A Confidence Machine for Sparse High-Order Interaction Model (Supplementary rebuttal)

March 19, 2023

Reviewer ug66

Q1. Why don't you show r-squared for HIV, friedman2, bodyfat?

Please see the results in Figures 1, 2, 3 and 4. Here, we compared the confidence interval lengths (CI length) and r2 scores of the proposed method (shim) with other simple (lasso) and complex (mlp, rf) models. Here, shim_s and shim_f respectively represent `split-CP` and `full-CP` for a SHIM. Similar notations are used for LASSO. 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.

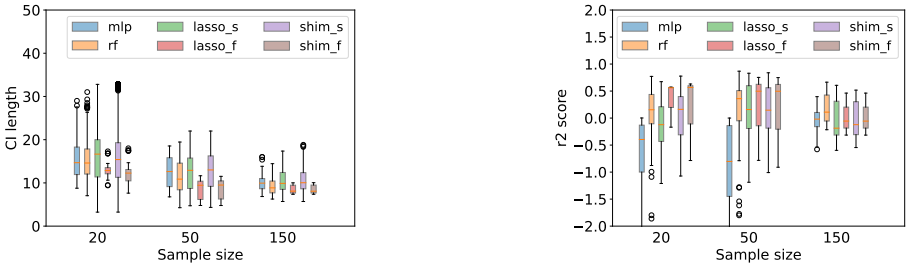


Figure 1: Results using HIV drug resistance data (3tc).

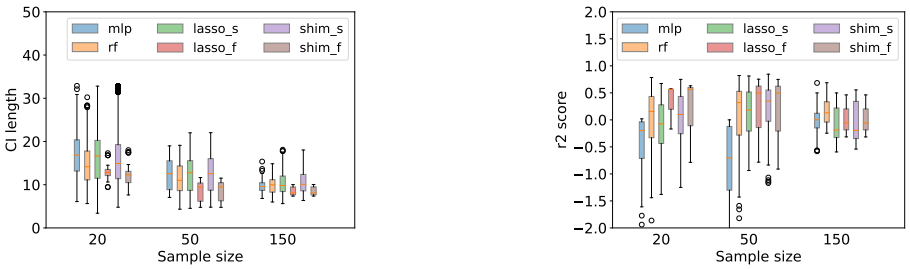


Figure 2: Results using HIV drug resistance data (abc).

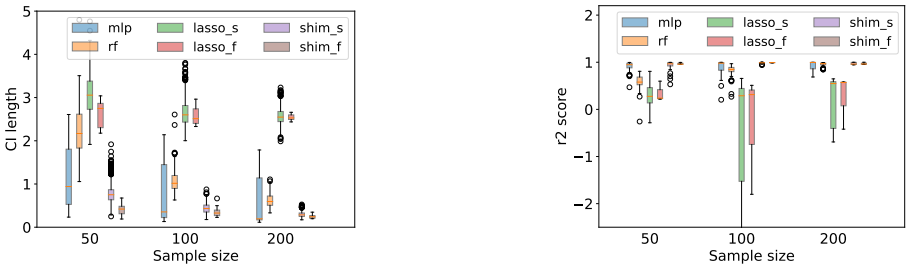


Figure 3: Results using continuous synthetic (friedman2) data.

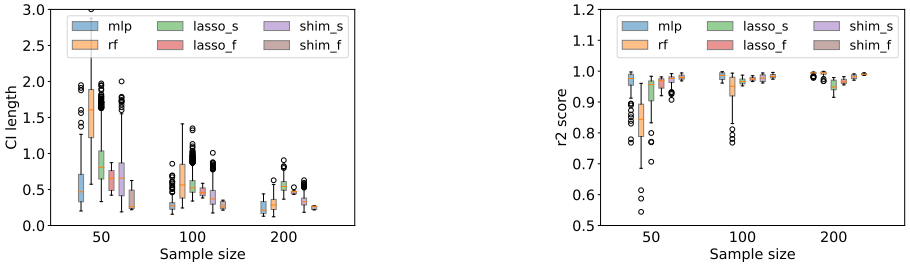


Figure 4: Results using continuous real world (bodyfat) data.

Reviewer 3MZo

Q6. Compare performance on large dataset ($n > 200$).

Please see the results in Figure 5

C1. Definition of $\mathcal{S}_i(\tau)$

The $\mathcal{S}_i, \forall i \in [n+1]$ is the absolute residual of the i^{th} data point. To compute this residual we fit the model with all $n+1$ data points, hence the full design matrix is used. The absolute residual of each data point does not depend on

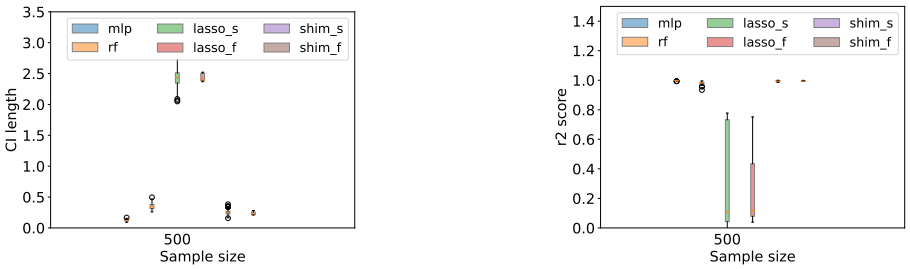


Figure 5: Results using synthetic continuous (friedman2) data for large sample size ($n = 500$).

the order of the data and hence $\pi(\tau)$ also does not depend on the order of the data.

C2. Why `split-CP` is computationally more efficient? Note that to naively construct a `full-CP` set in a regression setting, one needs to fit the model infinite times for all possible values of $y \in \mathbb{R}$ on the regression line. The number of model fittings depends on y and not on training sample size n. Hence, the naive computation of `full-CP` is prohibitive. Whereas, in case of a `split-CP`, one just needs to fit the model only once using the one part of the data and construct the CP set using the remaining part. That is why `split-CP` is computationally more efficient.

C3. The `aggregate-CP` provides an inflated confidence level of $1-2\alpha$. What is the problem?

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 6 and 7. 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 Figures 8a and 8b.

Experimental settings of Figures 6 and 7.

We used "sklearn.datasets.make_regression" to generate synthetic data. The following parameters have been used: n_features=5, n_informative=3, noise=1.

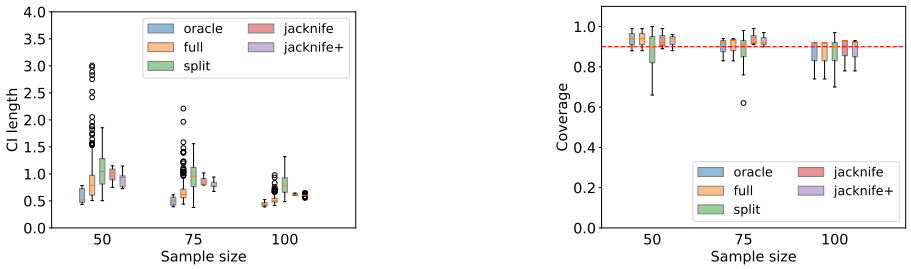


Figure 6: Comparing confidence interval lengths (CI lengths) and coverages among different methods of CP set constructions (oracle, full, split, jacknife and jacknife+) using $\lambda = 0.1$.

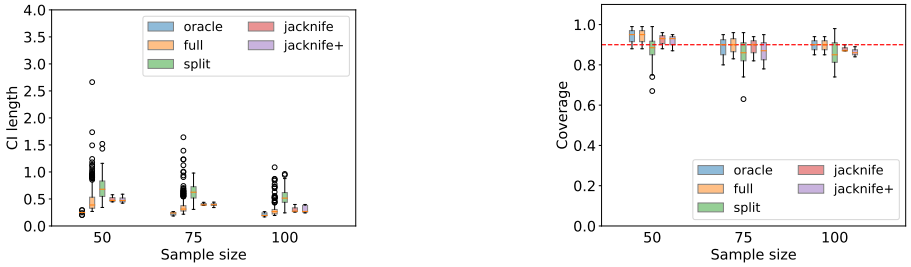
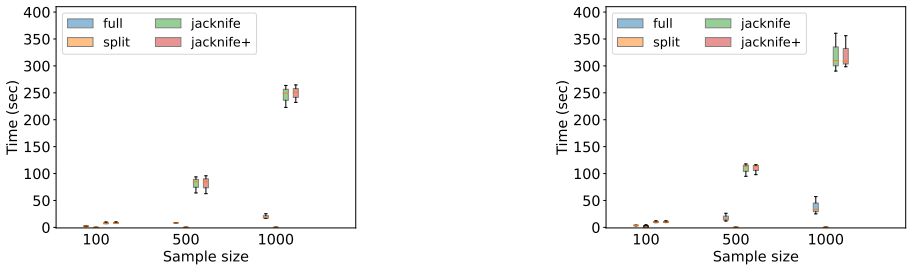


Figure 7: Comparing confidence interval lengths (CI lengths) and coverages among different methods of CP set constructions (oracle, full, split, jacknife and jacknife+) using $\lambda = 0.01$.

$n_{train} \in [50, 75, 100]$ 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. 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}. We reported results for two different λ values ($\lambda \in \{0.1, 0.01\}$).

Experimental settings of Figures 8a and 8b.

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 ($\lambda \in \{0.1, 0.01\}$).



(a) $\lambda = 0.1$ (b) $\lambda = 0.01$

Figure 8: Comparing execution times among different methods of CP set constructions (full, split, jacknife and jacknife+) for different sample sizes.

C4. In Table 1 and 2, what do "s" and "f" mean in shim_2s and shim_2f?

The shim_2s and shim_2f respectively represent `split-CP` and `full-CP` for a 2^{nd} order SHIM. We will add this as a caption of Tables 1 and 2 in the revised version of the paper as we did in the Figures 2, 3 and 4 of the paper.