau$. In the CP, although the coverage property is satisfied even when the model is misspecified, it is better to use sufficiently complex models for complex data because this enables us to obtain a more compact prediction set (i.e., shorter prediction intervals). Furthermore, it is important to note that the prediction set obtained by a full-CP is more compact than that obtained by a split-CP because only a part of instances in the available dataset is used for constructing the prediction set in split-CP. This means that it is valuable to construct a full-CP algorithm for sufficiently complex models. In this paper, we considered the *sparse high-order interaction model (SHIM)* which can represent complex nonlinear relationship, and proposed a *homotopy-mining* method to efficiently compute the *full-CP* set. We call the resulting machine as *SHIM confidence machine*. SHIM is formulated as a weighted sum of conjunction rules that are highly interpretable as well as accurate decision sets (Lakkaraju et al., 2016; Das et al., 2019). SHIM can capture the combinatorial interactions of multiple factors, which can prove to be beneficial for deciphering complex data. A conjunction rule of a SHIM looks like

$$I(-1.5 \leq x_{\cdot,1} \leq 2.3) \wedge I(x_{\cdot,3} \geq 20.0) \wedge I(x_{\cdot,5} \leq 7.5),$$

where $I(\cdot)$ refers to the indicator function. An intuitive illustration of a SHIM is shown in Figure 1A.

Related works: Since its inception, there have been several extensions and applications of the CP framework in diverse directions. Examples include the choice of conformity score and statistical efficiency (Lei & Wasserman, 2014; Lei et al., 2013), high-dimensional regression (Lei et al., 2018), classification (Lei, 2014; Sadinle et al., 2019), active learning (Ho & Wechsler, 2008), time series (Xu & Xie, 2021), few-shot learning (Fisch et al., 2021), text and speech completion (Dey et al., 2021), image classification (Angelopoulos et al., 2020), outlier detection (Laxhammar & Falkman, 2015; Bates et al., 2021). Recently, some approximate computation of the conformal set was proposed for regression in Ndiaye and Takeuchi (2019); Ndiaye and Takeuchi (2021); Abad et al. (2022) and for classification, significant advancement has been made in Cherubin et al. (2021) to compute exact full-CP.

Despite its attractive properties, the application of exact, full-CP in many practical problems remains an open problem owing to its high computational cost. By definition, the full-CP framework requires the refitting of model by augmenting the data for every possible candidate $\forall \tau \in \mathbb{R}$ of the unobserved y_{n+1}. It means that in a regression setting, one needs to refit the model an infinite number of times for all possible candidates on the real line $(\forall \tau \in \mathbb{R})$, and check the conformity by computing the *p*-values $\pi(\tau)$.

Hence, an efficient computation of the full-CP set is possible only for a handful of simple models (e.g., ordinary least square regression (OLS), ridge regression) in which the solution is explicitly represented as a function of τ. This enables us to derive closed-form full-CP sets and avoids an exhaustive search over the real line (Nouretdinov et al., 2001). Recently, Lei (2019) proposed a homotopy method to efficiently compute the full-CP set of the LASSO. The homotopy method exploits the piece-wise linearity of the LASSO solutions and avoids the computational burden of all

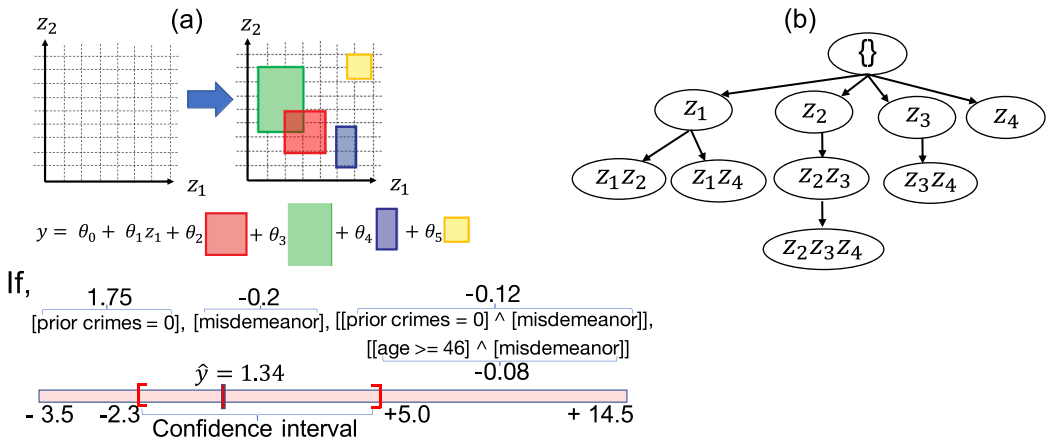


FIGURE 1 (a) An intuitive illustration of a SHIM solution space, which is a hyper rectangle (i.e., generalization of a rectangle for higher dimensions that essentially represents a cartesian product of intervals). The decision sets, the predicted response $(\hat{y})$ and the associated prediction interval for a randomly chosen example of data from the ProPublica 2-year recidivism criminal justice (COMPAS) dataset (Larson et al., 2016) are shown. (b) A tree of patterns has been constructed by exploiting the hierarchical structure of high-order interaction features (Note: not all patterns appear due to pruning)

possible candidates of y_{n+1} (see Section 3 for details). Unfortunately, such a structure does not exist for most of the models, and the application of full-CP is still an open question for complex models that can represent complex nonlinearity such as a random forest (RF), neural network (NN) etc., for which split-CP has been the only possible choice.

Contribution: Our main contribution in this paper is to develop an efficient algorithm to conduct exact full-CP for SHIM which can capture complex structures in the data by considering high-order interaction features. The exact full-CP for complex black box models such as RF and NN are intractable and an efficient computational method does not exist. For such black box models only split-CP has been possible. As of today, an efficient exact full-CP is possible only for simple regression models such as LASSO, ridge regression, and OLS. The proposed method adds SHIM to that list. To the best of our knowledge, this is also the first attempt to construct a conformal prediction set in the context of pattern mining model which is fitted with branch and bound approach. We also extended our framework to the elastic net and provided an algorithm to compute the exact full-CP set efficiently. The proposed method enables us to obtain a more compact prediction set (more compact than linear and comparable to non-linear models) by the full-CP for sufficiently flexible and complex SHIM. A SHIM uses higher-order interaction features and hence, the full-CP is computationally challenging; but we overcome this difficulty by introducing a method called *homotopy mining* which exploits the best of both homotopy and (pattern) mining methods. The computation of an exact full-CP for SHIM by homotopy mining can be interpreted as an extension of LASSO's exact full-CP in Lei (2019). The use of flexible but explainable machine learning models in high-stakes decision-making such as in healthcare, criminal justice, and other domains is highly appreciated in the literature (Angelino et al., 2018; Rudin, 2019). We believe that the exact full-CP of SHIMs has significant importance in practice where accuracy, statistical reliability, and interpretability are important. When a practitioner chooses SHIM as a predictor, the proposed method provides a computationally efficient solution to quantify the coverage of the prediction results. The source code is available at https://github.com/DipteshDas/CP-SHIM.

Notation: In this paper, for a natural number a, we use a notation $[a]=\{1,2,...,a\}$, the response vector of n instances is represented as $y\in\mathbb{R}^n$, while $y(\tau)\in\mathbb{R}^{n+1}$ is the augmented response vector constructed by augmenting the possible response value $\tau\in\mathbb{R}$ of the $(n+1)^{th}$ instance with y. Later we define $y(\tau)$ as a function of the variable τ as the vector $y(\tau)$ changes for different possible response value τ. The $y_i\in\mathbb{R},\forall i\in[n+1]$, represents the scalar response of the i^{th} instance, $X\in\mathbb{R}^{(n+1)\times p}$ is the design matrix of all $(n+1)$ instances, each having p features, $x_\ell\in\mathbb{R}^{n+1}$ is the ℓ^{th} column vector of X for $\ell\in[p]$, $X_{\mathcal{A}_\tau}\in\mathbb{R}^{(n+1)\times|\mathcal{A}_\tau|}\subseteq X$ is a smaller design matrix in which the columns are restricted to the elements of some subset $\mathcal{A}_\tau\subseteq[p]$.

2 | PROBLEM STATEMENT

Consider a regression problem with a response vector $y\in\mathbb{R}^n$ and m original covariate vectors $z_1,...,z_m$, where $z_\ell\in\mathbb{R}^n$ and $\ell\in[m]$. A high-order interaction model up to the dth order is then written as follows:

$$y=\sum_{\ell_1\in[m]}\theta_{\ell_1}z_{\ell_1}+\sum_{\substack{(\ell_1,\ell_2)\in[m]\times[m]\\ \ell_1\neq\ell_2}}\theta_{\ell_1,\ell_2}z_{\ell_1}z_{\ell_2}+...+\sum_{\substack{(\ell_1,...,\ell_d)\in[m]^d\\ \ell_1\neq...\neq\ell_d}}\theta_{\ell_1,...,\ell_d}z_{\ell_1}...z_{\ell_d}+\epsilon, \tag{1}$$

where $z_{\ell_1}...z_{\ell_d}$ is the element-wise product, scalar θ represents the coefficient and ϵ is the noise with $\mathbb{E}(\epsilon|X)=0$. In this study, we mainly consider each element of the original covariate vector $z_\ell\in\{0,1\}^n$. However, our model is equally applicable to covariate vectors defined in the domain $[0,1]^n$. To simplify the notation, it is convenient to write the high-order interaction model in (1) using the following matrix of concatenated vectors of all high-order interactions:

$$X=[\underbrace{z_1,...,z_m}_{1^{st}\text{order}},...,\underbrace{z_1...z_d,...,z_{m-d+1}...z_m}_{d^{th}\text{order}}]\in\mathbb{R}^{n\times p},$$

where $p:=\sum_{\kappa=1}^{d}\binom{m}{\kappa}$. Similarly, the coefficient vector associated with all possible high-order interaction terms can be written as follows:

$$\beta:=[\underbrace{\theta_1,...,\theta_m}_{1^{st}\text{order}},...,\underbrace{\theta_{1,...,d},...,\theta_{m-d+1,...,m}}_{d^{th}\text{order}}]^\top\in\mathbb{R}^p.$$

The high-order interaction model (1) is then simply written as a linear model $y=X\beta+\epsilon$. Unfortunately, p can be prohibitively large unless both m and d are fairly small. In the SHIM, we consider a sparse estimation of a high-order interaction model. An example of a SHIM is as follows:

$$y = \theta_3 z_3 + \theta_5 z_5 + \theta_{2,6} z_2 z_6 + \theta_{1,2,5,9} z_1 z_2 z_5 z_9 + \epsilon.$$

Before delving into our proposed method, we briefly overview the conformal prediction framework.

2.1 | Conformal prediction

A mere point estimation is insufficient for risk-sensitive automated decision-making problems (Angelino et al., 2018; Das et al., 2019; Rudin, 2019), such as in medical diagnosis and criminal justice. In such high-stake decision-making problems, if the estimators are equipped with the associated coverage information, the decision maker will be more informed and sufficiently confident to make a prudent decision when the stakes are high.

Full-CP: Given a labelled dataset $\mathcal{D}_n$ and a new observation x_{n+1}, the goal of the full-CP framework is to construct a set of likely values $\mathcal{C}(x_{n+1})$ of an unobserved y_{n+1} with a valid statistical coverage guarantee (Shafer & Vovk, 2008; Vovk et al., 2005), that is,

$$\mathbb{P}(y_{n+1} \in \mathcal{C}(x_{n+1})) \geq 1 - \alpha, \tag{2}$$

where $\alpha \in [0,1]$ determines the level of coverage. If we define a prediction function $\mu(\cdot)$ that maps the input X to the output y, then the essential idea of constructing a full-CP set is to fit a model $\mu_\tau(\cdot)$ with the augmented data $\mathcal{D}_n \cup (x_{n+1}, \tau)$ for every possible candidate $\tau \in \mathbb{R}$ and compare the prediction error of each instances. More precisely, let $y(\tau) = [y_1, \ldots, y_n, \tau]^\top \in \mathbb{R}^{n+1}$ be a vector augmented with τ. We define a score function that measures how well the model can predict each output variables; with the constraint that it should not depend on the order of the data instances (Lei, 2019). One such conformity score function generally used in linear regression setting is the (coordinate-wise) absolute residual, that is,

$$S(\tau) = |y(\tau) - \mu_\tau(X)|,$$

where we stack the input vectors in the design matrix $X = [x_1, \ldots, x_{n+1}]^\top$ in $\mathbb{R}^{(n+1)\times p}$. The conformity function can now be defined as

$$\pi(\tau) = 1 - \frac{1}{n+1} \sum_{i=1}^{n+1} \mathbb{1}_{S_i(\tau) \leq S_{n+1}(\tau)},$$

which evaluates how the prediction of the candidate τ using the fitted model $\mu_\tau(x_{n+1})$ is ranked compared to the prediction of the previously observed data y_i using $\mu_\tau(x_i)$. The conformal set is defined as

$$\mathcal{C}(x_{n+1}) = \{\tau : \pi(\tau) \geq \alpha\}, \tag{3}$$

which is merely the collection of candidates whose conformity is large enough. Vovk et al. (2005) showed that the defined conformal set $\mathcal{C}(x_{n+1})$ satisfies the coverage guarantee (2) as long as the data are i.i.d.

Split-CP: In split-CP (Papadopoulos et al., 2002), the data set $\mathcal{D}_n$ is split into two parts, defined as the training set $\mathcal{D}_{\text{tr}} = \{(x_1, y_1), \ldots, (x_{n'}, y_{n'})\}$ and the calibration set $\mathcal{D}_{\text{cal}} = \{(x_{n'+1}, y_{n'+1}), \ldots, (x_n, y_n)\}$, with $n' < n$. Then the model $(\mu^{\text{tr}}(\cdot))$ is fit with the training set $\mathcal{D}_{\text{tr}}$ only once, and the p-values, denoted as $\pi_{\text{split}}(\cdot)$, are determined using the calibration set $\mathcal{D}_{\text{cal}}$ as follows.

$$\pi_{\text{split}}(\tau) = 1 - \frac{1}{n - n'} \sum_{i=n'+1}^{n} \mathbb{1}_{S_i^{\text{cal}}(\tau) \leq S_{n+1}(\tau)},$$

where $S_i^{\text{cal}}(\tau) = |y_i - \mu^{\text{tr}}(x_i)|, \forall i \in [n'+1, n]$. Therefore, the split-CP can be defined as follows:

$$\mathcal{C}_{\text{split}}(x_{n+1}) = \{\tau : \pi_{\text{split}}(\tau) \geq \alpha\},$$

When the conformity score is defined as the absolute residual, then the split-CP set can be conveniently written as $\mathcal{C}_{\text{split}}(x_{n+1}) = \left[\mu^{\text{tr}}(x_{n+1}) \pm Q_{1-\alpha}^{\text{cal}}\right]$, where $Q_{1-\alpha}^{\text{cal}}$ is the $(1-\alpha)$ quantile of the calibration scores $S_i^{\text{cal}}, \forall i \in [n'+1, n]$.

Although split-CP enjoys the computational efficiency of single model fitting, it suffers from the poor statistical efficiency owing to the smaller sample size both in the model fitting and calibration phases.

Aggregate split-CP: To overcome the drawbacks of split-CP, cross-conformal prediction was introduced in Vovk (2015) and Vovk et al. (2018) where the dataset is partitioned into K folds and one performs a split conformal prediction by sequentially defining the k^{th} fold as the calibration set and the remaining as the training set for $k \in [K]$. The main difficulty lies in aggregating the different p-values while maintaining the validity of the method (see Carlsson et al. (2014); Linusson et al. (2017)).

Other works in this direction are Bonferroni-type aggregation (Lei & Wasserman, 2014; Solari & Djordjilović, 2022), Cauchy-type aggregation (Wu et al., 2023), out-of-bag ensemble (Gupta et al., 2022) and resampling techniques (Dunn et al., 2022). However, all these methods either depend on strong stability assumption of the underlying model or overly conservative. In general, it can be shown that the coverage level is inflated by a factor of 2, that is, theoretical coverage level is $1-2\alpha$ instead of $1-\alpha$, which is not improvable (see Barber et al. (2021)). Nevertheless, the practical performance is acceptable both computationally and statistically.

However, in this paper we are only interested in methods that provides a coverage level of $1-\alpha$. Hence, in our experiments we mainly compared our proposed methods with traditional split-CP that provides a coverage guarantee of $1-\alpha$. However, for the sake of completeness we also provided additional results in the supporting information (B. Additional Results and Experimental Details) where we compared our method with aggregate split-CP methods such as jackknife and jackknjife+.

3 | PROPOSED METHOD

We propose a *homotopy-mining* method to compute the exact full-CP set of a SHIM. The homotopy method refers to an optimization framework for solving a sequence of parameterized optimization problems. The basic idea of our method is to consider the following optimization problem:

$$\beta(\tau) \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y(\tau) - X\beta\|_2^2 + \lambda\|\beta\|_1. \tag{4}$$

By the optimality conditions, at optima, it holds

$$X^\top (X\beta(\tau) - y(\tau)) + \lambda s(\tau) = 0,$$

where for any ℓ in $[p]$,

$$s_\ell(\tau) \in \begin{cases} \{-1, +1\} & \text{if} \beta_\ell(\tau) \neq 0, \\ [-1, +1] & \text{if} \beta_\ell(\tau) = 0. \end{cases}$$

Let us define the active set

$$\mathcal{A}_\tau = \left\{\ell \in [p] : \ |x_\ell^\top w(\tau)| = \lambda\right\}, \tag{5}$$

where

$$w(\tau) = y(\tau) - X\beta(\tau) \tag{6}$$

is the residual (Efron et al., 2004). Lei (2019) showed that for a fixed λ, the exact solution path of LASSO with a variable response $y(\tau)$, characterized by τ can be shown to be piece-wise linear as stated in Proposition 3.1.

Proposition 3.1. If $\beta(\tau)$ have the same sign between two points τ_1 and τ_2, that is, $\text{sign}(\beta(\tau_1)) = \text{sign}(\beta(\tau_2)) = \text{sign}(\beta(\tau))$ for any $\tau \in [\tau_1, \tau_2)$, then $\mathcal{A}_\tau = \mathcal{A}_{\tau_1}$. Furthermore, assuming that $X_{\mathcal{A}_\tau}^\top X_{\mathcal{A}_\tau}$ is invertible, we have the linear relations

$$\begin{aligned} \beta_{\mathcal{A}_\tau}(\tau_2) &= \beta_{\mathcal{A}_\tau}(\tau_1) + (\tau_2 - \tau_1) \times \nu_{\mathcal{A}_\tau}(\tau), \\ s_{\mathcal{A}_\tau^c}(\tau_2) &= s_{\mathcal{A}_\tau^c}(\tau_1) + (\tau_2 - \tau_1) \times \gamma_{\mathcal{A}_\tau^c}(\tau)/\lambda, \end{aligned}$$

where the direction vectors are defined as

$$\nu_{\mathcal{A}_\tau}(\tau) = \left(X_{\mathcal{A}_\tau}^\top X_{\mathcal{A}_\tau}\right)^{-1} x_{n+1,\mathcal{A}_\tau},$$
$$\gamma_{\mathcal{A}_\tau^c}(\tau) = x_{n+1,\mathcal{A}_\tau^c} - \left(X_{\mathcal{A}_\tau^c}^\top X_{\mathcal{A}_\tau}\right)\nu_{\mathcal{A}_\tau}(\tau).$$

For simplicity, we will denote the step size $\Delta = \tau_2 - \tau_1 > 0$.

We call the mapping $\tau \mapsto \beta(\tau)$ the τ-path and the number of linear pieces of this τ-path is upper bounded by the number of all possible signs 3^p. Note that the possible values of sign(β) are $-1, 0$ and $+1$. This τ-path can be computed exactly using homotopy method (Efron et al., 2004; Mairal & Yu, 2012; Rosset & Zhu, 2007) that sequencially tracks and updates the sign and active set of the optimal solution by exploiting its linearity between each two consecutive transition points of direction $(\nu_{\mathcal{A}_\tau}(\tau), \gamma_{\mathcal{A}_\tau^c}(\tau))$ changes. At every consecutive step, represented by τ_t and τ_{t+1}, where t is an index of the transition points (kinks) of the τ-path, either of the following two events occurs:

- a zero variable becomes non-zero, that is,

$$\exists \ell \in \mathcal{A}_{\tau_t}^c \text{ s.t. } \left|x_\ell^\top w(\tau_{t+1})\right| = \lambda \text{ or,}$$

- a non-zero variable becomes zero, that is,

$$\exists \ell \in \mathcal{A}_{\tau_t} \text{ s.t. } \beta_\ell(\tau_t) \neq 0, \text{ but } \beta_\ell(\tau_{t+1}) = 0.$$

Overall, the next change in the active set (or change in direction vectors) occurs at $\tau_{t+1} = \tau_t + \Delta_{\ell^*}$, such that

$$\Delta_{\ell^*} = \min(\Delta_1(\ell_1^*), \Delta_2(\ell_2^*)), \tag{7}$$

where

$$\ell_1^* = \arg\min_{\ell \in \mathcal{A}_{\tau_t}} \Delta_1(\ell),$$
$$\ell_2^* = \arg\min_{\ell \in \mathcal{A}_{\tau_t}^c} \Delta_2(\ell), \text{and}$$

$$\Delta_1(\ell) = \left(-\frac{\beta_\ell(\tau_t)}{\nu_\ell(\tau_t)}\right)_{++},$$
$$\Delta_2(\ell) = \left(\lambda \frac{\text{sign}(\gamma_\ell(\tau_t)) - x_\ell^\top w(\tau_t)}{\gamma_\ell(\tau_t)}\right)_{++}.$$

Here, we use the convention that for any $a \in \mathbb{R}, (a)_{++} = a$ if $a > 0$ and ∞ otherwise. However, naively (by simply minimizing over all possible interaction terms) determining the step size of inclusion $(\Delta_2(\ell_2^*))$ will be intractable for the SHIM type problem. In SHIM, the search space grows exponentially due to the combinatorial effect of high-order interaction terms. Therefore, both the fitting and constructing the full-CP set of a SHIM are non-trivial because unless both m and d are very small, a high-order interaction model will have a significantly large number of parameters to be considered. Several algorithms for fitting a sparse high-order interaction model have been proposed in the literature (Nakagawa et al., 2016; Saigo et al., 2009; Tsuda, 2007). A common approach adopted in these existing works is to exploit the hierarchical structure of high-order interaction features.

In other words, a tree structure as in Figure 1b is considered and a branch-and-bound strategy is employed in order to avoid handling all the exponentially increasing number of high-order interaction features. Hence, we need efficient computational methods to make the computation practically feasible.

In the following section, we present an efficient tree pruning strategy that considers the tree structure of the interaction terms (or patterns). Here, each node of the tree represents an interaction term. The basic idea of tree pruning is that we construct a tree of interaction terms in a 'progressive manner'. That is, we keep track of the current minimum step size of inclusion up to the construction of ℓ^{th} pattern as we construct the tree progressively, and prune a large part of the tree if some bound condition fails (Lemma 3.2).

3.1 | Tree pruning (τ-path)

Definition of tree: A tree is constructed in such a way that for any pair of nodes (ℓ,ℓ'), where ℓ is the ancestor of ℓ', that is, $\ell \subset \ell'$, the following conditions are satisfied $\forall i \in [n+1]$:

$$x_{i\ell'} = 1 \Rightarrow x_{i\ell} = 1, \text{and} x_{i\ell} = 0 \Rightarrow x_{i\ell'} = 0.$$

The basic idea of our tree pruning condition is stated below. The equicorrelation condition for any active feature $k \in \mathcal{A}_\tau$ at a fixed λ can be written as $\left|x_k^\top w(\tau)\right| = \lambda$ (see (5)). Therefore, at $\tau = \tau_{t+1}$ such that $\tau_{t+1} = \tau_t + \Delta_2(\ell)$, any non-active feature $\ell \in \mathcal{A}^c_{\tau_t}$ becomes active if

$$\left|x_\ell^\top w(\tau_{t+1})\right| = \left|x_k^\top w(\tau_{t+1})\right|. \tag{8}$$

Now, using the triangular inequality, one can show that (8) will not have any solution if

$$|\rho_\ell(\tau_t)| + \Delta_2(\ell)(|\eta_\ell(\tau_t)| + x_{n+1,\ell}) < |\rho_k(\tau_t)| - \Delta_2(\ell)(|\eta_k(\tau_t)| + x_{n+1,k}), \tag{9}$$

where

$$\begin{aligned} \rho_\ell &= x_\ell^\top w(\tau_t) \text{and} \eta_\ell = x_\ell^\top v(\tau_t)\ \forall \ell \in \mathcal{A}^c_{\tau_t}, \\ \rho_k &= x_k^\top w(\tau_t) \text{and} \eta_k = x_k^\top v(\tau_t)\ \forall k \in \mathcal{A}_{\tau_t}, \\ v(\tau_t) &= X_{\mathcal{A}_{\tau_t}} \nu_{\mathcal{A}_{\tau_t}}(\tau_t). \end{aligned}$$

Therefore, (9) can be used to derive the pruning condition of the τ-path which is formally stated in Lemma 3.2.

Similar idea has been used in the context of graph mining Tsuda (2007) and selective inference of SHIM Das et al. (2021). Note that Tsuda (2007) provided a pruning condition for the exact regularization of graph data (λ-path) and Das et al. (2021) provided a pruning condition in the context of selective inference to characterize the conditional distribution of the test statistics. However, in our case, we adapted the similar idea to compute the exact full-CP set of SHIM.

Lemma 3.2. For any given node ℓ, if $\Delta_2(\ell_2^\dagger)$ is the current minimum step size, that is,

$$\ell_2^\dagger = \arg \min_{j \in \{1,2,\ldots,\ell\} \cap \mathcal{A}^c_\tau} \Delta_2(j),$$

then $\forall \ell' \supset \ell, \Delta_2(\ell') \geq \Delta_2(\ell_2^\dagger)$ if

$$b_\ell(w(\tau_t)) + \Delta_2(\ell_2^\dagger)(b_\ell(v(\tau_t)) + x_{n+1,\ell}) < |\rho_k(\tau_t)| - \Delta_2(\ell_2^\dagger)(|\eta_k(\tau_t)| + x_{n+1,k}), \tag{10}$$

where we defined for a vector $a \in \mathbb{R}^{n+1}$

$$b_\ell(a) := \max\left\{\sum_{a_i>0} |a_i| x_{i\ell}, \sum_{a_i<0} |a_i| x_{i\ell}\right\}.$$

The Lemma 3.2 essentially states that if the condition in (10) is satisfied, then one can safely ignore the subtree with ℓ as the root node, thereby dramatically improving the computational efficiency.

We extended our method to elastic net and call it ENet-SHIM. The details of ENet-SHIM, proof of Lemma 3.2, and the algorithms to compute the exact full-CP of SHIM are provided in the supporting information (A. Additional Technical Details).

4 | RESULTS AND DISCUSSIONS

We evaluated our proposed method using both synthetic and real-world data. For all experiments, we considered a coverage guarantee of 90%, that is, $\alpha = 0.1$. We compared the statistical efficiency of our proposed method (SHIM) with those of other simple (LASSO) and complex models

(NN, RF). For the complex models, we reported the split-CP because it is the only available method for computing the CP for complex models. We also demonstrated the statistical efficiency of full-CP in comparison to that of split-CP both for LASSO and SHIM. We used shorthand notations _s and _f to represent split-CP and full-CP respectively and the number before s/f represents the maximum order of interactions considered. For example in Table 1, shim_2s and shim_2f respectively represent split-CP and full-CP of a 2^{nd}-order SHIM. We used a multi-layer perceptron (MLP) as a neural network architecture in our experiments.

4.1 | Comparison of statistical efficiencies

4.1.1 | Synthetic data experiments

We generated random i.i.d. samples $(Z_i, y_i) \in \{0,1\}^m \times \mathbb{R}, i \in [n]$ in such a way that $100m(1-\zeta)\%$ features of $Z_i \in \mathbb{R}^m$ contain a value of 1 on average. Here, $\zeta \in [0,1]$ is a parameter that controls the sparsity of the design matrix, whereas the sparsity in the model coefficients are controlled by the regularizer λ. The effectiveness of the pruning (Figure 5 and Table 3) depends on the sparsity of the design matrix as it exploits the tree's anti-monotonicity property. High dimensional real-world data is generally very sparse and the choice of ζ in our experiments is just for the demonstration purpose.

The response $y_i \in \mathbb{R}$ is randomly generated from a normal distribution $\mathcal{N}(2\mu(Z_i), \sigma^2)$. For demonstration purposes, we considered a true model of up to fifth-order interactions, which is defined as $\mu(Z_i) = z_{i1} + z_{i1}z_{i2} + z_{i1}z_{i2}z_{i3} + z_{i1}z_{i3}z_{i4}z_{i5} + z_{i1}z_{i2}z_{i3}z_{i4}z_{i5}$, and set $\sigma = 1$. The choice of this model is merely for the demonstration purposes, and the proposed method is equally applicable to any chosen model. We used the same true model $\mu(Z_i)$ in both low and high dimensional settings (Tables 1 and 2 and Figure 2).

In the low-dimensional setting, we considered a training dataset of 150 instances and $m = 10$ original covariates, whereas for the high-dimensional setting, we considered a training dataset of 150 instances and $m = 100$. We kept the sparsity of the design matrix fixed at $\zeta = 0.4$ in the both settings. We varied the order of interactions $d = 2,3,\ldots$; however, in almost all the experiments, we found that the model get saturated after 3rd order interactions (i.e., the performance of SHIM did not change much for $d \geq 3$). Hence, we reported the results of up to 3rd order interactions. Note that the maximum order of interactions d is not needed to be specified beforehand. Our pruning condition takes care of the case even when d is not specified that is when the whole search space is considered (Figure 5 and Table 3). Later, we provided results (Figures 3 and 4) where we did not impose any constraints on the maximum pattern (max-pat) size d. For the both settings (low and high), we generated a test dataset of 50 instances. We generated 5 such independent random datasets and repeated the experiments 3 times. Hence, in total we reported the average results of 15 independent datasets, that is, $3 \times 5 \times 50 = 750$ test instances were considered. The details of the hyper parameter selection are given in the supporting information (B. Additional Results and Experimental Details).

From Tables 1 and 2 it can be observed that both in low- and high-dimensional settings, all the methods produced perfect (or nearly perfect) coverage$= 0.90$. The statistical efficiency of individual methods are compared using the length of prediction interval. A statistically efficient model is expected to produce a shorter prediction interval length. Comparing the length of prediction intervals, one can observe that the statistical efficiency of SHIM increases as we increase the order of interactions. A 3rd order SHIM produced the shortest average prediction interval lengths both in low- and high-dimensional settings.

To compare the fitting power of individual methods we reported the R-squared (r2) scores. It can be observed that the r2 scores of a 3rd order SHIM are comparable to the best performing complex model (RF) both in low- and high-dimensional settings. We also demonstrated the comparison of the prediction interval lengths (CI length) and R-squared (r2) scores of the proposed method (SHIM) with other simple (LASSO) and complex models (MLP, RF) for three different sample sizes, $n \in \{100,150,200\}, m = 10$ (Figure 2). It can be observed that the full-CP of 3rd order SHIM (shim3_f) produced the best results (shortest avg. length, highest avg. r2 score) in all the cases. It can also be observed that the performance of data splitting (in mlp, rf, lasso_s, shim3_s) is worse than that of full-CP. The split-CP tends to produce a longer prediction interval and smaller r2 score, and suffer from high variance due to the smaller data size as well as the additional randomness considered in data splitting. Additional results using highly sparse synthetic data generated from models with weak and strong signals are provided in the supporting information (B. Additional Results and Experimental Details).

TABLE 1 Comparison of the statistical power using low dimensional synthetic data ($m = 10, n = 150$).

	mlp	rf	lasso_s	lasso_f	shim_2s	shim_2f	shim_3s	shim_3f
length	1.94 (0.27)	1.84 (0.23)	2.32 (0.23)	2.18 (0.10)	2.00 (0.23)	1.82 (0.17)	1.96 (0.24)	1.76 (0.25)
cov	0.90 (0.05)	0.89 (0.06)	0.91 (0.04)	0.90 (0.05)	0.91 (0.04)	0.89 (0.05)	0.90 (0.04)	0.88 (0.04)
r2	0.61 (0.18)	0.66 (0.09)	0.51 (0.12)	0.54 (0.11)	0.63 (0.10)	0.67 (0.10)	0.63 (0.09)	0.67 (0.09)

Note: The bracketed values represent the standard deviations.

TABLE 2 Comparison of the statistical power using high dimensional synthetic data ($m = 100, n = 150$).

	mlp	rf	lasso_s	lasso_f	shim_2s	shim_2f	shim_3s	shim_3f
length	3.39 (0.51)	1.93 (0.32)	2.45 (0.40)	2.26 (0.19)	2.20 (0.36)	1.92 (0.26)	2.27 (0.38)	1.89 (0.30)
cov	0.91 (0.05)	0.90 (0.06)	0.90 (0.06)	0.92 (0.05)	0.90 (0.05)	0.91 (0.04)	0.90 (0.06)	0.90 (0.04)
r2	-0.01 (0.16)	0.66 (0.11)	0.45 (0.09)	0.53 (0.07)	0.54 (0.13)	0.65 (0.09)	0.52 (0.12)	0.66 (0.08)

Note: The bracketed values represent the standard deviations.

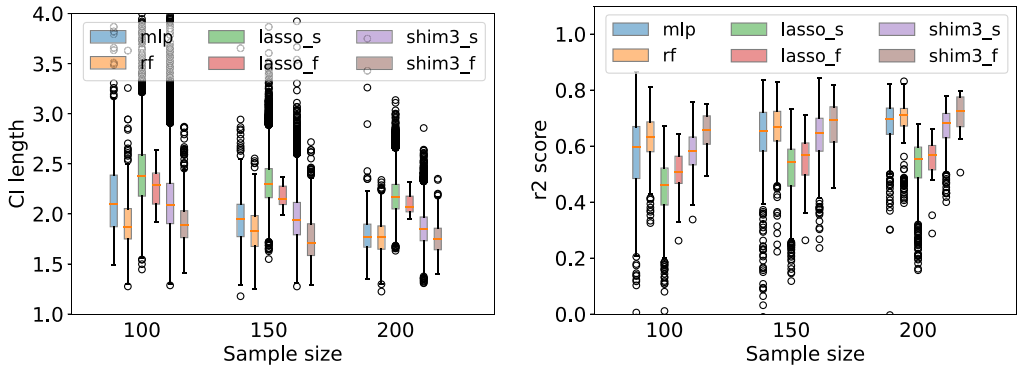


FIGURE 2 Comparison of the prediction interval lengths (CI length) and r2 scores of the proposed method (shim) with other simple (lasso) and complex models (mlp, rf) using synthetic data for different sample sizes.

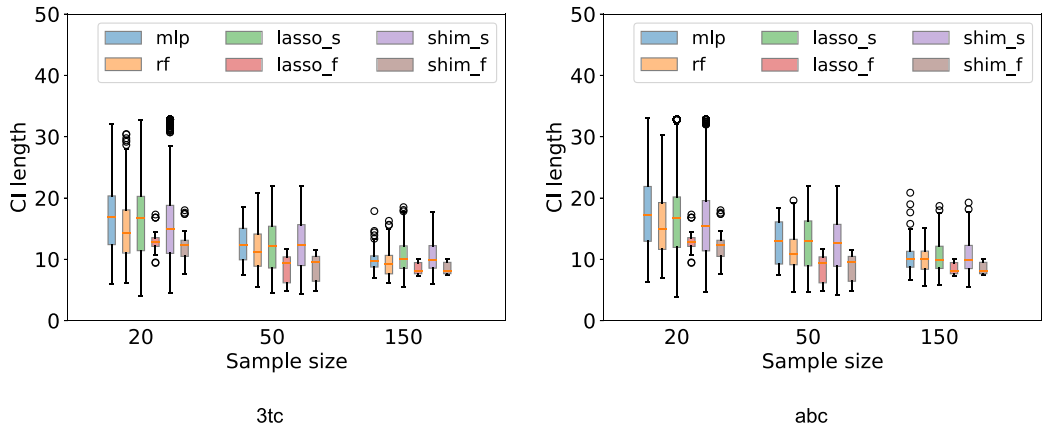


FIGURE 3 Comparison of the prediction interval lengths (CI length) using two hiv drug resistance data ('3tc' and 'abc') for different sample sizes. The comparison of r2-scores are provided in the supporting information (B. Additional Results and Experimental Details)

4.1.2 | Real-world data experiments

HIV Data. We applied our method on real world HIV drug resistance dataset. The HIV-1 sequence data was obtained from the Stanford HIV Drug Resistance Database (Rhee et al., (2003), (2006)). We applied our method on two NRTI drugs, Lamivudine (3TC) and Abacavir (ABC). The results are shown in Figure 3. Here, we considered top ten mutations. In the HIV data set most of the columns contain zeros, hence, we sorted them based on the number of 1s present in each column and selected the top ten columns. We subsampled the data to generate independent training sets for three different training set sizes, $n \in \{20,50,150\}$, each accompanied with a separate test set of 10 instances. We repeated the experiment three times and in total we reported the results of $3 \times 10 = 30$ test instances.

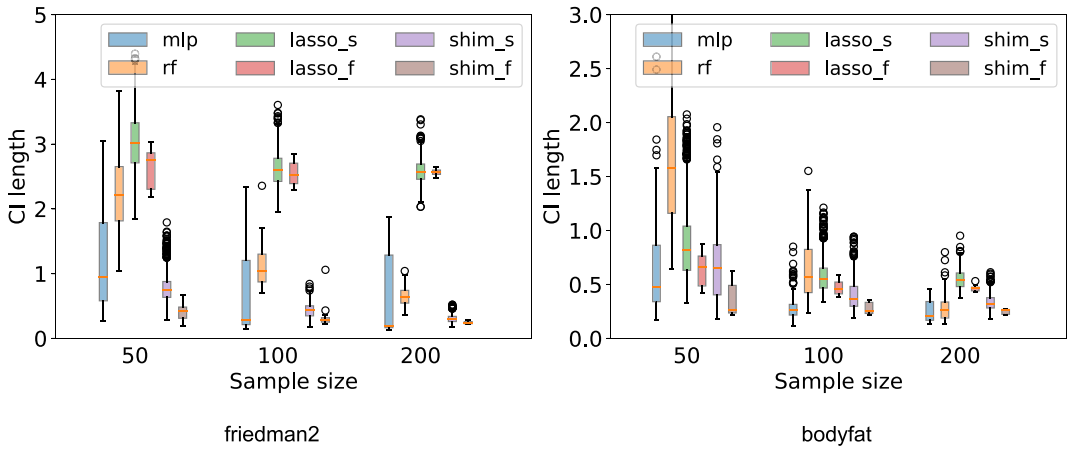


FIGURE 4 Comparison of the prediction interval lengths (CI length) using continuous synthetic (friedman2) and real world (bodyfat) data. The comparison of r2-scores and additional results using large $n = 500$ for friedman2 data are provided in the supporting information (B. Additional Results and Experimental Details)

4.1.3 | Experiments with continuous data

To demonstrate the efficacy of our method using continuous data we considered both synthetic and real world continuous data. As a synthetic dataset, we considered the standard Friedman2 dataset generated by the scikit-learn package. These data contain four independent features randomly distributed on specific intervals (for details, please see scikit-learn package). As a real dataset, we considered the bodyfat dataset obtained from the LIBSVM repository for regression data analysis. This dataset consists of 252 instances and 14 numerical features (for details please see LIBSVM repository). For both synthetic and real world datasets, we rescaled all the features values in the range of 0 and 1 and conducted all the experiments with rescaled features values. As before, we subsampled the data to generate independent training sets for three different training sizes, $n \in \{50,100,200\}$, each accompanied with a separate test set of 10 instances. We repeated the experiment three times and in total we reported the results of $3 \times 10 = 30$ test instances. The results are shown in Figure 4.

4.2 | Comparison of computational efficiencies

To demonstrate the computational efficiency of the proposed pruning strategy for the τ-path, we generated a synthetic dataset of $n = 100$ and $m = 30$ for three different sparsity levels of the design matrix ($\zeta = 0.4,0.7,0.9$) using the same 5th order model as used to demonstrate the statistical efficiency in Tables 1 and 2 and Figure 2. We compared both the fraction of nodes traversed (Figure 5) and the time taken (Table 3) against a different maximum interaction order d for three different sparsity levels ($\zeta = 0.4,0.7,0.9$) using two different λ values ($\lambda = 1,10$). It can be observed that the pruning is more effective at the deeper nodes of the tree and saturates after a certain depth of the tree. This is evident as the sparsity of the data increases at the deeper nodes, and the pruning exploits the anti-monotonicity of high-order interaction terms constructed as tree of patterns. In the case of the homotopy method without pruning, we stopped the execution of the program if the τ-path was not finished in one day. From Table 3, it can be observed that without the tree pruning, the construction of the τ-path is not practical as we progress to the deeper nodes of the tree because of the generation of an exponential number of high-order interaction terms. The maximum time taken by the τ-path with pruning was approximately 130s, for a 'max-pat'(d) size of 25, that is, for 1,073,709,892 nodes at $\lambda = 1, \zeta = 0.4$. Figure 5 shows the variation of node counts ('fraction of node counts') for different *max_pat* (d) size during the construction of τ-path. One can observe that our pruning condition is more effective at the deeper nodes of the tree as it exploits the tree's anti-monotonicity property. Although the worst-case complexity of the τ-path is exponential (considering all possible sign values of model coefficients, see discussion in Section 3 for details), it has been well-recognized that this worst-case rarely happens in practice (Li & Singer, 2018; Le Duy & Takeuchi, 2021). This is also evident from our experimental results in Table 4 of the supporting information (B. Additional Results and Experimental Details).

TABLE 3 Computation time (in sec) with and without pruning using two different λ values for three different sparsity levels (ζ).

		$\lambda = 1$						$\lambda = 10$					
		With pruning			Without pruning			With pruning			Without pruning		
d	Search space (# nodes)	$\zeta = 0.4$	$\zeta = 0.7$	$\zeta = 0.9$	$\zeta = 0.4$	$\zeta = 0.7$	$\zeta = 0.9$	$\zeta = 0.4$	$\zeta = 0.7$	$\zeta = 0.9$	$\zeta = 0.4$	$\zeta = 0.7$	$\zeta = 0.9$
2	465	0.09	0.10	0.03	0.10	0.08	0.07	0.08	0.08	0.03	0.07	0.07	0.12
3	4525	0.84	0.67	0.03	0.82	0.79	0.64	0.56	0.36	0.03	0.63	0.65	0.85
4	31930	4.49	1.51	0.03	5.91	5.32	4.73	2.14	0.90	0.03	4.12	4.42	3.87
5	174436	12.34	2.74	0.03	30.19	28.76	23.31	5.11	1.38	0.03	26.74	22.70	25.42
10	53009101	112.17	3.83	0.03	> 1 day	> 1 day	> 1 day	49.88	2.13	0.03	> 1 day	6891.44	6861.16
15	614429671	126.04	3.39	0.03	> 1 day	> 1 day	> 1 day	56.22	2.00	0.03	> 1 day	> 1 day	> 1 day
20	1050777736	126.86	3.51	0.03	> 1 day	> 1 day	> 1 day	55.33	2.13	0.03	> 1 day	> 1 day	> 1 day
25	1073709892	130.28	3.62	0.03	> 1 day	> 1 day	> 1 day	55.21	2.02	0.03	> 1 day	> 1 day	> 1 day

Note: All computation times were measured on an Intel(R) Xeon(R) Gold 6130 CPU @ 2.10 GHz.

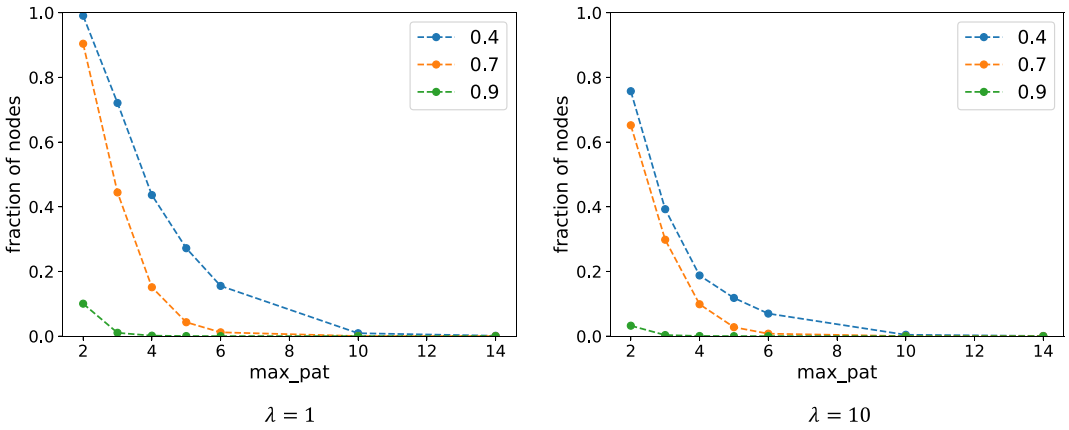


FIGURE 5 Variation of node counts ('fraction of node counts') for different *max_pat* (d) sizes. Here, the fraction of nodes for a specific d, represents the number of nodes traversed divided by the total number of possible combinations of interaction terms using the 'max-pat' size of d. The results are shown for two λ values ($\lambda = 1,10$) and for three different sparsity levels (0.4, 0.7, 0.9). The pruning is more effective as the data is more sparse.

5 | CONCLUSION

In this paper we proposed an algorithm to efficiently compute the full-CP set of SHIM. Through numerical experiments, we demonstrated that SHIM is statistically superior than the vanilla lasso and comparable to other complex models (NN, RF). The prediction interval along with a point estimation is better than a point estimation alone. The computation of a full-CP set for other complex models (NN, RF) remains an open question. SHIM is interpretable and it generates the decision function as a weighted combination of *decision sets* (if-then rules). The homotopy-mining based exact full-CP of SHIM is better than the split-CP of SHIM. The efficient pruning strategy makes the computation of exact full-CP set tractable for SHIM.

AUTHOR CONTRIBUTIONS

Diptesh Das introduced and formalized the contributions of the paper. He developed the theory, its necessary proofs and implemented the algorithms. He wrote and analyzed the technical details, designed and performed the numerical experiments, and prepared figures and/or tables. Eugene Ndiaye guided in conceptualization and formulation in this study. He checked and analyzed the the technical details, wrote some parts of technical details, and helped in design of the numerical experiments in this study. Ichiro Takeuchi contributed to the conceptualization, formulation, and design of the numerical experiments in this study. He analyzed the technical details and data and guided in preparation of figures and/or tables.

ORCID

Diptesh Das https://orcid.org/0000-0002-6736-9596

REFERENCES

Alquier, P. (2021). User-friendly introduction to pac-bayes bounds. arXiv preprint arXiv:2110.11216.

Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., & Rudin, C. (2018). Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, *18*, 1–78.

Angelopoulos, A. N., Bates, S., Jordan, M. I., & Malik, J. (2020). Uncertainty Sets for Image Classifiers using Conformal Prediction. In *International Conference on Learning Representations*.

Barber, R. F., Candes, E. J., Ramdas, A., & Tibshirani, R. J. (2021). Predictive inference with the jackknife+. The Annals of Statistics.

Barber, R. F., Candes, E. J., Ramdas, A., & Tibshirani, R. J. (2023). Conformal prediction beyond exchangeability. *The Annals of Statistics*, *51*(2), 816–845.

Bates, S., Candès, E., Lei, L., Romano, Y., & Sesia, M. (2023). Testing for outliers with conformal p-values. *The Annals of Statistics*, *51*(1), 149–178.

Carlsson, L., Eklund, M., & Norinder, U. (2014). Aggregated conformal prediction. IFIP International Conference on Artificial Intelligence Applications and Innovations.

Cherubin, G., Chatzikokolakis, K., & Jaggi, M. (2021). Exact optimization of conformal predictors via incremental and decremental learning. In *International Conference on Machine Learning*, pp. 1836–1845. PMLR.

Clyde, M., Çetinkaya-Rundel, M., Rundel, C., Banks, D., Chai, C., & Huang, L. (2021). An introduction to bayesian thinking-a companion to the statistics with r course. GitHub repository: GitHub.

Das, D., Le Duy, V. N., 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*, *36*, pp. 9999–10007.

Das, D., Ito, J., Kadowaki, T., & Tsuda, K. (2019). An interpretable machine learning model for diagnosis of alzheimer's disease. *PeerJ*, *7*, e6543.

Dey, N., Ding, J., Ferrell, J., Kapper, C., Lovig, M., Planchon, E., & Williams, J. P. (2022). Conformal prediction for text infilling and part-of-speech prediction. *The New England Journal of Statistics in Data Science*, *1*(1), 69–83.

Dunn, R., Wasserman, L., & Ramdas, A. (2022). Distribution-free prediction sets for two-layer hierarchical models. *Journal of the American Statistical Association*, 1–12.

Duy, V. N. L., Toda, H., Sugiyama, R., & Takeuchi, I. (2020). Computing valid p-value for optimal changepoint by selective inference using dynamic programming. *Advances in Neural Information Processing Systems*, *33*, 11356–11367.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, *32*(2), 407–499.

Fisch, A., Schuster, T., Jaakkola, T., & Barzilay, R. (2021). In *International Conference on Machine Learning*, PMLR, pp. 3329–3339.

Fithian, W., Sun, D., & Taylor, J. (2014). Optimal inference after model selection. arXiv preprint arXiv:1410.2597.

Gibbs, I., Cherian, J. J., & Candès, E. J. (2023). Conformal prediction with conditional guarantees. arXiv preprint arXiv:2305.12616.

Gupta, C., Kuchibhotla, A. K., & Ramdas, A. (2022). Nested conformal prediction and quantile out-of-bag ensemble methods. *Pattern Recognition*, *127*, 108496.

Ho, S.-S., & Wechsler, H. (2008). Query by transduction. In *IEEE transactions on pattern analysis and machine intelligence*, *30*, pp. 1557–1571.

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable decision sets A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1675–1684.

Larson, J., Mattu, S., Kirchner, L., & Angwin, J. (2016). How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, *9*(1), 3.

Laxhammar, R., & Falkman, G. (2015). Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*, *74*(1), 67–94.

Le Duy, V. N., & Takeuchi, I. (2021). Parametric programming approach for more powerful and general lasso selective inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 901–909. PMLR.

Lee, J. D., Sun, D. L., Sun, Y., & Taylor, J. E. (2016). Exact post-selection inference, with application to the lasso. *Annals of Statistics*, *44*(3), 907–927.

Lei, J. (2014). Classification with confidence. *Biometrika*, *101*(4), 755–769.

Lei, J. (2019). Fast exact conforMalization of the lasso using piecewise linear homotopy. *Biometrika*, *106*(4), 749–764.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., & Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, *113*(523), 1094–1111.

Lei, J., Robins, J., & Wasserman, L. (2013). Distribution-free prediction sets. *Journal of the American Statistical Association*, *108*(501), 278–287.

Lei, J., & Wasserman, L. (2014). Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, *76*, 71–96.

Lei, L., & Candès, E. J. (2021). Conformal inference of counterfactuals and individual treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *83*(5), 911–938.

Li, Y., & Singer, Y. (2018). The well-tempered lasso. In *International Conference on Machine Learning*, pp. 3024–3032. PMLR.

Linusson, H., Norinder, U., Boström, H., Johansson, U., & Löfström, T. (2017). On the calibration of aggregated conformal predictors. In *Conformal and probabilistic prediction and applications*.

Mairal, J., & Yu, B. (2012). Complexity analysis of the lasso regularization path. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1835–1842.

Martinez, J. A., Bhatt, U., Weller, A., & Cherubin, G. (2023). Approximating full conformal prediction at scale via influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*, pp. 6631–6639.

Nakagawa, K., Suzumura, S., Karasuyama, M., Tsuda, K., & Takeuchi, I. (2016). Safe pattern pruning An efficient approach for predictive pattern mining. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 1785–1794.

Ndiaye, E., & Takeuchi, I. (2019). Computing full conformal prediction set with approximate homotopy. *Advances in Neural Information Processing Systems*, *32*.

Ndiaye, E., & Takeuchi, I. (2023). Root-finding approaches for computing conformal prediction set. *Machine Learning*, *112*(1), 151–176.

Nouretdinov, I., Melluish, T., & Vovk, V. (2001). Ridge regression confidence machine. In *ICML*, pp. 385–392. Citeseer.

Papadopoulos, H., Proedrou, K., Vovk, V., & Gammerman, A. (2002). Inductive confidence machines for regression, *European Conference on Machine Learning*: Springer, pp. 345–356.

Rhee, S.-Y., Gonzales, M. J., Kantor, R., Betts, B. J., Ravela, J., & Shafer, R. W. (2003). Human immunodeficiency virus reverse transcriptase and protease sequence database. *Nucleic acids research*, *31*(1), 298–303.

Rhee, S.-Y., Taylor, J., Wadhera, G., Ben-Hur, A., Brutlag, D. L., & Shafer, R. W. (2006). Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proceedings of the National Academy of Sciences*, *103*(46), 17355–17360.

Rosset, S., & Zhu, J. (2007). Piecewise linear regularized solution paths. *The Annals of Statistics*, *35*, 1012–1030.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

Sadinle, M., Lei, J., & Wasserman, L. (2019). Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, *114*(525), 223–234.

Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., & Tsuda, K. (2009). gboost: a mathematical programming approach to graph classification and regression. *Machine Learning*, *75*(1), 69–89.

Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction (pp. 371–421).

Shawe-Taylor, J., & Williamson, R. C. (1997). A pac analysis of a bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pp. 2–9.

Solari, A., & Djordjilović, V. (2022). Multi split conformal prediction. *Statistics & Probability Letters*, *184*, 109395.

Tsuda, K. (2007). Entire regularization paths for graph data. In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, New York, NY, USA, pp. 919–926. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273612.

Vovk, V. (2015). Cross-conformal predictors. Annals of Mathematics and Artificial Intelligence.

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world*: Springer Science & Business Media.

Vovk, V., Nouretdinov, I., Manokhin, V., & Gammerman, A. (2018). Cross-conformal predictive distributions. In *Conformal and Probabilistic Prediction and Applications*, pp. 37–51. PMLR.

Wu, X., Huo, Y., & Zou, C. (2023). Multi-split conformal prediction via cauchy aggregation. *Stat*, *12*(1), e522.

Xu, C., & Xie, Y. (2021). Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pp. 11559–11569. PMLR.

AUTHOR BIOGRAPHIES

Diptesh Das is a researcher at the Department of Computational Biology and Medical Sciences of the University of Tokyo. His research interest focuses on statistical machine learning and its application in high-stake decision-making problems. Prior to his current position, he served as a post-doctoral researcher at the University of Tokyo, a project assistant professor at Nagoya Institute of Technology, and subsequently a researcher at Nagoya University, Japan. His PhD (University of Tokyo) thesis focuses on interpretable machine learning models for medical data. Besides his expertise in academics, he has vast work experience in the industry as well. After earning a BSc (Physics Honors) and a BTech (Instrumentation Engineering) degrees from the University of Calcutta, India, he started his professional career at the Innovation Lab, Kolkata, TATA Consultancy Services Ltd. (TCS), where he worked for several years. He then moved to the UK to pursue an MSc (Advanced Computing) from the University of Bristol, UK, where he started a new journey in academia.

Eugene Ndiaye is a research scientist working on uncertainty quantification and optimization. Previously, he was a Tennenbaum President's Postdoctoral Fellow in the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Institute of Technology and was also a Postdoctoral Researcher at RIKEN Center for Advanced Intelligence Project in the Data-Driven Biomedical Science Team. He holds a PhD in Applied Mathematics with a doctoral thesis focused on the design and analysis of faster and safer optimization algorithms for variable selection and hyperparameter calibration in high dimension at Télécom ParisTech, EDMH, Université Paris-Saclay.

Ichiro Takeuchi is a professor at Nagoya University in Japan and leads a team at the RIKEN Center for Advanced Intelligence Project. He earned his BEng, MEng, and DEng degrees from Nagoya University in 1996, 1998, and 2000, respectively. After serving as a post-doctoral researcher under Prof. Yoshua Bengio in Montreal, Canada, he became a tenured assistant professor at Mie University in 2001. He later achieved associate and full professor positions at Nagoya Institute of Technology in 2008 and 2015, respectively, and finally a full professor position at Nagoya University in 2022. His research focuses on machine learning theory and algorithms and their applications in bio-medical and material sciences.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Das, D., Ndiaye, E., & Takeuchi, I. (2024). A confidence machine for sparse high-order interaction model. *Stat*, *13*(1), e633. https://doi.org/10.1002/sta4.633

Supporting Information: Additional information for this article is available.

A. Additional Technical Details

A.1. Proof of Lemma 3.2

Before proving the Lemma 3.2, we introduce the following two propositions:

Proposition A.1. *Let's define for a vector* $a \in \mathbb{R}^{n+1}$

$$b_\ell(a) = \max\left\{\sum_{a_i<0}|a_i|x_{i\ell}, \sum_{a_i>0}|a_i|x_{i\ell}\right\},$$

then if we expand $\rho_\ell(\tau_t)$ *and* $\eta_\ell(\tau_t)$ *separately for positive and negative values of* $w_i(\tau_t)$ *and* $v_i(\tau_t), \forall i \in [n+1]$ *respectively, we can write*

$$|\rho_\ell(\tau_t)| \le b_\ell(w(\tau_t)),$$
$$|\eta_\ell(\tau_t)| \le b_\ell(v(\tau_t)),$$

Proof of Proposition A.1. We have

$$\begin{aligned}
|x_\ell^\top a| &= \left|\sum_{i=1}^{n+1} a_i x_{i\ell}\right| \\
&= \left|\sum_{a_i>0}|a_i|x_{i\ell} - \sum_{a_i<0}|a_i|x_{i\ell}\right| \\
&\le \max\left\{\sum_{a_i>0}|a_i|x_{i\ell}, \sum_{a_i<0}|a_i|x_{i\ell}\right\} \\
&=: b_\ell(a).
\end{aligned}$$

□

Note that $\rho_\ell = x_\ell^\top w(\tau_t)$ and $\eta_\ell = x_\ell^\top v(\tau_t)$, and here we used a generic vector a in place of $w(\tau_t)$ and $v(\tau_t)$ to keep the proof simple.

Proposition A.2. *By using the tree anti-monotonicity property i.e.,* $x_{i\ell} \ge x_{i\ell'}, \ \forall \ell' \supset \ell, \forall i \in [n+1]$ *as defined in the definition of tree (section 3.1), we have*

$$b_\ell(a) \ge b_{\ell'}(a),$$

Proof of Proposition A.2. From the definition of tree we have $x_{i\ell} \ge x_{i\ell'}, \forall \ell' \supset \ell, \forall i \in [n+1]$. Hence, we can write

$$\begin{aligned}
b_\ell(a) &= \max\left\{\sum_{a_i<0}|a_i|x_{i\ell}, \sum_{a_i>0}|a_i|x_{i\ell}\right\} \\
&\ge \max\left\{\sum_{a_i<0}|a_i|x_{i\ell'}, \sum_{a_i>0}|a_i|x_{i\ell'}\right\} \\
&=: b_{\ell'}(a).
\end{aligned}$$

□

Now, for any node ℓ' such that $\ell' \supset \ell$, we can write (8) as follows.

$$\begin{aligned}
&\left|\rho_{\ell'}(\tau_t) - \Delta_2(\ell')(\eta_{\ell'}(\tau_t) - x_{n+1,\ell'})\right| \\
&\qquad = \left|\rho_k(\tau_t) - \Delta_2(\ell')(\eta_k(\tau_t) - x_{n+1,k})\right| \qquad (11)
\end{aligned}$$

Here, we used the fact that

$$\beta_{\mathcal{A}_{\tau_t}}(\tau_{t+1}) = \beta_{\mathcal{A}_{\tau_t}}(\tau_t) + \Delta_2(\ell')\nu_{\mathcal{A}_{\tau_t}}(\tau_t).$$

The right hand side (r.h.s.) of (11) has a lower bound i.e.

$$\begin{aligned}|\rho_k(\tau_t) - &\Delta_2(\ell')(\eta_k(\tau_t) - x_{n+1,k})| \\ &\geq |\rho_k(\tau_t)| - \Delta_2(\ell')(|\eta_k(\tau_t)| + |x_{n+1,k}|),\end{aligned}$$

and the left hand side (l.h.s.) of (11) has an upper bound i.e.

$$\begin{aligned}|\rho_{\ell'}(\tau_t) - &\Delta_2(\ell')(\eta_{\ell'}(\tau_t) - x_{n+1,\ell'})| \\ &\leq |\rho_{\ell'}(\tau_t)| + \Delta_2(\ell')(|\eta_{\ell'}(\tau_t)| + |x_{n+1,\ell'}|).\end{aligned}$$

The above two bounds are derived by considering the fact that for any $a \in \mathbb{R}, b \in \mathbb{R}, c \in \mathbb{R}$ and $d \in \mathbb{R} > 0$ we can write the following:

$$\begin{aligned}&|a - d(b-c)| \leq |a| + d(|b| + |c|) \quad \text{and} \\ &|a - d(b-c)| \geq |a| - d(|b| + |c|).\end{aligned}$$

Therefore, for (11) to have a solution the following condition needs to be satisfied.

$$\begin{aligned}|\rho_{\ell'}(\tau_t)| + &\Delta_2(\ell')(|\eta_{\ell'}(\tau_t)| + |x_{n+1,\ell'}|) \\ &\geq |\rho_k(\tau_t)| - \Delta_2(\ell')(|\eta_k(\tau_t)| + |x_{n+1,k}|).\end{aligned} \tag{12}$$

Hence, (11) will not have any solution if the following condition (13) is satisfied.

$$\begin{aligned}|\rho_{\ell'}(\tau_t)| + &\Delta_2(\ell')(|\eta_{\ell'}(\tau_t)| + x_{n+1,\ell'}) \\ &< |\rho_k(\tau_t)| - \Delta_2(\ell')(|\eta_k(\tau_t)| + x_{n+1,k}).\end{aligned} \tag{13}$$

Now, using Proposition A.1 we can further write (14) which implies (13).

$$\begin{aligned}b_{\ell'}(w(\tau_t)) + &\Delta_2(\ell')(b_{\ell'}(v(\tau_t)) + x_{n+1,\ell'}) \\ &< |\rho_k(\tau_t)| - \Delta_2(\ell')(|\eta_k(\tau_t)| + x_{n+1,k}).\end{aligned} \tag{14}$$

Proof of Lemma 3.2. We now prove Lemma 3.2 by contradiction, that is we assume that at any node ℓ, the condition (10) stated in Lemma 3.2 holds, and there exists one $\ell' \supset \ell : \Delta_2(\ell') < \Delta_2(\ell_2^\dagger)$; then show that this is a contradiction.

$$\begin{aligned}\therefore \quad &|\rho_k(\tau_t)| - \Delta_2(\ell')(|\eta_k(\tau_t)| + x_{n+1,k}) \\ &> |\rho_k(\tau_t)| - \Delta_2(\ell_2^\dagger)(|\eta_k(\tau_t)| + x_{n+1,k}), \\ &\qquad\qquad \because \Delta_2(\ell') < \Delta_2(\ell_2^\dagger) \\ &> b_\ell(w(\tau_t)) + \Delta_2(\ell_2^\dagger)(b_\ell(v(\tau_t)) + x_{n+1,\ell}), \\ &\qquad\qquad \text{using (10)} \\ &> b_{\ell'}(w(\tau_t)) + \Delta_2(\ell_2^\dagger)(b_{\ell'}(v(\tau_t)) + x_{n+1,\ell'}), \\ &\qquad\qquad \text{(Proposition A.2)}, \\ &> b_{\ell'}(w(\tau_t)) + \Delta_2(\ell')(b_{\ell'}(v(\tau_t)) + x_{n+1,\ell'}), \\ &\qquad\qquad \because \Delta_2(\ell') < \Delta_2(\ell_2^\dagger).\end{aligned}$$

Therefore, we got

$$\begin{aligned}|\rho_k(\tau_t)| - &\Delta_2(\ell')(|\eta_k(\tau_t)| + x_{n+1,k}) \\ &> b_{\ell'}(w(\tau_t)) + \Delta_2(\ell')(b_{\ell'}(v(\tau_t)) + x_{n+1,\ell'}) \\ &\implies \ell' \text{ is infeasible,} \\ &\qquad \text{(using (14))} \\ &\implies \Delta_2(\ell') \geq \Delta_2(\ell_2^\dagger).\end{aligned}$$

□

This completes the proof of Lemma 3.2. Hence, if the pruning condition in Lemma 3.2 holds, then we do not need to search the sub-tree with ℓ as the root node, and hence increasing the efficiency of the search procedure.

A.2. Algorithms

The algorithms to compute the τ-path and the full-CP set of SHIM are given in Algorithm 1 and Algorithm 2 respectively. The Algorithm 1 returns the transition points ($\mathbb{T}$), the LASSO solutions ($\mathbb{B}$) and the active sets ($\mathbb{A}$) at those transition points which are subsequently provided to Algorithm 2. In Algorithm 2, for each linear piece of the τ-path, i.e. $\forall \tau \in (\tau_t, \tau_{t+1})$ for any two consecutive kinks τ_t and τ_{t+1}, we need to first identify the points at which $|w_{n+1}(\tau)| = |w_i(\tau)|, \forall i \in [n]$ (Line 4). Let us denote those points as $\{\tau_t = u_1, u_2, u_3, \ldots, u_r = \tau_{t+1}\}$ (for simplicity we slightly abused the notation here). Then we need to check that at which of those points the condition stated in (3) is satisfied (Line 5, 6, 7) to determine the full-CP set of SHIM. We used the following search range $[y_{min}, y_{max}]$ to construct the τ-path in Algorithm 1.

$$[y_{min}, y_{max}] = [y_{(0)} - 0.5(y_{(n)} - y_{(0)}), y_{(n)} + 0.5(y_{(n)} - y_{(0)})],$$

where $y_{(0)} \leq y_{(1)} \leq \ldots \leq y_{(n)}$ are the order statistics of the response vector y (see Remark 5 in (Lei, 2019) for details).

Algorithm 1 Compute τ-path

Input: $Z \in \mathbb{R}^{(n+1)\times m}, y \in \mathbb{R}^n, [y_{min}, y_{max}], \lambda$
Initialization:
  $t = 1, \tau_1 = y_{min}, \mathcal{A}_{\tau_1} = \{\ell \in [p] : \beta_\ell(\tau_1) \neq 0\}$,
  $\mathbb{T} = \{\tau_1\}, \mathbb{B} = \{\beta_{\mathcal{A}_{\tau_1}}(\tau_1)\}, \mathbb{A} = \{\mathcal{A}_{\tau_1}\}$
while $(\tau_t < y_{max})$ **do**
  Compute Δ_{ℓ^*} using (7) and Lemma 3.2
  if $\Delta_{\ell^*} = \Delta_1(\ell_1^*)$ **then**

    $\mathcal{A}_{\tau_{t+1}} \leftarrow \mathcal{A}_{\tau_t} \setminus \{\ell_1^*\}$ → remove ℓ_1^* from $\mathcal{A}_{\tau_t}$
  end if
  if $\Delta_{\ell^*} = \Delta_2(\ell_2^*)$ **then**

    $\mathcal{A}_{\tau_{t+1}} \leftarrow \mathcal{A}_{\tau_t} \cup \{\ell_2^*\}$ → add ℓ_2^* into $\mathcal{A}_{\tau_t}$
  end if
  Update:
    $\tau_{t+1} \leftarrow \tau_t + \Delta_{\ell^*}$
    $\beta_{\mathcal{A}_{\tau_t}}(\tau_{t+1}) \leftarrow \beta_{\mathcal{A}_{\tau_t}}(\tau_t) + \Delta_{\ell^*} \nu_{\mathcal{A}_{\tau_t}}(\tau_t)$
    $\mathbb{T} = \mathbb{T} \cup \{\tau_{t+1}\}$
    $\mathbb{B} = \mathbb{B} \cup \{\beta_{\mathcal{A}_{\tau_t}}(\tau_{t+1})\}$
    $\mathbb{A} = \mathbb{A} \cup \{\mathcal{A}_{\tau_{t+1}}\}$
    t = t +1
end while
Output: $\mathbb{T}, \mathbb{B}, \mathbb{A}$

Algorithm 2 Compute full-CP

1: **Input:** $Z \in \mathbb{R}^{(n+1)\times m}, y \in \mathbb{R}^n, \alpha \in [0,1], \mathbb{T}, \mathbb{B}, \mathbb{A}$
2: Initialization: $t = 0, t_{max} = |\mathbb{T}|, \mathcal{C} = \{\emptyset\}$
3: **while** $(t < t_{max})$ **do**
4: $\{u_2, \ldots, u_{r-1}\} = \{u \in (\tau_t, \tau_{t+1})$ such that $|w_i(u)| = |w_{n+1}(u)| \forall i \in [n]\}$
5: $\mathcal{T}_t = \{\tau_t\} \cup \{u_2, u_3, \ldots, u_{r-1}\} \cup \{\tau_{t+1}\}$
6: $\mathcal{H}_t = \{h \in \{1, \ldots, |\mathcal{T}_t|\} : \pi(\mathcal{T}_t(h)) \geq \alpha\}$ using $\mathbb{B}$ and $\mathbb{A}$
7: $\mathcal{C} = \mathcal{C} \cup_{h \in \mathcal{H}_t} [\mathcal{T}_t(h), \mathcal{T}_t(h+1))$
8: t = t +1
9: **end while**
10: **Output:** $\mathcal{C}$

A.3. Extension for Elastic Net (ENet-SHIM)

A common problem of the LASSO is that if the data has correlated features, then the LASSO picks only one of them and ignores the rest, which leads to instability. To solve this problem (Zou & Hastie, 2005) proposed the Elastic Net. This feature correlation problem is very much evident in the SHIM-type problem, and hence we extended our framework for the Elastic Net. We solve the following optimization problem to extend our framework for the Elastic Net:

$$\beta(\tau) \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2} \|y(\tau) - X\beta\|_2^2 + \frac{1}{2}\alpha \|\beta\|_2^2 + \lambda \|\beta\|_1 . \tag{15}$$

The elastic net optimization problem can actually be formulated as a LASSO optimization problem using augmented data. If we consider an augmented data defined as $\tilde{X} = \begin{pmatrix} X \\ \sqrt{\alpha} I_p \end{pmatrix} \in \mathbb{R}^{(n+1+p)\times p}$ and $\tilde{y}(\tau) = \begin{pmatrix} y(\tau) \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{n+1+p}$, where $I_p \in \mathbb{R}^{p\times p}$ is an identity matrix and $\mathbf{0} \in \mathbb{R}^p$ is a zero vector, then solving the elastic net optimization problem (15) for a fixed λ, is equivalent to solving the following problem.

$$\beta(\tau) \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2} \left\| \tilde{y}(\tau) - \tilde{X}\beta \right\|_2^2 + \lambda \|\beta\|_1 .$$

Step size (τ-path): If we consider two real values τ_t and τ_{t+1} ($\tau_{t+1} > \tau_t$) at which the active set does not change, and their signs also remain the same, then we can write

$$\beta_{\mathcal{A}_{\tau_t}}(\tau_{t+1}) - \beta_{\mathcal{A}_{\tau_t}}(\tau_t) = \nu_{\mathcal{A}_{\tau_t}}(\tau_t)(\tau_{t+1} - \tau_t),$$

where

$$\nu_{\mathcal{A}_{\tau_t}}(\tau_t) = (X_{\mathcal{A}_{\tau_t}}^\top X_{\mathcal{A}_{\tau_t}} + \alpha I_{|\mathcal{A}_{\tau_t}|})^{-1} x_{n+1;\mathcal{A}_{\tau_t}}.$$

Note that here the only change compared to the vanilla SHIM is the addition of an $\alpha I_{|\mathcal{A}_{\tau_t}|}$ term to the expression of $\nu_{\mathcal{A}_{\tau_t}}(\tau_t)$. Now, one can also derive a similar expression of the step size of inclusion and deletion as done for the vanilla SHIM τ-path (7), considering the updated expression of $\nu_{\mathcal{A}_{\tau_t}}(\tau_t)$.

A.3.1. TREE PRUNING (τ-PATH)

The no solution condition (13) with the augmented data $(\tilde{X}, \tilde{y})$ can be written as follows:

$$\begin{aligned} &|\tilde{\rho}_{\ell'}(\tau_t)| + \Delta_2(\ell')(|\tilde{\eta}_{\ell'}(\tau_t)| + x_{n+1,\ell'}) \\ &\qquad < |\tilde{\rho}_k(\tau_t)| - \Delta_2(\ell')(|\tilde{\eta}_k(\tau_t)| + x_{n+1,k}), \end{aligned} \tag{16}$$

where

$$\begin{aligned} \tilde{\rho}_{\ell'}(\tau_t) &= \tilde{x}_{\ell'}^\top \tilde{w}(\tau_t) \text{ and} \\ \tilde{\eta}_{\ell'}(\tau_t) &= \tilde{x}_{\ell'}^\top \tilde{v}(\tau_t), \forall \ell' \in \mathcal{A}_{\tau_t}^c, \\ \tilde{\rho}_k(\tau_t) &= \tilde{x}_k^\top \tilde{w}(\tau_t) \text{ and} \\ \tilde{\eta}_k(\tau_t) &= \tilde{x}_k^\top \tilde{v}(\tau_t), \forall k \in \mathcal{A}_{\tau_t}, \\ \tilde{w}(\tau_t) &= \tilde{y}(\tau_t) - \tilde{X}_{\mathcal{A}_{\tau_t}} \beta_{\mathcal{A}_{\tau_t}}(\tau_t) \in \mathbb{R}^{n+1+p}, \\ \tilde{v}(\tau_t) &= \tilde{X} \nu_{\tau_t} \in \mathbb{R}^{n+1+p}. \end{aligned}$$

Now one can show that the tree pruning condition for the τ-path of ENet-SHIM can be defined as stated in Lemma A.3.

Lemma A.3. *For any given node ℓ, if $\Delta_2(\ell_2^\dagger)$ is the current minimum step size, that is,*

$$\ell_2^\dagger = \underset{j \in \{1,2,\ldots,\ell\} \cap \mathcal{A}_{\tau_t}^c}{\arg\min} \Delta_2(j),$$

then $\forall \ell' \supset \ell, \Delta_2(\ell') \geq \Delta_2(\ell_2^\dagger)$ if

$$\begin{aligned} &b_\ell(w(\tau_t)) + \Delta_2(\ell_2^\dagger)(b_\ell(v(\tau_t)) + x_{n+1,\ell}) \\ &\qquad < |\bar{\rho}_k(\tau_t)| - \Delta_2(\ell_2^\dagger)(|\bar{\eta}_k(\tau_t)| + x_{n+1,k}), \end{aligned} \tag{17}$$

where

$$\bar{\rho}_k(\tau_t) = \sum_{i=1}^{n} w_i(\tau_t) x_{ik} - \alpha\beta_k,$$
$$\bar{\eta}_k(\tau_t) = \sum_{i=1}^{n} v_i(\tau_t) x_{ik} + \alpha\nu_k.$$

Although, theoretically the standard elastic net can be solved as a LASSO optimization problem by considering an augmented dataset as explained in section A.3, this data augmentation can be prohibitively expensive in case of ENet-SHIM due to the combinatorial effects of interaction terms. Actually, by expanding the expression of $\tilde{\rho}_{\ell'}(\tau_t)$, $\tilde{\rho}_k(\tau_t)$, $\tilde{\eta}_{\ell'}(\tau_t)$ and $\tilde{\eta}_k(\tau_t)$ separately for the original data and the augmented part of the augmented data, one can show that most of the terms will disappear, and one just need to consider additional terms $-\alpha\beta_k(\tau_t)$ and $\alpha\nu_k(\tau_t)$ while evaluating the expression of $\tilde{\rho}_k(\tau_t)$ and $\tilde{\eta}_k(\tau_t)$ respectively. However, $\tilde{\rho}_{\ell'}(\tau_t)$ and $\tilde{\eta}_{\ell'}(\tau_t)$ will remain the same as in the original data, i.e., $\tilde{\rho}_{\ell'}(\tau_t)$=$\rho_{\ell'}(\tau_t)$ and $\tilde{\eta}_{\ell'}(\tau_t)$=$\eta_{\ell'}(\tau_t)$. The formal proof of Lemma A.3 is given below.

Proof of Lemma A.3. Let's consider

$$\tilde{w}(\tau_t) = \tilde{y}(\tau_t) - \tilde{X}_{\mathcal{A}_{\tau_t}} \beta_{\mathcal{A}_{\tau_t}}(\tau_t) \in \mathbb{R}^{n+1+p}$$

and

$$w(\tau_t) = y(\tau_t) - \bar{X}_{\mathcal{A}_{\tau_t}} \beta_{\mathcal{A}_{\tau_t}}(\tau_t) \in \mathbb{R}^{n+1},$$

where $p = |\mathcal{A}_{\tau_t}| + |\mathcal{A}^c_{\tau_t}|$, then we can write

$$\tilde{w}_i(\tau_t) = \begin{cases} w_i(\tau_t) & \text{if} \quad i \le n+1, \\ -\sqrt{\alpha}\beta_{\ell'}(\tau_t) & \text{if} \quad n+1 < i \le n+1+|\mathcal{A}_{\tau_t}|, \\ & \qquad\qquad \forall \ell' \in \mathcal{A}_{\tau_t}, \\ 0 & \text{if} \quad n+1+|\mathcal{A}_{\tau_t}| < i \le n+1+p. \end{cases} \tag{18}$$

Similarly considering $\tilde{v}(\tau_t) = \tilde{X}\nu(\tau_t) \in \mathbb{R}^{n+1+p}$ and $v(\tau_t) = \bar{X}\nu(\tau_t) \in \mathbb{R}^{n+1}$, we can write

$$\tilde{v}_i(\tau_t) = \begin{cases} v_i(\tau_t) & \text{if} \quad i \le n+1, \\ \sqrt{\alpha}\nu_{\ell'}(\tau_t) & \text{if} \quad n+1 < i \le n+1+|\mathcal{A}_{\tau_t}|, \\ & \qquad\qquad \forall \ell' \in \mathcal{A}_{\tau_t}, \\ 0 & \text{if} \quad n+1+|\mathcal{A}_{\tau_t}| < i \le n+1+p, \end{cases} \tag{19}$$

and, considering $\tilde{X} \in \mathbb{R}^{(n+1+p)\times p}$ and $X \in \mathbb{R}^{(n+1)\times p}$ we can write

$$\tilde{x}_{i\ell'} = \begin{cases} x_{i\ell'} & \text{if} \quad i \le n+1, \\ \sqrt{\alpha} & \text{if} \quad i > n+1 \ \text{ and } \ (i-n-1) = \ell', \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Therefore, $\forall \ell' \in \mathbb{R}^p$ we can write

$$\begin{aligned} \tilde{\rho}_{\ell'}(\tau_t) &= \tilde{x}_{\ell'}^\top \tilde{w}(\tau_t) \\ &= \sum_{i=1}^{n+1+p} \tilde{w}_i(\tau_t)\tilde{x}_{i\ell'} \\ &= \sum_{i=1}^{n+1} \tilde{w}_i(\tau_t)\tilde{x}_{i\ell'} + \sum_{i=n+2}^{n+1+|\mathcal{A}_{\tau_t}|} \tilde{w}_i(\tau_t)\tilde{x}_{i\ell'} \\ &\quad + \sum_{i=n+2+|\mathcal{A}_{\tau_t}|}^{n+p} \tilde{w}_i(\tau_t)\tilde{x}_{i\ell'}. \end{aligned}$$

Now, using (18) and (20) the second and the third quantity in the above expression can be written as follows:

$$\sum_{i=n+2}^{n+1+|\mathcal{A}_{\tau_t}|} \tilde{w}_i(\tau_t)\tilde{x}_{i\ell'} = \begin{cases} (-\sqrt{\alpha}\beta_{\ell'}(\tau_t))(\sqrt{\alpha}), \\ \qquad\qquad \text{if } (i-n-1)=\ell', \\ 0 \quad \text{otherwise,} \end{cases}$$

and,

$$\sum_{i=n+2+|\mathcal{A}_{\tau_t}|}^{n+p} \tilde{w}_i(\tau_t)\tilde{x}_{i\ell'} = 0.$$

Therefore,

$$\begin{aligned} \tilde{\rho}_{\ell'}(\tau_t) &= \tilde{x}_{\ell'}^\top \tilde{w}(\tau_t) \\ &= \sum_{i=1}^{n+1} w_i(\tau_t)x_{i\ell'} \\ &= \rho_{\ell'}(\tau_t), \ \ \forall \ell' \in \mathcal{A}_{\tau_t}^c \ \text{ and,} \end{aligned}$$

$$\begin{aligned} \tilde{\rho}_k(\tau_t) &= \tilde{x}_k^\top \tilde{w}(\tau_t) \\ &= \sum_{i=1}^{n+1} w_i(\tau_t)x_{ik} - \alpha\beta_k(\tau_t) \\ &= \rho_k(\tau_t) - \alpha\beta_k(\tau_t), \ \ \forall k \in \mathcal{A}_{\tau_t} \\ &=: \bar{\rho}_k(\tau_t). \end{aligned}$$

Similarly, using (19) and (20) we can write

$$\begin{aligned} \tilde{\eta}_{\ell'}(\tau_t) &= \tilde{x}_{\ell'}^\top \tilde{v}(\tau_t) \\ &= \sum_{i=1}^{n+1} v_i(\tau_t)x_{i\ell'} \\ &= \eta_{\ell'}(\tau_t), \ \ \forall \ell' \in \mathcal{A}_{\tau_t}^c, \qquad \text{and,} \end{aligned}$$

$$\begin{aligned} \tilde{\eta}_k(\tau_t) &= \tilde{x}_k^\top \tilde{v}(\tau_t) \\ &= \sum_{i=1}^{n+1} v_i(\tau_t)x_{ik} + \alpha\nu_k(\tau_t) \\ &= \eta_k(\tau_t) + \alpha\nu_k(\tau_t), \ \ \forall k \in \mathcal{A}_{\tau_t} \\ &= \bar{\eta}_k(\tau_t). \end{aligned}$$

□

Therefore, we can write (16) as follows.

$$\begin{aligned} &|\rho_{\ell'}(\tau_t)| + \Delta_2(\ell')(|\eta_{\ell'}(\tau_t)| + x_{n+1,\ell'}) \\ &\qquad < |\bar{\rho}_k(\tau_t)| - \Delta_2(\ell')(|\bar{\eta}_k(\tau_t)| + x_{n+1,k}), \end{aligned} \tag{21}$$

Now, using PropositionA.1, we can simplify (21) as follows.

$$\begin{aligned} &b_{\ell',w(\tau_t)} + \Delta_2(\ell')(b_{\ell',v(\tau_t)} + x_{n+1,\ell'}) \\ &\qquad < |\bar{\rho}_k(\tau_t)| - \Delta_2(\ell')(|\bar{\eta}_k(\tau_t)| + x_{n+1,k}), \end{aligned} \tag{22}$$

Now similar to the Lemma 3.2 one can formally prove the Lemma A.3 using (22) and Proposition A.2.

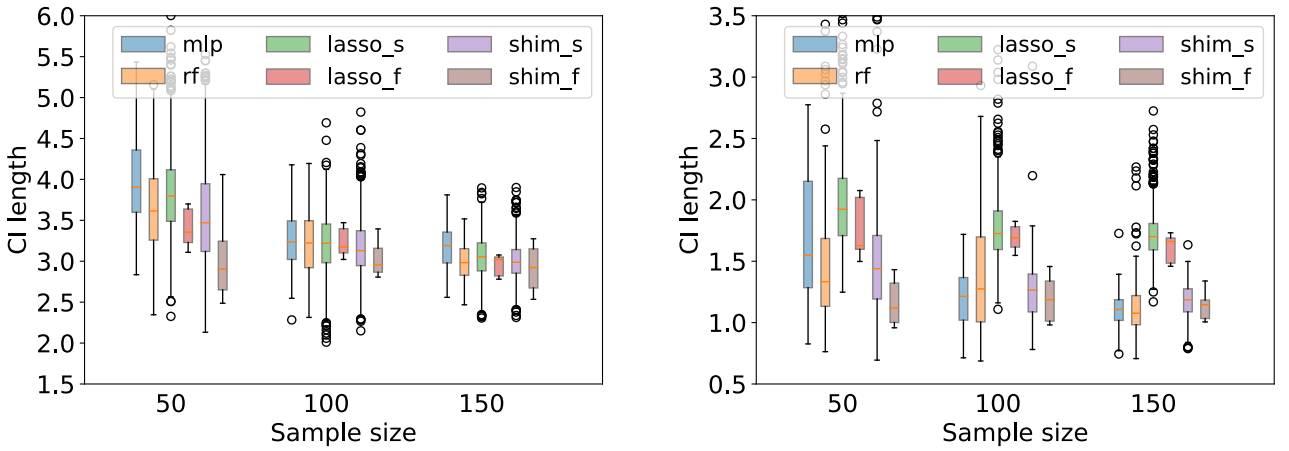


Figure 6: Comparison of the confidence interval lengths (CI length) of the proposed method (shim) with other simple (lasso) and complex models (mlp, rf) using synthetic data for different sample sizes. Left figure shows the results using model-1 (weak signal) and the right figure shows the results using model-2 (strong signal). Here, we did not mention any max order of interaction and the entire search space is used to choose the best SHIM model by the algorithm automatically.

B. Additional Results and Experimental Details

B.1. Hyper parameter selection

The hyper parameter selection for all the methods (e.g. λ in LASSO, SHIM; hidden layer's sizes in MLP, etc.) is done based on 5-fold cross-validation using a separate set of 15 independent data sets. These 15 data sets which are used for model selection are different from the 15 independent data sets (3X5=15 data sets) that are used to report the statistical performances. For example, in the case of synthetic data experiments, we considered different training sizes (n=100, 150, 200) for each data set. For each setting (e.g., n=100), we found the hyper parameters by averaging over this separate set of 15 independent data sets (e.g., of n=100). Once, we found the best hyper parameter, then we used that to report statistical performance using a different set of 15 independent data sets (e.g., 3X5=15 data sets of n=100). For MLP, the activations are chosen from {identity, relu, logistic, tanh} and the hidden layer's sizes are chosen from all possible combinations of {50, 100, 150} nodes, considering both 2-hidden layers and 3-hidden layers architectures. The most frequent activation and architecture chosen based a on separate set of 15 independent datasets are considered to report the average results. Similarly for RF, the number of estimators (n_estimators) and the min samples in the leaf of a tree (min_samples_leaf) are chosen from {50, 100, 200} and {0.1, 0.05, 0.01} respectively. The median λ value based on the separate set of 15 independent datasets is considered to report the results of LASSO and SHIM. For the λ selection, we considered the range of λ defined in $[\lambda_{max}/2, 0)$. For, MLP and RF, we used the standard *scikit-learn* implementation.

B.2. Additional results using synthetic data

To further compare the performance of the proposed method we considered highly sparse data ($\zeta = 0.6$) in two different experimental settings. (1) **model-1**: We generated a model with weak signals that is we considered a true model of up to third-order interactions, which is defined as $\mu(x_i) = 1.0z_1 + 1.0z_1z_2 + 1.0z_1z_2z_3$ and (2) **model-2**: We generated a model with strong signals that is we considered a true model of up to third-order interactions, which is defined as $\mu(x_i) = 5.0z_1 + 5.0z_1z_2 + 5.0z_1z_2z_3$. In both the settings we set $\sigma = 1$. The choice of these models is merely for demonstration purposes, and the proposed method is equally applicable to any chosen model. For both the settings (weak and strong), we generated a dataset of $n \in \{50, 100, 150\}$ training instances, each accompanied with $n = 10$ test instances. Here we considered a covariate size of $m = 5$. We generated 3 such independent random datasets. Hence, in total we reported the average results of $3 \times 10 = 30$ test instances. For split-CP, we repeated the experiments 30 times to highlight the effect of randomness in the confidence set generation. For SHIM, we did not mention any max-pat size of d, i.e. the entire search space is considered for exploration (tree generation) and the proposed tree pruning condition takes care of it to improve the efficiency of the search. The best shim model is automatically chosen by the algorithm corresponding to the best hyper parameter λ (see section B.1 for details of hyper parameter selection). The results are shown in Fig. 6 where the left figure corresponds to the model-1

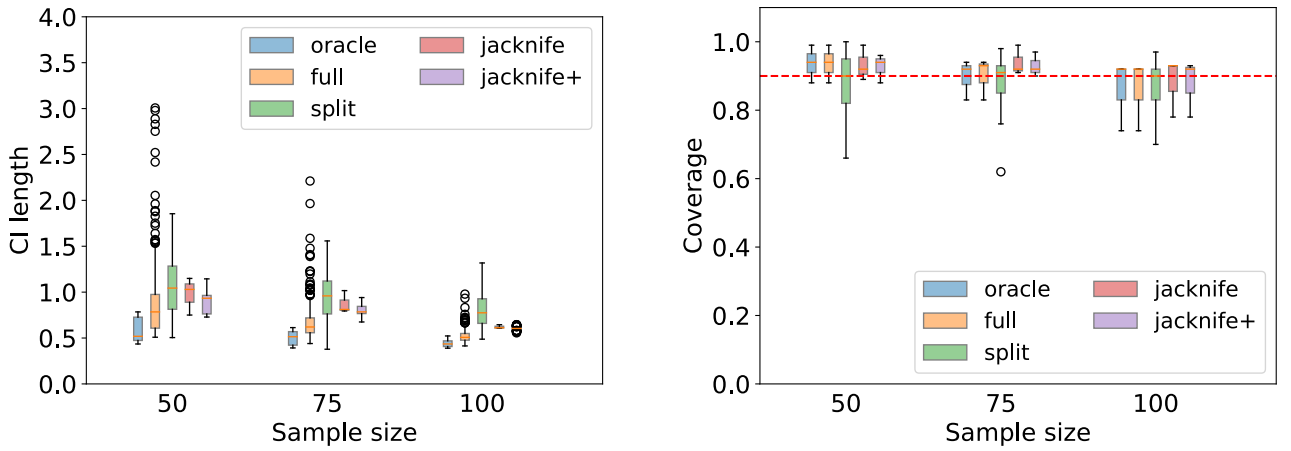


Figure 7: Comparing confidence interval lengths (CI lengths) and coverages among different methods (oracle, full, split, jacknife and jacknife+) of CP set constructions for SHIM using $\lambda = 0.1$. For oracle-CP we used true y_{n+1} in both the training and calibration phases. Note that in reality we don't know true y_{n+1}.

(weak signal) and the right figure corresponds to the model-2 (strong signal). In both the settings, the full-CP methods (lasso_f, shim_f) produced more compact confidence sets irrespective of sample sizes. It can be observed that in case of highly sparse data, the shim model produces better results, more specifically when the sample size is small and the signal is weak.

B.3. Comparing with aggregate-CP (jackknife and jackknife+)

For the sake of completeness we also compared our exact full-CP method with aggregate-CP methods such as jackknife and jackknife+. We compared the confidence interval lengths (CI length), coverage and the time taken for different sample sizes The results are shown in Figures 7, 8 and 9. We reported results for two different λ values ($\lambda = 0.1$ and $\lambda = 0.01$). In CP, the choice of $\alpha \in [0, 1]$ determines the level of confidence $(1 - \alpha)$ in the prediction. This essentially determines the statistical efficiency (length of the CP set). A high confidence generally leads to a wider confidence set. For example, a 90% confidence set is generally wider than a 80% confidence set. Therefore, if we specify $\alpha = 0.1$, then `split-CP` and `full-CP` guarantees a $(1 - 0.1) \times 100\% = 90\%$ confidence set, whereas `aggregate-CP` can only guarantee a $(1 - 2 \times 0.1) \times 100\% = 80\%$ confidence set for the same α. Therefore, if we want to ensure 90% confidence in `aggregate-CP`, then this will lead to a wider confidence set. Please see the results in Figures 7 and 8. Here, we want to highlight that the best possible theoretical coverage guarantee of `jacknife+` is $1 - 2\alpha$, whereas `jacknife` has no guarantee. Both `jacknife` and `jacknife+` are also computationally very expensive (they require n model fits) for large sample size n as shown in Figure 9.

Experimental settings of Figures 7 and 8.
We used "sklearn.datasets.make_regression" to generate synthetic data. The following parameters have been used: n_features=5, n_informative=3, noise=1. $n_{train} \in [50, 75, 100]$ and $n_{test} = 100$. We repeated experiments 3 times, hence we reported results of $3 \times 100 = 300$ test instances. For `split-CP`, we repeated experiments 30 times to showcase the effect of randomization. We reported results for two different λ values ($\lambda \in \{0.1, 0.01\}$).

Experimental settings of Figures 9.

We used "sklearn.datasets.make_regression" to generate synthetic data. The following parameters have been used: n_features=5, n_informative=3, noise=1. $n_{train} \in [100, 500, 1000]$ and $n_{test} = 5$. We repeated experiments 3 times, hence we reported results of $3 \times 5 = 15$ test instances. For `split-CP`, we repeated experiments 5 times. We reported results for two different λ values ($\lambda \in \{0.1, 0.01\}$).

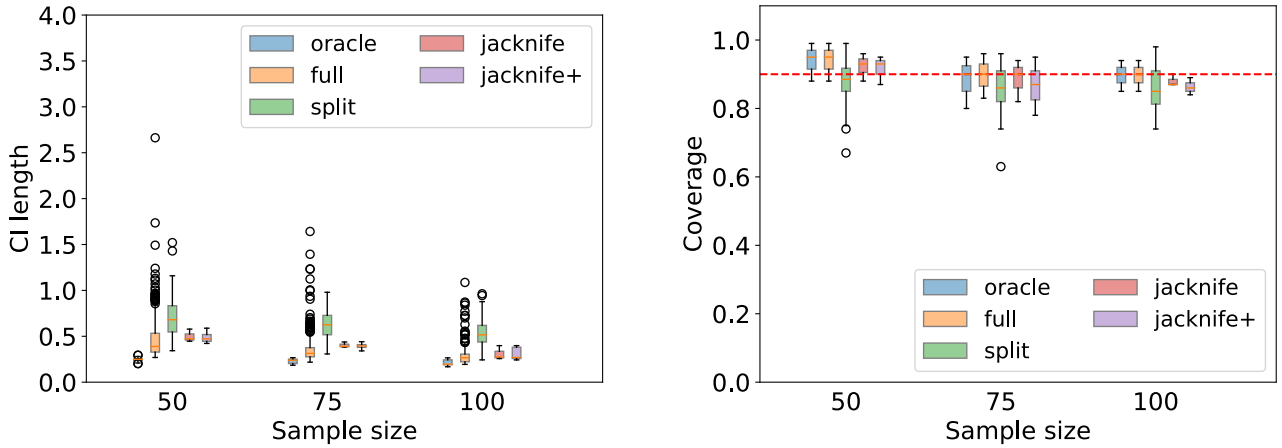
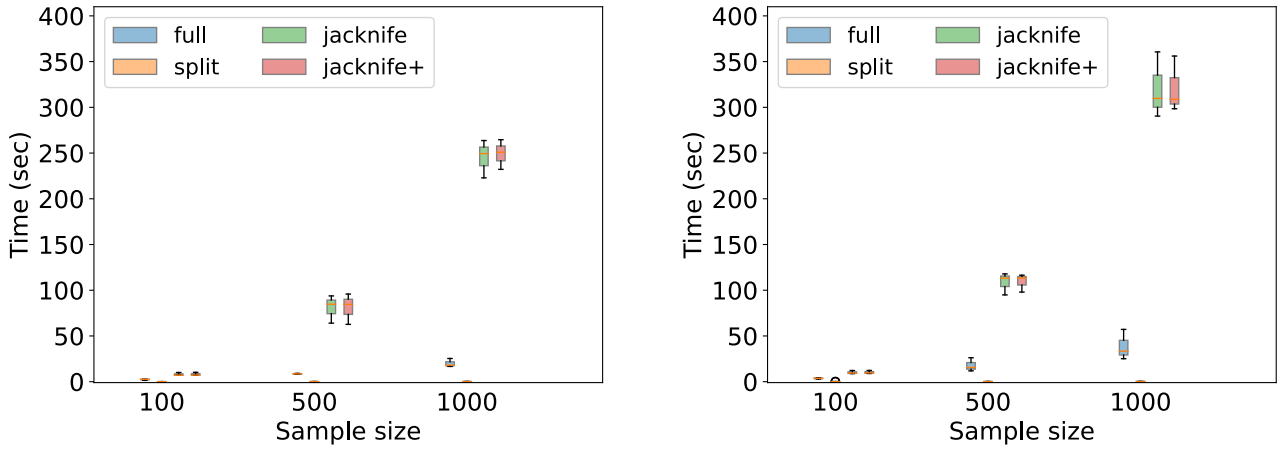


Figure 8: Comparing confidence interval lengths (CI lengths) and coverages among different methods (oracle, full, split, jacknife and jacknife+) of CP set constructions for SHIM using $\lambda = 0.01$. For oracle-CP we used true y_{n+1} in both the training and calibration phases. Note that in reality we don't know true y_{n+1}.



$\lambda = 0.1$ $\lambda = 0.01$

Figure 9: Comparing execution times among different methods (full, split, jacknife and jacknife+) of CP set constructions for SHIM using different sample sizes. For oracle-CP we used true y_{n+1} in both the training and calibration phases. Note that in reality we don't know true y_{n+1}.

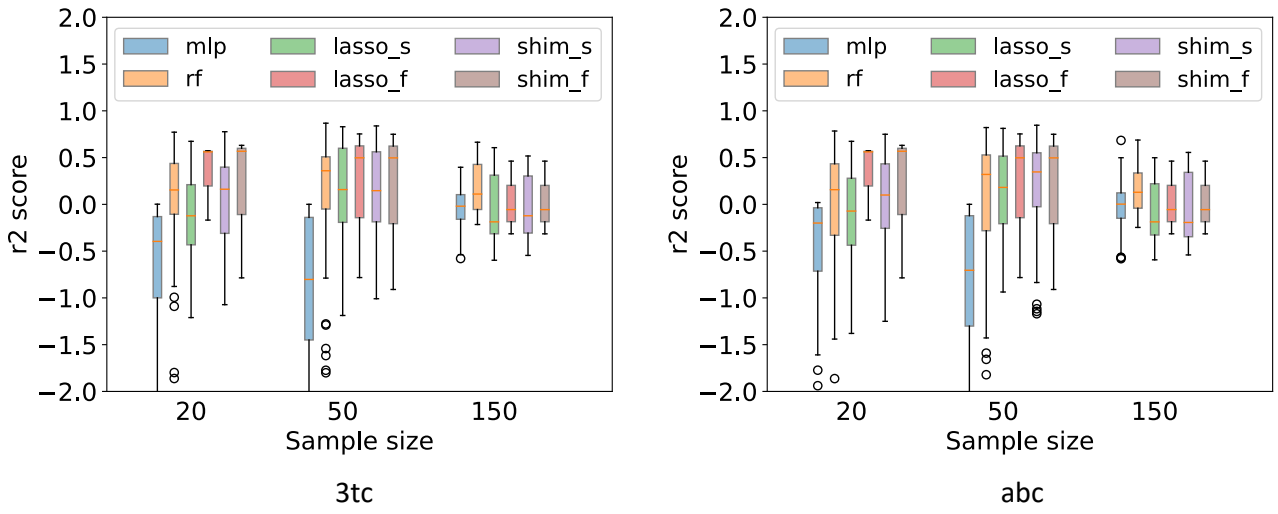


Figure 10: Comparing r2-scores of different methods using three different sample sizes for hiv datasets (3tc and abc).

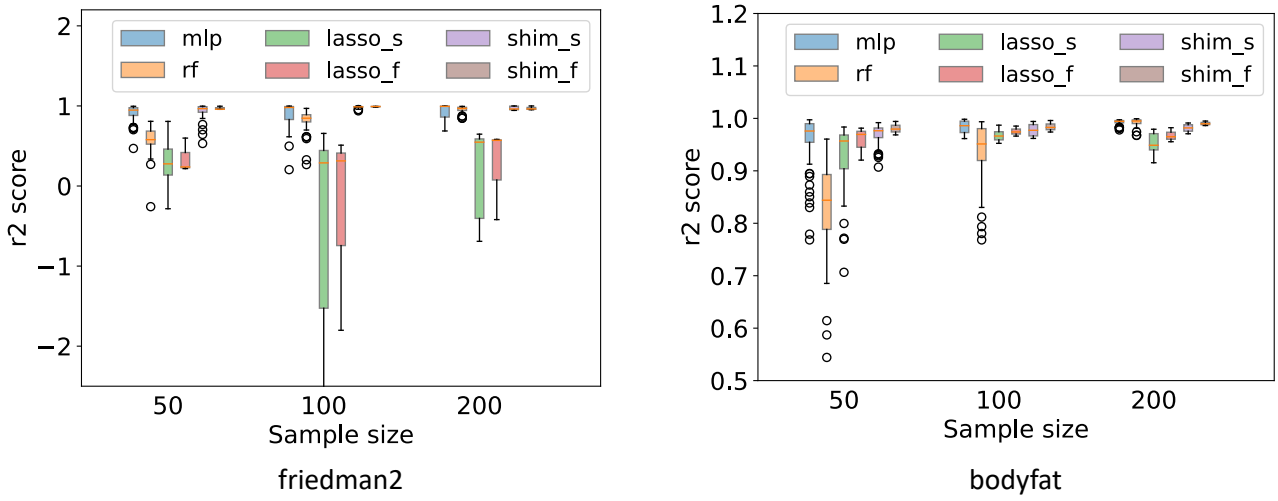


Figure 11: Comparing r2-scores of different methods using three different sample sizes for friedman2 and bodyfat datasets.

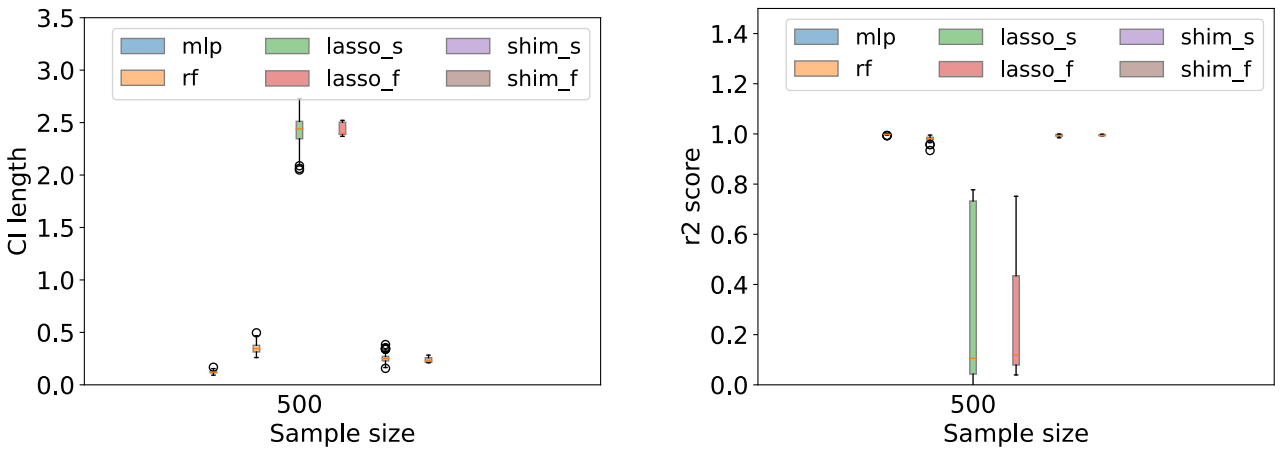


Figure 12: Comparison of the confidence interval lengths (CI length) and r2 scores of the proposed method (shim) with other simple (lasso) and complex models (mlp, rf) using continuous synthetic (friedman2) for a large sample size $n = 500$. shim_s and shim_f respectively represent split-CP and full-CP for a SHIM. Here, we did not mention any max order of interaction and the entire search space is used to choose the best SHIM model by the algorithm automatically.

d	Search space (# nodes)	$\lambda = 1$			$\lambda = 10$		
		$\zeta = 0.4$	$\zeta = 0.7$	$\zeta = 0.9$	$\zeta = 0.4$	$\zeta = 0.7$	$\zeta = 0.9$
2	465	200	134	7	9	7	2
3	4525	317	129	7	9	7	2
4	31930	314	129	7	9	7	2
5	174436	312	129	7	9	7	2
10	53009101	312	129	7	9	7	2
15	614429671	312	129	7	9	7	2
20	1050777736	312	129	7	9	7	2
25	1073709892	312	129	7	9	7	2

Table 4: Number of homotopy transition points (# kinks) along the τ-path of SHIM using two different λ values ($\lambda = 1, 10$) for three different sparsity levels ($\zeta = 0.4, 0.7, 0.9$) for different value of "max_pat" (d) sizes.

REFERENCE

Zou, H., & Hastie, T. (2005) Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2): 301–320.