Symbolic-regression boosting

Moshe Sipper1 · Jason H. Moore2

Received: 26 November 2020 / Revised: 3 February 2021 / Accepted: 9 February 2021 / Published online: 23 March 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract
Modifying standard gradient boosting by replacing the embedded weak learner in favor of a strong(er) one, we present SyRBo: symbolic-regression boosting. Experiments over 98 regression datasets show that by adding a small number of boosting stages—between 2 and 5—to a symbolic regressor, statistically significant improvements can often be attained. We note that coding SyRBo on top of any symbolic regressor is straightforward, and the added cost is simply a few more evolutionary rounds. SyRBo is essentially a simple add-on that can be readily added to an extant symbolic regressor, often with beneficial results.

Keywords Symbolic regression · Gradient boosting · Genetic programming

1 Introduction

In machine learning, a weak learner is defined as a learner that can produce an hypothesis that performs only slightly better than random guessing, while a strong learner can with high probability output an hypothesis that is correct on all but an arbitrarily small fraction of the instances.

In his seminal paper, “The strength of weak learnability”, Schapire [15] described a method “for converting a weak learning algorithm into one that achieves arbitrarily high accuracy.” Over the years, a plethora of highly successful boosting algorithms that transform weak learners into strong ones have been devised [1, 3, 4, 9].

A recent rigorous benchmarking study of four symbolic regression algorithms versus nine machine learning approaches found that “symbolic regression performs strongly compared to state-of-the-art gradient boosting algorithms” (they also found that “in terms of running times [symbolic regression] is among the slowest of the available methodologies”) [12]. Herein we wish to combine boosting with symbolic

* Moshe Sipper
sipper@gmail.com

1 Department of Computer Science, Ben-Gurion University, Beer Sheva 84105, Israel
2 Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA 19104-6021, USA
regression, asking whether gradient boosting might improve a strong(er) learner in the form of a symbolic regressor. We answer in the affirmative, demonstrating that improved results can be readily obtained, at relatively little added cost.

In the next section we describe our method and discuss related work. Section 3 presents the experimental setup and our results, followed by concluding remarks in Sect. 4.

2 Symbolic-regression boosting (SyRBo)

For our experiments we used the popular scikit-learn Python package [14, 16] due to its superb ability to handle much of the tedious desiderata of machine learning coding and experimentation. We then chose the GPLearn package [5], which implements tree-based genetic programming (GP) symbolic regression, is relatively fast, and—importantly—interfaces seamlessly with scikit-learn.

The main idea behind SyRBo is simple: we replace the boosted weak learner of gradient boosting (typically a decision tree) with a (possibly) strong learner, specifically, a GP-based symbolic regressor.

Algorithm 1 provides the pseudocode (the code is available at https://github.com/moshesipper). SyRBo receives the number of boosting stages as a parameter (one might consider the actual number of stages to be one less, as the first stage performs the initial prediction). Fitting a model to data is done in a standard gradient-boosting manner, through successive stages, where each stage fits a learner to the pseudo-residuals of the previous stage; prediction is performed by summing up all learner predictions. The only change involves the learners themselves, which are not decision trees but rather symbolic regressors, evolved by calling the SymbolicRegressor function with the given population size and generation count (both set to 200). The function set used by SymbolicRegressor is given in Table 1. To facilitate the symbolic regressor’s handling of diverse features scales, the dataset rows undergo L2 normalization (i.e., the feature values in a row have a unit L2 norm).
Table 1 Function set used by SymbolicRegressor

| Function | Arity | Description                       |
|----------|-------|-----------------------------------|
| add      | 2     | Addition                          |
| sub      | 2     | Subtraction                       |
| mul      | 2     | Multiplication                    |
| div      | 2     | Protected division (near-zero denominator returns 1) |
| sqrt     | 1     | Protected square root (uses absolute value of argument) |
| log      | 1     | Protected log (uses absolute value of argument, near-zero argument returns 0) |
| abs      | 1     | Absolute value                    |
| neg      | 1     | Negative                          |
| inv      | 1     | Protected inverse (near-zero argument returns 0) |
| max      | 2     | Maximum                           |
| min      | 2     | Minimum                           |
| if3      | 3     | if3(x1, x2, x3) returns x2 if x1 ≥ 0 else returns x3 |
| if4      | 4     | if4(x1, x2, x3, x4) returns x3 if x1 ≥ x2 else returns x4 |

Algorithm 1 SyRBo.

Input:

stages ← number of boosting stages
population_size ← 200
generations ← 200

1: function INIT(stages, population_size, generations)
2: Initialize an empty SyRBo object with given parameters
3: boosters = {}  # Initialize an empty list of boosters

4: function FIT(X, y)  # X: training inputs, y: target values
5: for stage ← 0 to stages-1 do
6:   gp = SymbolicRegressor(population_size, generations)  # Initialize a GP regressor
7:   gp.fit(X, y)  # Fit regressor to (training) data
8:   Add gp to boosters  # Add the fitted GP regressor to the list of boosters
9: prediction = y - gp.predict(X)  # Compute pseudo-residuals

10: function PREDICT(X)  # X: inputs
11: prediction = 0  # Vector of zeros whose length equals number of instances in dataset
12: for i ← 0 to stages-1 do
13:   prediction = prediction + boosters[i].predict(X)
14: Return prediction

We note that our aim herein was to demonstrate SyRBo’s being an add-on that can be added to any symbolic regressor. As such, this paper is not about symbolic regression per se, but about performance benefits to be gained if one is using it. We thus contented ourselves with the standard function set of gplearn (adding only 2
conditionals), with all other parameters set to defaults (except for population size and generation count).

Regarding previous work, it would seem that by and large the emphasis in boosting techniques has been on weak learners, typically decision trees. Works using strong learners in the context of boosting employed mainly AdaBoost-like [3] boosting. Modest success was attained by [2, 6]. Wickramaratna et al. [17] showed that boosting a strong learner with AdaBoost may, in fact, contribute to performance degradation. Within the domain of GP, AdaBoost-like boosting of dataset sample weights has been used with some success [7, 8, 10, 13]. Perhaps closest to our work is that of [11], who presented an interesting iterative approach, Sequential Symbolic Regression, wherein each iteration applies a transformation based on a geometric semantic crossover operator. In contrast, our work is based on gradient boosting, is more generic in that it can work with any form of symbolic regression, and is also easier to code and apply to any extant project.

3 Experimental setup and results

Can this (fairly) simple gradient boosting-like setup improve symbolic regression? We tested SyRBo on regression datasets from the PMLB repository [12], using our cluster of Intel® Xeon® E5-2650L servers. Of the 120 datasets we selected the 98 with 3000 instances or less. Figure 1 shows a “bird’s-eye view” of the datasets.

The pseudo-code for the experimental setup is given in Algorithm 2. For each dataset we performed 30 replicate runs, with 5-fold cross validation per replicate. SyRBo and SymbolicRegressor (with equal population size and generations) were trained on 4 folds and tested on the left-out test fold.

![Fig. 1 A “bird’s-eye view” of the 98 datasets used in this study: number of instances (left) and number of features (right)](image)
(Algorithm 2) Experimental setup.

**Input:**

- `dataset ←` dataset to be used
- `algorithms ← {SyRBo, SymbolicRegressor}`

**Output:**

Performance measures (over test sets)

1: for `rep ← 1 to 30` do
2: Shuffle `dataset` and generate 5 folds
3: for `fold ← 1 to 5` do
4: Split `dataset` into *training* and *test* sets according to `fold`
5: for `alg in algorithms` do
6: Use `alg` to fit a *model* to *training* set
7: Test resultant *model* on *test* set

For each of the 98 datasets we recorded the mean absolute error attained per algorithm over each of the 30 replicate runs, per each of the 5 test folds (i.e., following training). We ran 4 separate experiments, over all 98 datasets, with number of stages equal to 2, 3, 4, and 5, respectively.

Table 2 shows a summary of our results (detailed results can be found in the Appendix). For each dataset we computed the median of the test scores of all 30 replicates, with 5 folds per replicate (a total of 150 test-score values). A win for SyRBo was then a better (lower) median value than SymbolicRegressor’s. To assess whether a win for a specific dataset was significant or not, we performed a 10,000-round permutation test, comparing the scores of SyRBo with SymbolicRegressor; if the p-value was < 0.05 the win was considered significant, else it was not (in which case SyRBo was at least not performing worse than SymbolicRegressor). In addition, when SyRBo “lost” to SymbolicRegressor we performed a 10,000-round permutation test, comparing the two algorithms’ scores; if the p-value was ≥ 0.05 the loss was considered insignificant.

| Datasets | Stages | Wins | Significant | Losses | Insignificant |
|----------|--------|------|-------------|--------|---------------|
| 98       | 2      | 78   | 48          | 20     | 16            |
| 98       | 3      | 83   | 63          | 15     | 13            |
| 98       | 4      | 84   | 71          | 14     | 12            |
| 98       | 5      | 87   | 70          | 11     | 9             |
As seen in the table, statistically significant improvements can often be attained, and, moreover, rarely does SyRBo result in statistically significant worse results. Using SyRBo is thus a good bet, and, furthermore, it is easily coded and the added computational cost is not high.

### 4 Concluding remarks

We presented SyRBo, a gradient boosting-style algorithm, wherein the decision tree is replaced by a symbolic regressor. Testing the merits of our new method we showed that symbolic regression results can be consistently improved. Moreover, as can be seen in Algorithm 1, coding SyRBo on top of any symbolic regressor is straightforward, and the added cost is simply a few more evolutionary rounds. SyRBo is essentially a simple add-on that can be readily added to a symbolic regressor, often with beneficial results.

There are a number of avenues we can offer for future exploration:

- Add known boosting tricks of the trade, such as a learning rate and dynamic early stopping (similar to XGBRegressor’s ‘early stopping rounds’ parameter).
- Our focus herein was on regression. It would seem worthwhile to examine SyRBo for classification.
- We used a rather basic symbolic regressor. The GP literature is rife with many other types of regressors, which might be used in conjunction with SyRBo. More generally, other types of GP might offer productive ways to evolve programs that might serve as strong learners.
- Comparison to non-symbolic-regressor methods.
- While we focused on gradient boosting, other types of boosting techniques might be examined as to whether they might be a good fit for SyRBo.

### Appendix: detailed results

The results of all experiments over all datasets are given in Tables 3, 4, 5, and 6 for number of stages equal to 2, 3, 4, and 5, respectively. As noted in Sect. 3, for each of the 98 datasets we recorded the mean absolute error attained per algorithm over each of the 30 replicate runs, per each of the 5 test folds. We then computed the median of these scores, which are presented under ‘mean absolute error’ in the tables. Under ‘pval’ we show the results of the 10,000-round permutation tests between the scores of SyRBo and SymbolicRegressor, with a ‘!’ denoting a significant win for SyRBo and a ‘=’ denoting an insignificant loss for SyRBo. Under ‘run times’ we show the median run times for SyRBo and SymbolicRegressor. ‘SR’ denotes SymbolicRegressor.
| Dataset                  | Mean absolute error and pval                  | Run times               |
|--------------------------|----------------------------------------------|-------------------------|
| 1027_ESL                 | SyRBo: 1.02, SR: 1.04, pval: 9.5E-02          | SyRBo: 59.89s, SR: 31.17s |
| 1028_SWD                 | SyRBo: 0.61, SR: 0.62, pval: 2.2E-01          | SyRBo: 46.61s, SR: 23.87s |
| 1029_LEV                 | SyRBo: 0.62, SR: 0.65, pval: 0.0E+00 !        | SyRBo: 54.6s, SR: 26.46s   |
| 1030 ERA                 | SyRBo: 1.45, SR: 1.46, pval: 3.9E-01          | SyRBo: 61.8s, SR: 30.06s   |
| 1089_USCrime             | SyRBo: 25.59, SR: 27.52, pval: 2.3E-01        | SyRBo: 123.68s, SR: 73.81s |
| 1096_FacultySalaries     | SyRBo: 3.56, SR: 3.63, pval: 6.2E-01          | SyRBo: 61.05s, SR: 36.83s   |
| 192_vineyard             | SR: 2.44, SyRBo: 2.52, pval: 2.1E-01 =       | SyRBo: 66.24s, SR: 37.82s   |
| 195_auto_price           | SR: 1990.78, SyRBo: 2090.58, pval: 8.9E-02 = | SyRBo: 316.35s, SR: 195.05s |
| 207_autoPrice            | SR: 1955.88, SyRBo: 2093.54, pval: 2.8E-02   | SyRBo: 424.44s, SR: 176.55s |
| 210_cloud                | SR: 0.51, SyRBo: 0.52, pval: 4.1E-01 =       | SyRBo: 44.47s, SR: 22.64s   |
| 228_elusage              | SyRBo: 12.97, SR: 13.72, pval: 3.2E-01        | SyRBo: 126.28s, SR: 89.94s   |
| 229_pwLinear             | SyRBo: 1.56, SR: 1.66, pval: 1.2E-02 !        | SyRBo: 65.29s, SR: 35.59s   |
| 230_machine_cpu          | SyRBo: 43.04, SR: 45.95, pval: 1.2E-01        | SyRBo: 166.36s, SR: 92.74s   |
| 4544_GeographicalOriginalofMusic | SR: 0.49, SyRBo: 0.5, pval: 8.2E-02 =       | SyRBo: 62.38s, SR: 36.86s   |
| 485_analcatdata_vehicle  | SyRBo: 151.48, SR: 179.22, pval: 1.5E-03 !   | SyRBo: 220.1s, SR: 135.99s  |
| 505_tecator              | SR: 5.17, SyRBo: 5.83, pval: 1.0E-02          | SyRBo: 92.7s, SR: 51.74s    |
| 519_vinnie               | SR: 1.27, SyRBo: 1.31, pval: 1.3E-02          | SyRBo: 64.34s, SR: 35.1s    |
| 522_pm10                 | SyRBo: 0.68, SR: 0.69, pval: 2.1E-01          | SyRBo: 44.3s, SR: 22.12s    |
| 523_analcatdata_neavote  | SR: 0.51, SyRBo: 0.52, pval: 7.6E-01 =       | SyRBo: 59.63s, SR: 35.95s   |
| 527_analcatdata_election2000 | SR: 38724.65, SyRBo: 39617.02, pval: 7.8E-01 = | SyRBo: 562.68s, SR: 230.48s  |
| 542_pollution            | SR: 176.33, SyRBo: 185.12, pval: 5.9E-01 =   | SyRBo: 215.87s, SR: 141.38s  |
| 547_no2                  | SR: 0.58, SyRBo: 0.58, pval: 8.7E-01 =       | SyRBo: 57.51s, SR: 29.23s   |
| 556_analcatdata_apnea2   | SR: 825.75, SyRBo: 840.89, pval: 7.3E-01 =   | SyRBo: 164.27s, SR: 68.58s   |
### Table 3 (continued)

| Dataset                  | Mean absolute error and pval  | Run times               |
|--------------------------|-------------------------------|-------------------------|
| 557_analcatdata_apnea1   | SyRBo: 828.37, SR: 844.69, pval: 7.1E-01 | SyRBo: 130.15s, SR: 50.06s |
| 560_bodyfat              | SR: 4.21, SyRBo: 4.64, pval: 4.0E-04 | SyRBo: 59.26s, SR: 33.41s |
| 561_cpu                  | SyRBo: 30.99, SR: 34.95, pval: 8.4E-02 | SyRBo: 115.22s, SR: 77.02s |
| 579_fri_c0_250_5         | SyRBo: 0.42, SR: 0.45, pval: 0.0E+00 ! | SyRBo: 53.18s, SR: 26.01s |
| 581_fri_c3_500_25        | SyRBo: 0.71, SR: 0.72, pval: 2.3E-03 ! | SyRBo: 43.32s, SR: 21.5s |
| 582_fri_c1_500_25        | SyRBo: 0.7, SR: 0.72, pval: 3.8E-02 ! | SyRBo: 53.82s, SR: 26.82s |
| 583_fri_c1_1000_50       | SyRBo: 0.73, SR: 0.74, pval: 1.5E-02 ! | SyRBo: 61.81s, SR: 30.2s |
| 584_fri_c4_500_25        | SyRBo: 0.7, SR: 0.71, pval: 3.6E-02 ! | SyRBo: 53.68s, SR: 26.69s |
| 586_fri_c3_1000_25       | SyRBo: 0.69, SR: 0.71, pval: 6.7E-03 ! | SyRBo: 54.44s, SR: 27.13s |
| 588_fri_c4_1000_100      | SyRBo: 0.73, SR: 0.73, pval: 8.7E-01 | SyRBo: 41.83s, SR: 21.16s |
| 589_fri_c2_1000_25       | SyRBo: 0.7, SR: 0.71, pval: 2.2E-02 ! | SyRBo: 53.3s, SR: 26.58s |
| 590_fri_c0_1000_50       | SyRBo: 0.39, SR: 0.4, pval: 1.0E-02 ! | SyRBo: 55.05s, SR: 28.77s |
| 591_fri_c1_100_10        | SyRBo: 0.71, SR: 0.73, pval: 1.6E-01 | SyRBo: 55.02s, SR: 26.14s |
| 592_fri_c4_1000_25       | SyRBo: 0.72, SR: 0.72, pval: 2.2E-01 | SyRBo: 52.62s, SR: 26.27s |
| 593_fri_c1_1000_10       | SyRBo: 0.65, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 54.59s, SR: 26.25s |
| 594_fri_c2_100_5         | SyRBo: 0.64, SR: 0.68, pval: 1.9E-02 ! | SyRBo: 55.12s, SR: 27.14s |
| 595_fri_c0_1000_10       | SyRBo: 0.39, SR: 0.44, pval: 0.0E+00 ! | SyRBo: 54.06s, SR: 27.02s |
| 596_fri_c2_250_5         | SyRBo: 0.63, SR: 0.68, pval: 0.0E+00 ! | SyRBo: 45.99s, SR: 22.61s |
| 597_fri_c2_500_5         | SyRBo: 0.61, SR: 0.67, pval: 0.0E+00 ! | SyRBo: 55.13s, SR: 26.54s |
| 598_fri_c0_1000_25       | SyRBo: 0.41, SR: 0.43, pval: 2.0E-04 ! | SyRBo: 53.61s, SR: 27.51s |
| 599_fri_c2_1000_5        | SyRBo: 0.57, SR: 0.67, pval: 0.0E+00 ! | SyRBo: 55.04s, SR: 26.24s |
| 601_fri_c1_250_5         | SyRBo: 0.58, SR: 0.65, pval: 0.0E+00 ! | SyRBo: 55.37s, SR: 26.62s |
| 602_fri_c3_250_10        | SyRBo: 0.69, SR: 0.72, pval: 7.3E-03 ! | SyRBo: 52.95s, SR: 25.94s |

Springer
| Dataset         | Mean absolute error and pval | Run times          |
|-----------------|------------------------------|--------------------|
| SyRBo: 0.4, SR: 0.4, pval: 9.6E-01 | SyRBo: 42.83s, SR: 22.17s |
| SyRBo: 0.68, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 43.33s, SR: 21.33s |
| SyRBo: 0.69, SR: 0.69, pval: 7.7E-01 ! | SyRBo: 44.26s, SR: 21.92s |
| SyRBo: 0.64, SR: 0.67, pval: 0.0E+00 ! | SyRBo: 54.98s, SR: 26.73s |
| SyRBo: 0.72, SR: 0.73, pval: 1.7E-01 ! | SyRBo: 43.72s, SR: 21.98s |
| SyRBo: 0.66, SR: 0.7, pval: 0.0E+00 ! | SyRBo: 57.53s, SR: 28.12s |
| SyRBo: 0.41, SR: 0.44, pval: 0.0E+00 ! | SyRBo: 42.59s, SR: 20.94s |
| SyRBo: 0.64, SR: 0.66, pval: 3.1E-01 ! | SyRBo: 56.53s, SR: 27.55s |
| SyRBo: 0.59, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 56.08s, SR: 26.31s |
| SyRBo: 0.59, SR: 0.62, pval: 0.0E+00 ! | SyRBo: 54.03s, SR: 26.4s |
| SyRBo: 0.67, SR: 0.7, pval: 5.6E-03 ! | SyRBo: 51.54s, SR: 25.62s |
| SyRBo: 0.74, SR: 0.74, pval: 9.6E-01 ! | SyRBo: 52.4s, SR: 26.34s |
| SyRBo: 0.58, SR: 0.65, pval: 0.0E+00 ! | SyRBo: 54.41s, SR: 26.8s |
| SyRBo: 0.72, SR: 0.73, pval: 1.0E-01 ! | SyRBo: 51.91s, SR: 26.01s |
| SyRBo: 0.72, SR: 0.74, pval: 1.5E-03 ! | SyRBo: 42.36s, SR: 21.08s |
| SyRBo: 0.44, SR: 0.47, pval: 2.9E-02 ! | SyRBo: 43.37s, SR: 21.4s |
| SyRBo: 0.72, SR: 0.73, pval: 1.4E-01 ! | SyRBo: 44.87s, SR: 22.63s |
| SyRBo: 0.64, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 53.96s, SR: 26.17s |
| SR: 0.46, SyRBo: 0.46, pval: 9.3E-01 = | SyRBo: 43.75s, SR: 21.15s |
| SyRBo: 0.73, SR: 0.73, pval: 2.6E-01 ! | SyRBo: 42.0s, SR: 21.23s |
| SyRBo: 0.63, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 52.34s, SR: 25.3s |
| SyRBo: 0.59, SR: 0.66, pval: 0.0E+00 ! | SyRBo: 56.7s, SR: 27.28s |
| SyRBo: 0.6, SR: 0.68, pval: 0.0E+00 ! | SyRBo: 55.86s, SR: 26.53s |
Table 3 (continued)

| Dataset       | Mean absolute error and pval | Run times |
|---------------|------------------------------|-----------|
| 633_fri_c0_500_25 | SyRBo: 0.4, SR: 0.42, pval: 0.0E+00 | SyRBo: 51.78s, SR: 26.37s |
| 634_fri_c2_100_10 | SyRBo: 0.68, SR: 0.69, pval: 9.0E-01 | SyRBo: 54.48s, SR: 26.52s |
| 635_fri_c0_250_10 | SyRBo: 0.44, SR: 0.52, pval: 0.0E+00 | SyRBo: 43.65s, SR: 21.72s |
| 637_fri_c1_500_50 | SyRBo: 0.75, SR: 0.76, pval: 1.1E-01 | SyRBo: 44.14s, SR: 22.12s |
| 641_fri_c1_500_10 | SyRBo: 0.67, SR: 0.74, pval: 0.0E+00 | SyRBo: 54.06s, SR: 26.39s |
| 643_fri_c2_500_25 | SyRBo: 0.74, SR: 0.75, pval: 1.8E-01 | SyRBo: 42.11s, SR: 20.85s |
| 644_fri_c4_250_25 | SyRBo: 0.73, SR: 0.74, pval: 4.2E-01 | SyRBo: 41.93s, SR: 20.77s |
| 645_fri_c3_500_50 | SyRBo: 0.7, SR: 0.7, pval: 8.6E-01 | SyRBo: 41.84s, SR: 20.81s |
| 646_fri_c3_500_10 | SyRBo: 0.64, SR: 0.68, pval: 1.0E-04 | SyRBo: 52.88s, SR: 25.77s |
| 647_fri_c1_250_10 | SyRBo: 0.65, SR: 0.73, pval: 0.0E+00 | SyRBo: 50.08s, SR: 25.95s |
| 648_fri_c1_250_50 | SyRBo: 0.72, SR: 0.73, pval: 3.8E-01 | SyRBo: 52.98s, SR: 26.88s |
| 649_fri_c0_500_5 | SyRBo: 0.4, SR: 0.46, pval: 0.0E+00 | SyRBo: 52.84s, SR: 25.93s |
| 650_fri_c0_500_50 | SyRBo: 0.38, SR: 0.39, pval: 2.7E-02 | SyRBo: 52.92s, SR: 27.49s |
| 651_fri_c0_100_25 | SyRBo: 0.52, SR: 0.53, pval: 4.9E-01 | SyRBo: 51.51s, SR: 25.99s |
| 653_fri_c0_250_25 | SyRBo: 0.4, SR: 0.41, pval: 8.7E-03 | SyRBo: 53.46s, SR: 27.13s |
| 654_fri_c0_500_10 | SyRBo: 0.42, SR: 0.46, pval: 0.0E+00 | SyRBo: 52.64s, SR: 26.29s |
| 656_fri_c1_100_5 | SyRBo: 0.57, SR: 0.66, pval: 0.0E+00 | SyRBo: 55.37s, SR: 27.63s |
| 657_fri_c2_250_10 | SyRBo: 0.63, SR: 0.7, pval: 0.0E+00 | SyRBo: 53.59s, SR: 25.96s |
| 658_fri_c3_250_25 | SyRBo: 0.73, SR: 0.75, pval: 1.6E-01 | SyRBo: 51.63s, SR: 25.6s |
| 659_sleuth_ex1714 | SyRBo: 6745.19, SR: 8208.12, pval: 2.2E-02 | SyRBo: 366.9s, SR: 179.81s |
| 663_rabe_266 | SR: 19.76, SyRBo: 20.19, pval: 4.7E-01 | SyRBo: 95.03s, SR: 54.36s |
| 665_sleuth_case2002 | SR: 5.08, SyRBo: 5.33, pval: 6.8E-02 | SyRBo: 53.61s, SR: 31.3s |
| 666_rmftsa_ladata | SR: 1.64, SyRBo: 1.64, pval: 9.2E-01 | SyRBo: 45.69s, SR: 25.33s |
| Dataset                        | Mean absolute error and pval | Run times         |
|-------------------------------|------------------------------|-------------------|
| 678_visualizing_environmental | SyRBo: 2.46, SR: 2.51, pval: 4.4E-01 | SyRBo: 50.56s, SR: 28.09s |
| 687_sleuth_ex1605             | SR: 13.26, SyRBo: 14.33, pval: 5.4E-02 = | SyRBo: 90.32s, SR: 51.23s |
| 690_visualizing_galaxy        | SyRBo: 259.23, SR: 461.52, pval: 0.0E+00 ! | SyRBo: 263.2s, SR: 137.71s |
| 695_chatfield_4               | SR: 17.47, SyRBo: 17.81, pval: 4.0E-01 = | SyRBo: 112.3s, SR: 57.87s |
| 706_sleuth_case1202           | SR: 48.76, SyRBo: 52.21, pval: 8.1E-02 = | SyRBo: 120.17s, SR: 77.54s |
| 712_chscase_geyser1           | SyRBo: 8.3, SR: 9.0, pval: 1.0E-04 ! | SyRBo: 71.33s, SR: 47.21s |
| Dataset                  | Mean absolute error and pval | Run times        |
|-------------------------|------------------------------|------------------|
| 1027_ESL                | SyRBo: 1.01, SR: 1.04, pval: 1.1E-02 ! | SyRBo: 70.24s, SR: 24.65s |
| 1028_SWD                | SyRBo: 0.61, SR: 0.62, pval: 1.7E-01 | SyRBo: 70.79s, SR: 25.03s |
| 1029_LEV                | SyRBo: 0.62, SR: 0.64, pval: 0.0E+00 ! | SyRBo: 69.2s, SR: 22.65s |
| 1030_ERA                | SyRBo: 1.43, SR: 1.46, pval: 2.0E-02 ! | SyRBo: 72.45s, SR: 24.73s |
| 1089_USCrime            | SyRBo: 25.31, SR: 27.01, pval: 1.8E-01 | SyRBo: 134.46s, SR: 57.85s |
| 1096_FacultySalaries    | SR: 3.57, SyRBo: 3.6, pval: 7.5E-01 = | SyRBo: 74.28s, SR: 30.96s |
| 192_vineyard            | SR: 2.42, SyRBo: 2.54, pval: 5.9E-02 = | SyRBo: 72.9s, SR: 30.64s |
| 195_auto_price          | SyRBo: 1955.32, SR: 2049.73, pval: 2.0E-01 | SyRBo: 558.45s, SR: 164.11s |
| 207_autoPrice           | SyRBo: 1945.66, SR: 1968.41, pval: 8.6E-01 | SyRBo: 465.0s, SR: 133.33s |
| 210_cloud               | SyRBo: 0.5, SR: 0.51, pval: 5.9E-01 | SyRBo: 68.09s, SR: 23.48s |
| 228_elusage             | SyRBo: 12.68, SR: 14.45, pval: 2.9E-03 ! | SyRBo: 127.75s, SR: 64.49s |
| 229_pwLinear            | SyRBo: 1.49, SR: 1.63, pval: 1.9E-03 ! | SyRBo: 73.73s, SR: 27.64s |
| 230_machine_cpu         | SyRBo: 40.74, SR: 44.23, pval: 1.2E-01 | SyRBo: 244.09s, SR: 101.61s |
| 4544_GeographicalOriginalofMusic | SyRBo: 0.49, SR: 0.49, pval: 9.7E-01 | SyRBo: 91.17s, SR: 38.4s |
| 485_analcatdata_vehicle | SyRBo: 155.87, SR: 184.07, pval: 5.8E-03 ! | SyRBo: 363.3s, SR: 144.26s |
| 505_tecator             | SR: 5.02, SyRBo: 5.35, pval: 1.8E-01 = | SyRBo: 140.1s, SR: 60.43s |
| 519_vinnie              | SR: 1.26, SyRBo: 1.27, pval: 5.5E-01 = | SyRBo: 92.19s, SR: 35.46s |
| 522_pm10                | SyRBo: 0.67, SR: 0.69, pval: 3.1E-02 ! | SyRBo: 81.01s, SR: 27.42s |
| 523_analcatdata_neavote | SyRBo: 0.49, SR: 0.5, pval: 5.4E-01 | SyRBo: 91.61s, SR: 39.44s |
| 527_analcatdata_election2000 | SR: 42367.13, SyRBo: 42794.6, pval: 8.7E-01 = | SyRBo: 782.84s, SR: 187.09s |
| 542_pollution           | SR: 179.19, SyRBo: 183.2, pval: 6.9E-01 = | SyRBo: 323.15s, SR: 137.38s |
| 547_no2                 | SyRBo: 0.57, SR: 0.59, pval: 8.1E-03 ! | SyRBo: 84.6s, SR: 28.94s |
| 556_analcatdata_apnea2  | SR: 838.3, SyRBo: 841.71, pval: 9.2E-01 = | SyRBo: 253.91s, SR: 91.68s |
| Dataset | Mean absolute error and pval | Run times |
|---------|-----------------------------|-----------|
| 557_analcatdata_apnea1 | SR: 838.25, SyRBo: 871.41, pval: 4.8E-01 | SyRBo: 209.53s, SR: 55.88s |
| 560_bodyfat | SR: 4.23, SyRBo: 4.34, pval: 2.1E-01 | SyRBo: 104.88s, SR: 41.81s |
| 561_cpu | SyRBo: 30.41, SR: 33.93, pval: 1.8E-01 | SyRBo: 208.96s, SR: 89.89s |
| 579_fri_c0_250_5 | SyRBo: 0.4, SR: 0.45, pval: 3.1E-03 | SyRBo: 79.92s, SR: 26.46s |
| 581_fri_c3_500_25 | SyRBo: 0.7, SR: 0.72, pval: 1.7E-01 | SyRBo: 79.28s, SR: 25.77s |
| 582_fri_c1_500_25 | SyRBo: 0.68, SR: 0.72, pval: 0.0E+00 | SyRBo: 77.19s, SR: 25.52s |
| 583_fri_c1_1000_50 | SyRBo: 0.72, SR: 0.74, pval: 0.0E+00 | SyRBo: 75.06s, SR: 25.16s |
| 584_fri_c4_500_25 | SyRBo: 0.68, SR: 0.71, pval: 0.0E+00 | SyRBo: 75.45s, SR: 24.92s |
| 586_fri_c3_1000_25 | SyRBo: 0.68, SR: 0.7, pval: 0.0E+00 | SyRBo: 79.06s, SR: 26.17s |
| 588_fri_c4_1000_100 | SyRBo: 0.72, SR: 0.73, pval: 1.7E-01 | SyRBo: 78.02s, SR: 26.38s |
| 589_fri_c2_1000_25 | SyRBo: 0.68, SR: 0.71, pval: 0.0E+00 | SyRBo: 79.93s, SR: 26.46s |
| 590_fri_c0_1000_50 | SyRBo: 0.37, SR: 0.41, pval: 0.0E+00 | SyRBo: 80.78s, SR: 28.43s |
| 591_fri_c1_100_10 | SyRBo: 0.71, SR: 0.74, pval: 3.9E-02 | SyRBo: 79.19s, SR: 25.39s |
| 592_fri_c4_1000_25 | SyRBo: 0.7, SR: 0.72, pval: 1.0E-04 | SyRBo: 81.93s, SR: 27.1s |
| 593_fri_c1_1000_10 | SyRBo: 0.61, SR: 0.71, pval: 0.0E+00 | SyRBo: 82.54s, SR: 26.59s |
| 594_fri_c2_100_5 | SyRBo: 0.64, SR: 0.71, pval: 0.0E+00 | SyRBo: 80.74s, SR: 26.65s |
| 595_fri_c0_1000_10 | SyRBo: 0.35, SR: 0.44, pval: 0.0E+00 | SyRBo: 82.43s, SR: 27.34s |
| 596_fri_c2_250_5 | SyRBo: 0.6, SR: 0.68, pval: 0.0E+00 | SyRBo: 80.88s, SR: 26.18s |
| 597_fri_c2_500_5 | SyRBo: 0.58, SR: 0.68, pval: 0.0E+00 | SyRBo: 81.55s, SR: 26.48s |
| 598_fri_c0_1000_25 | SyRBo: 0.36, SR: 0.43, pval: 0.0E+00 | SyRBo: 80.75s, SR: 27.68s |
| 599_fri_c2_250_5 | SyRBo: 0.65, SR: 0.66, pval: 0.0E+00 | SyRBo: 83.39s, SR: 26.86s |
| 601_fri_c1_250_5 | SyRBo: 0.56, SR: 0.67, pval: 0.0E+00 | SyRBo: 81.61s, SR: 26.23s |
| 602_fri_c3_250_10 | SyRBo: 0.68, SR: 0.72, pval: 0.0E+00 | SyRBo: 78.76s, SR: 25.62s |
| Dataset          | Mean absolute error and pval | Run times          |
|------------------|------------------------------|--------------------|
| 603_fri_c0_250_50 | SyRBo: 0.39, SR: 0.41, pval: 7.3E-03 | SyRBo: 77.87s, SR: 27.07s |
| 604_fri_c4_500_10 | SyRBo: 0.66, SR: 0.71, pval: 0.0E+00 | SyRBo: 79.58s, SR: 26.2s |
| 605_fri_c2_250_25 | SyRBo: 0.69, SR: 0.7, pval: 5.3E-01 | SyRBo: 79.46s, SR: 26.31s |
| 606_fri_c2_1000_10 | SyRBo: 0.59, SR: 0.67, pval: 0.0E+00 | SyRBo: 82.15s, SR: 26.38s |
| 607_fri_c4_1000_50 | SyRBo: 0.71, SR: 0.73, pval: 1.4E-02 | SyRBo: 78.32s, SR: 26.14s |
| 608_fri_c3_1000_10 | SyRBo: 0.61, SR: 0.7, pval: 0.0E+00 | SyRBo: 82.63s, SR: 26.62s |
| 609_fri_c0_1000_5 | SyRBo: 0.37, SR: 0.44, pval: 0.0E+00 | SyRBo: 79.77s, SR: 25.95s |
| 611_fri_c3_100_5 | SyRBo: 0.61, SR: 0.67, pval: 8.0E-03 | SyRBo: 84.03s, SR: 28.26s |
| 612_fri_c1_1000_5 | SyRBo: 0.55, SR: 0.69, pval: 0.0E+00 | SyRBo: 82.23s, SR: 26.03s |
| 613_fri_c3_250_5 | SyRBo: 0.58, SR: 0.64, pval: 0.0E+00 | SyRBo: 79.23s, SR: 25.57s |
| 615_fri_c4_250_10 | SyRBo: 0.66, SR: 0.7, pval: 0.0E+00 | SyRBo: 79.95s, SR: 26.18s |
| 616_fri_c4_500_50 | SyRBo: 0.74, SR: 0.74, pval: 6.5E-01 | SyRBo: 80.63s, SR: 26.95s |
| 617_fri_c3_500_5 | SyRBo: 0.57, SR: 0.65, pval: 0.0E+00 | SyRBo: 80.92s, SR: 26.7s |
| 618_fri_c3_1000_50 | SyRBo: 0.71, SR: 0.73, pval: 4.0E-04 | SyRBo: 78.7s, SR: 26.25s |
| 620_fri_c1_1000_25 | SyRBo: 0.7, SR: 0.73, pval: 0.0E+00 | SyRBo: 79.9s, SR: 26.4s |
| 621_fri_c0_100_10 | SyRBo: 0.41, SR: 0.45, pval: 8.0E-03 | SyRBo: 78.91s, SR: 25.96s |
| 622_fri_c2_1000_50 | SyRBo: 0.71, SR: 0.73, pval: 3.0E-04 | SyRBo: 66.25s, SR: 22.17s |
| 623_fri_c4_1000_10 | SyRBo: 0.62, SR: 0.68, pval: 0.0E+00 | SyRBo: 81.44s, SR: 26.48s |
| 624_fri_c0_100_5 | SyRBo: 0.42, SR: 0.47, pval: 0.0E+00 | SyRBo: 78.17s, SR: 25.11s |
| 626_fri_c2_500_50 | SyRBo: 0.71, SR: 0.72, pval: 4.6E-01 | SyRBo: 78.22s, SR: 26.39s |
| 627_fri_c2_500_10 | SyRBo: 0.61, SR: 0.69, pval: 0.0E+00 | SyRBo: 85.52s, SR: 27.54s |
| 628_fri_c3_1000_5 | SyRBo: 0.59, SR: 0.66, pval: 0.0E+00 | SyRBo: 86.2s, SR: 27.97s |
| 631_fri_c1_500_5 | SyRBo: 0.56, SR: 0.66, pval: 0.0E+00 | SyRBo: 82.42s, SR: 26.2s |
| Dataset       | Mean absolute error and pval | Run times            |
|--------------|-----------------------------|----------------------|
| 633_fri_c0_500_25 | SyRBo: 0.37, SR: 0.43, pval: 0.0E+00 ! | SyRBo: 78.32s, SR: 26.68s |
| 634_fri_c2_100_10 | SyRBo: 0.64, SR: 0.68, pval: 9.0E-04 ! | SyRBo: 80.8s, SR: 26.22s |
| 635_fri_c0_250_10 | SyRBo: 0.39, SR: 0.52, pval: 0.0E+00 ! | SyRBo: 61.06s, SR: 20.16s |
| 637_fri_c1_500_50 | SyRBo: 0.76, SR: 0.76, pval: 7.5E-01 | SyRBo: 79.6s, SR: 26.58s |
| 641_fri_c1_500_10 | SyRBo: 0.62, SR: 0.73, pval: 0.0E+00 ! | SyRBo: 81.02s, SR: 26.14s |
| 643_fri_c2_500_25 | SyRBo: 0.74, SR: 0.76, pval: 2.1E-02 ! | SyRBo: 78.8s, SR: 26.08s |
| 644_fri_c4_250_25 | SyRBo: 0.72, SR: 0.74, pval: 2.0E-01 | SyRBo: 80.38s, SR: 26.48s |
| 645_fri_c3_500_50 | SyRBo: 0.7, SR: 0.71, pval: 3.6E-01 | SyRBo: 80.26s, SR: 26.57s |
| 646_fri_c3_500_10 | SyRBo: 0.63, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 80.1s, SR: 26.1s |
| 647_fri_c1_250_100 | SyRBo: 0.64, SR: 0.73, pval: 0.0E+00 ! | SyRBo: 80.7s, SR: 26.13s |
| 648_fri_c1_250_50 | SyRBo: 0.74, SR: 0.74, pval: 8.4E-01 | SyRBo: 77.4s, SR: 26.33s |
| 649_fri_c0_500_5 | SyRBo: 0.37, SR: 0.46, pval: 0.0E+00 ! | SyRBo: 76.0s, SR: 24.91s |
| 650_fri_c0_500_50 | SyRBo: 0.37, SR: 0.4, pval: 0.0E+00 ! | SyRBo: 82.74s, SR: 28.93s |
| 651_fri_c0_100_25 | SyRBo: 0.5, SR: 0.53, pval: 1.7E-02 ! | SyRBo: 77.58s, SR: 26.06s |
| 653_fri_c0_250_25 | SyRBo: 0.37, SR: 0.41, pval: 0.0E+00 ! | SyRBo: 62.56s, SR: 21.2s |
| 654_fri_c0_500_10 | SyRBo: 0.37, SR: 0.46, pval: 0.0E+00 ! | SyRBo: 63.3s, SR: 20.95s |
| 656_fri_c1_100_5 | SyRBo: 0.55, SR: 0.65, pval: 0.0E+00 ! | SyRBo: 87.52s, SR: 28.95s |
| 657_fri_c2_250_10 | SyRBo: 0.64, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 84.08s, SR: 27.15s |
| 658_fri_c3_250_25 | SyRBo: 0.73, SR: 0.74, pval: 1.4E-01 | SyRBo: 77.71s, SR: 25.61s |
| 659_sleuth_ex1714 | SyRBo: 7231.48, SR: 7604.58, pval: 5.2E-01 | SyRBo: 660.4s, SR: 183.78s |
| 663_rabe_266     | SR: 19.49, SyRBo: 20.01, pval: 2.4E-01 = | SyRBo: 167.69s, SR: 72.88s |
| 665_sleuth_case2002 | SR: 5.3, SyRBo: 5.3, pval: 9.7E-01 = | SyRBo: 97.79s, SR: 38.17s |
| 666_rnftsa_ladata | SyRBo: 1.58, SR: 1.62, pval: 3.0E-01 | SyRBo: 83.81s, SR: 32.08s |
| Dataset                                                                 | Mean absolute error and pval                          | Run times                  |
|------------------------------------------------------------------------|-------------------------------------------------------|----------------------------|
| 678_visualizing_environmental                                          | SR: 2.44, SyRBo: 2.46, pval: 8.1E-01 =                | SyRBo: 72.36s, SR: 28.73s   |
| 687_sleuth_ex1605                                                      | SR: 13.04, SyRBo: 14.99, pval: 1.0E-04                  | SyRBo: 130.96s, SR: 46.48s  |
| 690_visualizing_galaxy                                                | SyRBo: 212.82, SR: 440.08, pval: 1.0E-04 !            | SyRBo: 408.21s, SR: 125.28s |
| 695_chatfield_4                                                        | SR: 16.79, SyRBo: 18.34, pval: 9.0E-03                 | SyRBo: 155.94s, SR: 57.64s  |
| 706_sleuth_case1202                                                    | SR: 49.57, SyRBo: 51.69, pval: 1.8E-01 =              | SyRBo: 172.97s, SR: 75.44s  |
| 712_chscase_geyser1                                                    | SyRBo: 8.2, SR: 8.88, pval: 0.0E+00 !                 | SyRBo: 100.15s, SR: 47.06s  |
Table 5 4-stage SyRBo: results of all datasets

| Dataset         | Mean absolute error and pval | Run times       |
|-----------------|------------------------------|-----------------|
| 1027_ESL        | SyRBo: 1.0, SR: 1.04, pval: 2.0E-04 ! | SyRBo: 93.37s, SR: 24.88s |
| 1028_SWD        | SyRBo: 0.61, SR: 0.62, pval: 1.3E-02 ! | SyRBo: 93.66s, SR: 24.91s |
| 1029_LEV        | SyRBo: 0.62, SR: 0.65, pval: 0.0E+00 ! | SyRBo: 92.83s, SR: 22.9s |
| 1030_Era        | SyRBo: 1.42, SR: 1.45, pval: 2.0E-04 ! | SyRBo: 96.63s, SR: 24.49s |
| 1089_USCrime    | SyRBo: 25.17, SR: 25.74, pval: 7.2E-01 | SyRBo: 168.69s, SR: 59.24s |
| 1096_FacultySalaries | SR: 3.51, SyRBo: 3.53, pval: 8.4E-01 = | SyRBo: 87.13s, SR: 27.14s |
| 192_vineyard    | SR: 2.34, SyRBo: 2.55, pval: 4.0E-02 ! | SyRBo: 97.59s, SR: 31.41s |
| 195_auto_price  | SyRBo: 1881.2, SR: 2047.73, pval: 8.4E-03 ! | SyRBo: 606.2s, SR: 156.64s |
| 207_autoPrice   | SyRBo: 1883.63, SR: 2046.39, pval: 6.0E-02 | SyRBo: 759.26s, SR: 155.12s |
| 210_cloud       | SyRBo: 0.49, SR: 0.5, pval: 9.4E-01 ! | SyRBo: 89.57s, SR: 23.4s |
| 228_elusage     | SyRBo: 12.45, SR: 14.36, pval: 6.0E-04 ! | SyRBo: 176.15s, SR: 77.55s |
| 229_pwLinear    | SyRBo: 1.49, SR: 1.59, pval: 1.4E-02 ! | SyRBo: 111.62s, SR: 33.75s |
| 230_machine_cpu | SyRBo: 43.28, SR: 47.09, pval: 7.9E-02 ! | SyRBo: 273.04s, SR: 91.55s |
| 4544_GeographicalO-riginalofMusic | SyRBo: 0.49, SR: 0.49, pval: 2.9E-01 ! | SyRBo: 117.26s, SR: 38.11s |
| 485_analcatdata_vehicle | SR: 151.23, SR: 186.52, pval: 8.0E-04 ! | SyRBo: 391.71s, SR: 131.13s |
| 505_tecator     | SR: 5.01, SyRBo: 5.05, pval: 9.0E-01 = | SyRBo: 161.8s, SR: 58.38s |
| 519_vinnie      | SR: 1.26, SyRBo: 1.3, pval: 9.1E-02 = | SyRBo: 118.67s, SR: 35.63s |
| 522_pm10        | SyRBo: 0.66, SR: 0.69, pval: 2.0E-04 ! | SyRBo: 111.11s, SR: 28.72s |
| 523_analcatdata_neavote | SyRBo: 0.5, SR: 0.5, pval: 9.4E-01 ! | SyRBo: 114.11s, SR: 37.29s |
| 527_analcatdata_election2000 | SR: 41409.25, SyRBo: 43867.25, pval: 4.9E-01 = | SyRBo: 865.92s, SR: 160.52s |
| 542_pollution   | SyRBo: 180.88, SR: 188.26, pval: 4.4E-01 ! | SyRBo: 367.71s, SR: 141.35s |
| 547_no2         | SyRBo: 0.56, SR: 0.59, pval: 4.0E-04 ! | SyRBo: 109.17s, SR: 28.85s |
| 556_analcatdata_apnea2 | SR: 869.07, SR: 881.56, pval: 8.6E-01 = | SyRBo: 238.22s, SR: 75.84s |
| 557_analcatdata_apnea1 | SyRBo: 861.47, SR: 869.01, pval: 9.1E-01 = | SyRBo: 215.12s, SR: 54.48s |
| 560_bodyfat     | SR: 4.24, SyRBo: 4.37, pval: 3.0E-01 = | SyRBo: 129.35s, SR: 40.74s |
| 561_cpu         | SyRBo: 29.33, SR: 35.67, pval: 3.6E-03 ! | SyRBo: 254.82s, SR: 95.29s |
| 579_fri_c0_250_5 | SyRBo: 0.38, SR: 0.45, pval: 0.0E+00 ! | SyRBo: 83.29s, SR: 20.24s |
| 581_fri_c3_500_25 | SyRBo: 0.68, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 107.74s, SR: 26.62s |
| 582_fri_c1_500_25 | SyRBo: 0.66, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 83.86s, SR: 20.72s |
| 583_fri_c1_1000_50 | SyRBo: 0.7, SR: 0.74, pval: 0.0E+00 ! | SyRBo: 107.04s, SR: 26.75s |
| 584_fri_c4_500_25 | SyRBo: 0.67, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 106.2s, SR: 26.11s |
| 586_fri_c3_1000_25 | SyRBo: 0.66, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 106.28s, SR: 26.3s |
| 588_fri_c4_1000_100 | SyRBo: 0.72, SR: 0.72, pval: 6.5E-01 ! | SyRBo: 105.29s, SR: 26.7s |
| 589_fri_c2_1000_25 | SyRBo: 0.67, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 89.31s, SR: 22.21s |
| 590_fri_c0_1000_50 | SyRBo: 0.36, SR: 0.4, pval: 0.0E+00 ! | SyRBo: 107.96s, SR: 28.67s |
| 591_fri_c1_100_10 | SyRBo: 0.68, SR: 0.74, pval: 6.6E-03 ! | SyRBo: 107.71s, SR: 26.22s |
| 592_fri_c4_1000_25 | SyRBo: 0.69, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 104.6s, SR: 25.92s |
| 593_fri_c1_1000_10 | SyRBo: 0.58, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 84.35s, SR: 20.31s |
| 594_fri_c2_100_5 | SyRBo: 0.62, SR: 0.7, pval: 0.0E+00 ! | SyRBo: 86.38s, SR: 21.38s |
| 595_fri_c0_1000_10 | SyRBo: 0.33, SR: 0.44, pval: 0.0E+00 ! | SyRBo: 110.51s, SR: 27.53s |
Table 5 (continued)

| Dataset      | Mean absolute error and pval | Run times          |
|--------------|------------------------------|--------------------|
| 596_fri_c2_250_5 | SyRBo: 0.59, SR: 0.69, pval: 0.0E+00 | SyRBo: 106.65s, SR: 26.23s |
| 597_fri_c2_500_5 | SyRBo: 0.57, SR: 0.67, pval: 0.0E+00 | SyRBo: 106.16s, SR: 26.2s |
| 598_fri_c0_1000_25 | SyRBo: 0.35, SR: 0.43, pval: 0.0E+00 | SyRBo: 106.81s, SR: 27.65s |
| 599_fri_c2_1000_5 | SyRBo: 0.54, SR: 0.67, pval: 0.0E+00 | SyRBo: 110.59s, SR: 26.69s |
| 601_fri_c1_250_5 | SyRBo: 0.53, SR: 0.66, pval: 0.0E+00 | SyRBo: 106.99s, SR: 26.05s |
| 602_fri_c3_250_10 | SyRBo: 0.66, SR: 0.73, pval: 0.0E+00 | SyRBo: 107.64s, SR: 26.21s |
| 603_fri_c0_250_50 | SyRBo: 0.39, SR: 0.4, pval: 2.9E-02 | SyRBo: 104.74s, SR: 27.57s |
| 604_fri_c4_500_10 | SyRBo: 0.63, SR: 0.71, pval: 0.0E+00 | SyRBo: 105.4s, SR: 25.79s |
| 605_fri_c2_250_25 | SyRBo: 0.68, SR: 0.7, pval: 6.7E-02 | SyRBo: 103.7s, SR: 25.76s |
| 607_fri_c4_1000_50 | SyRBo: 0.72, SR: 0.73, pval: 6.9E-02 | SyRBo: 111.03s, SR: 27.78s |
| 608_fri_c3_1000_10 | SyRBo: 0.59, SR: 0.7, pval: 0.0E+00 | SyRBo: 118.4s, SR: 28.63s |
| 609_fri_c0_1000_5 | SyRBo: 0.34, SR: 0.44, pval: 0.0E+00 | SyRBo: 116.25s, SR: 28.29s |
| 611_fri_c3_100_5 | SyRBo: 0.62, SR: 0.65, pval: 1.3E-02 | SyRBo: 106.5s, SR: 26.92s |
| 612_fri_c1_1000_5 | SyRBo: 0.54, SR: 0.69, pval: 0.0E+00 | SyRBo: 95.28s, SR: 22.96s |
| 613_fri_c3_250_5 | SyRBo: 0.56, SR: 0.64, pval: 0.0E+00 | SyRBo: 85.81s, SR: 20.96s |
| 615_fri_c4_250_10 | SyRBo: 0.64, SR: 0.7, pval: 0.0E+00 | SyRBo: 103.59s, SR: 25.5s |
| 616_fri_c4_500_50 | SyRBo: 0.73, SR: 0.74, pval: 7.3E-01 | SyRBo: 101.64s, SR: 25.4s |
| 617_fri_c3_500_5 | SyRBo: 0.55, SR: 0.66, pval: 0.0E+00 | SyRBo: 106.97s, SR: 26.56s |
| 618_fri_c3_1000_50 | SyRBo: 0.7, SR: 0.72, pval: 0.0E+00 | SyRBo: 106.08s, SR: 26.47s |
| 620_fri_c1_1000_25 | SyRBo: 0.68, SR: 0.74, pval: 0.0E+00 | SyRBo: 109.58s, SR: 27.21s |
| 621_fri_c0_100_10 | SyRBo: 0.4, SR: 0.45, pval: 0.0E+00 | SyRBo: 103.48s, SR: 25.55s |
| 622_fri_c2_1000_50 | SyRBo: 0.71, SR: 0.73, pval: 6.0E-04 | SyRBo: 106.03s, SR: 26.9s |
| 623_fri_c4_1000_10 | SyRBo: 0.6, SR: 0.69, pval: 0.0E+00 | SyRBo: 109.83s, SR: 26.66s |
| 624_fri_c0_100_5 | SyRBo: 0.41, SR: 0.47, pval: 0.0E+00 | SyRBo: 110.93s, SR: 26.78s |
| 626_fri_c2_500_50 | SyRBo: 0.72, SR: 0.73, pval: 8.7E-02 | SyRBo: 104.49s, SR: 26.41s |
| 627_fri_c2_500_10 | SyRBo: 0.58, SR: 0.69, pval: 0.0E+00 | SyRBo: 106.73s, SR: 25.88s |
| 628_fri_c3_1000_5 | SyRBo: 0.58, SR: 0.66, pval: 0.0E+00 | SyRBo: 108.33s, SR: 26.76s |
| 631_fri_c1_500_5 | SyRBo: 0.54, SR: 0.67, pval: 0.0E+00 | SyRBo: 107.88s, SR: 26.04s |
| 633_fri_c0_500_25 | SyRBo: 0.34, SR: 0.42, pval: 0.0E+00 | SyRBo: 104.15s, SR: 26.69s |
| 634_fri_c2_100_10 | SyRBo: 0.67, SR: 0.71, pval: 7.3E-03 | SyRBo: 106.29s, SR: 26.07s |
| 635_fri_c0_250_10 | SyRBo: 0.37, SR: 0.52, pval: 0.0E+00 | SyRBo: 102.86s, SR: 25.5s |
| 637_fri_c1_500_50 | SyRBo: 0.74, SR: 0.76, pval: 5.0E-04 | SyRBo: 104.06s, SR: 26.12s |
| 641_fri_c1_500_10 | SyRBo: 0.59, SR: 0.74, pval: 0.0E+00 | SyRBo: 108.72s, SR: 26.43s |
| 643_fri_c2_500_25 | SyRBo: 0.71, SR: 0.76, pval: 0.0E+00 | SyRBo: 105.35s, SR: 26.16s |
| 644_fri_c4_250_25 | SyRBo: 0.72, SR: 0.75, pval: 4.9E-03 | SyRBo: 104.33s, SR: 25.79s |
| 645_fri_c3_500_50 | SR: 0.7, SyRBo: 0.7, pval: 7.6E-01 | SyRBo: 103.7s, SR: 25.58s |
| 646_fri_c3_500_10 | SyRBo: 0.61, SR: 0.69, pval: 0.0E+00 | SyRBo: 91.22s, SR: 22.33s |
| 647_fri_c1_250_10 | SyRBo: 0.61, SR: 0.74, pval: 0.0E+00 | SyRBo: 103.42s, SR: 25.12s |
| 648_fri_c1_250_50 | SyRBo: 0.72, SR: 0.75, pval: 2.1E-03 | SyRBo: 108.59s, SR: 27.71s |
| 649_fri_c0_500_5 | SyRBo: 0.34, SR: 0.46, pval: 0.0E+00 | SyRBo: 112.89s, SR: 27.46s |
| 650_fri_c0_500_50 | SyRBo: 0.36, SR: 0.39, pval: 0.0E+00 | SyRBo: 85.82s, SR: 22.59s |
| Dataset                  | Mean absolute error and pval | Run times           |
|--------------------------|-----------------------------|---------------------|
| 651_fri_c0_100_25        | SyRBo: 0.51, SR: 0.53, pval: 4.4E-02   | SyRBo: 111.2s, SR: 28.12s |
| 653_fri_c0_250_25        | SyRBo: 0.36, SR: 0.41, pval: 0.0E+00   | SyRBo: 83.61s, SR: 21.3s    |
| 654_fri_c0_500_10        | SyRBo: 0.35, SR: 0.46, pval: 0.0E+00   | SyRBo: 84.78s, SR: 21.04s   |
| 656_fri_c1_100_5         | SyRBo: 0.54, SR: 0.64, pval: 0.0E+00   | SyRBo: 86.37s, SR: 21.66s    |
| 657_fri_c2_250_10        | SyRBo: 0.63, SR: 0.69, pval: 0.0E+00   | SyRBo: 105.03s, SR: 25.64s  |
| 658_fri_c3_250_25        | SyRBo: 0.72, SR: 0.75, pval: 8.2E-03   | SyRBo: 103.48s, SR: 25.53s   |
| 659_sleuth_ex1714        | SyRBo: 6437.7, SR: 7641.02, pval: 7.0E-03 | SyRBo: 723.0s, SR: 171.93s   |
| 663_rabe_266             | SR: 19.85, SyRBo: 20.79, pval: 6.7E-02 = | SyRBo: 188.09s, SR: 66.1s    |
| 665_sleuth_case2002      | SR: 5.06, SyRBo: 5.15, pval: 4.3E-01 = | SyRBo: 125.81s, SR: 37.11s   |
| 666_rmftsa_ladata        | SyRBo: 1.58, SR: 1.65, pval: 3.1E-02   | SyRBo: 110.41s, SR: 31.81s   |
| 678_visualizing_environment | SR: 2.44, SyRBo: 2.46, pval: 8.1E-01 = | SyRBo: 116.57s, SR: 35.84s   |
| 687_sleuth_ex1605        | SR: 13.27, SyRBo: 15.03, pval: 1.5E-03 | SyRBo: 187.9s, SR: 58.29s    |
| 690_visualizing_galaxy   | SyRBo: 219.9, SR: 470.38, pval: 0.0E+00! | SyRBo: 599.04s, SR: 171.58s  |
| 695_chatfield_4          | SR: 16.84, SyRBo: 17.27, pval: 4.3E-01 = | SyRBo: 245.5s, SR: 82.54s    |
| 706_sleuth_case1202      | SR: 46.85, SyRBo: 48.25, pval: 2.8E-01 = | SyRBo: 286.46s, SR: 95.73s    |
| 712_chscase_geyser1      | SyRBo: 8.27, SR: 9.11, pval: 0.0E+00! | SyRBo: 114.63s, SR: 43.05s    |
| Dataset               | Mean absolute error and pval | Run times          |
|----------------------|------------------------------|--------------------|
| 1027_ESL             | SyRBo: 0.99, SR: 1.04, pval: 0.0E+00 ! | SyRBo: 138.45s, SR: 30.02s |
| 1028_SWD             | SyRBo: 0.6, SR: 0.62, pval: 3.2E-02 ! | SyRBo: 112.18s, SR: 24.21s |
| 1029_LEV             | SyRBo: 0.61, SR: 0.65, pval: 0.0E+00 ! | SyRBo: 108.52s, SR: 21.54s |
| 1030 Era             | SyRBo: 1.41, SR: 1.46, pval: 0.0E+00 ! | SyRBo: 114.87s, SR: 23.88s |
| 1089_USCrime         | SR: 25.79, SyRBo: 26.52, pval: 6.6E-01 = | SyRBo: 244.05s, SR: 70.49s |
| 1096_FacultySalaries | SR: 3.44, SyRBo: 3.66, pval: 2.0E-01 = | SyRBo: 137.41s, SR: 36.12s |
| 192_vineyard         | SR: 2.45, SyRBo: 2.53, pval: 2.7E-01 = | SyRBo: 145.21s, SR: 37.57s |
| 195_auto_price       | SyRBo: 1846.49, SR: 1984.64, pval: 2.5E-02 ! | SyRBo: 1136.73s, SR: 181.96s |
| 207_autoPrice        | SyRBo: 1917.2, SR: 2004.04, pval: 3.0E-01 | SyRBo: 840.85s, SR: 146.49s |
| 210_cloud            | SyRBo: 0.5, SR: 0.51, pval: 5.9E-01 | SyRBo: 136.83s, SR: 28.6s |
| 228_elusage          | SyRBo: 11.83, SR: 13.25, pval: 8.6E-03 ! | SyRBo: 168.85s, SR: 59.4s |
| 229_pwLinear         | SyRBo: 1.49, SR: 1.58, pval: 4.6E-02 ! | SyRBo: 147.58s, SR: 35.66s |
| 230_machine_cpu      | SyRBo: 41.55, SR: 47.64, pval: 6.6E-03 ! | SyRBo: 302.39s, SR: 89.29s |
| 4544_GeographicalOriginalof-Music | SyRBo: 0.49, SR: 0.5, pval: 5.6E-01 | SyRBo: 139.81s, SR: 37.08s |
| 485_analcatdata_vehicle | SyRBo: 152.36, SR: 185.49, pval: 6.0E-04 ! | SyRBo: 402.2s, SR: 123.11s |
| 505_tecator          | SyRBo: 5.03, SR: 5.42, pval: 9.3E-02 | SyRBo: 192.01s, SR: 56.36s |
| 519_vinnie           | SR: 1.26, SyRBo: 1.28, pval: 2.0E-01 = | SyRBo: 144.14s, SR: 34.81s |
| 522_pm10             | SyRBo: 0.66, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 109.71s, SR: 22.76s |
| 523_analcatdata_neavote | SyRBo: 0.5, SR: 0.51, pval: 2.4E-01 | SyRBo: 145.96s, SR: 40.78s |
| 527_analcatdata_election2000 | SyRBo: 41978.87, SR: 42335.38, pval: 9.2E-01 | SyRBo: 1276.47s, SR: 189.6s |
| 542_pollution        | SyRBo: 184.53, SR: 186.47, pval: 8.6E-01 | SyRBo: 482.44s, SR: 152.25s |
| 547_no2              | SyRBo: 0.56, SR: 0.59, pval: 1.2E-03 ! | SyRBo: 149.77s, SR: 32.23s |
| 556_analcatdata_apnea2 | SyRBo: 831.42, SR: 837.88, pval: 9.0E-01 | SyRBo: 374.25s, SR: 108.94s |
Table 6 (continued)

| Dataset                  | Mean absolute error and pval | Run times |
|--------------------------|-------------------------------|-----------|
| 557_analcatdata_apnea1   | SR: 833.02, SyRBo: 845.73, 7.5E-01 = | SyRBo: 333.67s, SR: 66.34s |
| 560_bodyfat              | SyRBo: 4.26, SR: 4.28, pval: 9.1E-01 | SyRBo: 127.64s, SR: 31.6s |
| 561_cpu                  | SyRBo: 28.02, SR: 32.76, pval: 2.5E-02 ! | SyRBo: 268.56s, SR: 94.84s |
| 579_fri_c0_250_5         | SyRBo: 0.38, SR: 0.45, pval: 0.0E+00 ! | SyRBo: 127.54s, SR: 24.79s |
| 581_fri_c3_500_25        | SyRBo: 0.68, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 105.16s, SR: 20.78s |
| 582_fri_c1_500_25        | SyRBo: 0.66, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 104.49s, SR: 20.64s |
| 583_fri_c1_1000_50       | SyRBo: 0.7, SR: 0.74, pval: 0.0E+00 ! | SyRBo: 109.61s, SR: 21.76s |
| 584_fri_c4_500_25        | SyRBo: 0.66, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 131.15s, SR: 26.05s |
| 586_fri_c3_1000_25       | SyRBo: 0.67, SR: 0.7, pval: 0.0E+00 ! | SyRBo: 111.2s, SR: 21.99s |
| 588_fri_c4_1000_100      | SyRBo: 0.72, SR: 0.72, pval: 5.1E-01 | SyRBo: 109.24s, SR: 21.66s |
| 589_fri_c2_1000_25       | SyRBo: 0.67, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 131.08s, SR: 26.12s |
| 590_fri_c0_1000_50       | SyRBo: 0.35, SR: 0.4, pval: 0.0E+00 ! | SyRBo: 130.53s, SR: 27.88s |
| 591_fri_c1_100_10        | SyRBo: 0.68, SR: 0.74, pval: 1.1E-02 ! | SyRBo: 108.96s, SR: 21.35s |
| 592_fri_c4_1000_25       | SyRBo: 0.68, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 106.21s, SR: 20.99s |
| 593_fri_c1_1000_10       | SyRBo: 0.55, SR: 0.71, pval: 0.0E+00 ! | SyRBo: 108.31s, SR: 20.94s |
| 594_fri_c2_100_5         | SyRBo: 0.61, SR: 0.68, pval: 0.0E+00 ! | SyRBo: 111.13s, SR: 22.03s |
| 595_fri_c0_1000_10       | SyRBo: 0.31, SR: 0.44, pval: 0.0E+00 ! | SyRBo: 106.28s, SR: 21.17s |
| 596_fri_c2_250_5         | SyRBo: 0.58, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 104.44s, SR: 20.42s |
| 597_fri_c2_500_5         | SyRBo: 0.55, SR: 0.67, pval: 0.0E+00 ! | SyRBo: 106.51s, SR: 20.93s |
| 598_fri_c0_1000_25       | SyRBo: 0.33, SR: 0.43, pval: 0.0E+00 ! | SyRBo: 107.87s, SR: 22.01s |
| 599_fri_c2_1000_5        | SyRBo: 0.53, SR: 0.67, pval: 0.0E+00 ! | SyRBo: 107.85s, SR: 20.91s |
| 601_fri_c1_250_5         | SyRBo: 0.52, SR: 0.66, pval: 0.0E+00 ! | SyRBo: 132.58s, SR: 25.88s |
| 602_fri_c3_250_10        | SyRBo: 0.63, SR: 0.72, pval: 0.0E+00 ! | SyRBo: 103.12s, SR: 19.87s |
Table 6 (continued)

| Dataset          | Mean absolute error and pval | Run times       |
|------------------|------------------------------|-----------------|
| 603_fri_c0_250_50| SyRBo: 0.38, SR: 0.41, pval: 0.0E+00 | SyRBo: 104.63s, SR: 21.89s |
| 604_fri_c4_500_10| SyRBo: 0.62, SR: 0.72, pval: 0.0E+00 | SyRBo: 107.77s, SR: 21.26s |
| 605_fri_c2_250_25| SyRBo: 0.67, SR: 0.7, pval: 5.1E-03 | SyRBo: 100.7s, SR: 20.1s |
| 606_fri_c2_1000_10| SyRBo: 0.56, SR: 0.68, pval: 0.0E+00 | SyRBo: 132.9s, SR: 25.78s |
| 607_fri_c4_1000_50| SyRBo: 0.71, SR: 0.73, pval: 3.0E-02 | SyRBo: 128.27s, SR: 25.68s |
| 608_fri_c3_1000_10| SyRBo: 0.57, SR: 0.7, pval: 0.0E+00 | SyRBo: 137.65s, SR: 26.7s |
| 609_fri_c0_1000_5| SyRBo: 0.33, SR: 0.44, pval: 0.0E+00 | SyRBo: 137.83s, SR: 26.65s |
| 611_fri_c3_100_5  | SyRBo: 0.61, SR: 0.66, pval: 3.3E-03 | SyRBo: 133.97s, SR: 27.56s |
| 612_fri_c1_1000_5  | SyRBo: 0.53, SR: 0.68, pval: 0.0E+00 | SyRBo: 142.24s, SR: 27.26s |
| 613_fri_c3_250_5  | SyRBo: 0.56, SR: 0.65, pval: 0.0E+00 | SyRBo: 138.42s, SR: 27.06s |
| 615_fri_c4_250_10  | SyRBo: 0.64, SR: 0.7, pval: 0.0E+00 | SyRBo: 105.66s, SR: 20.78s |
| 616_fri_c4_500_50  | SyRBo: 0.73, SR: 0.74, pval: 3.6E-01 | SyRBo: 129.48s, SR: 25.82s |
| 617_fri_c3_500_5  | SyRBo: 0.55, SR: 0.64, pval: 0.0E+00 | SyRBo: 134.39s, SR: 26.93s |
| 618_fri_c3_1000_50  | SyRBo: 0.71, SR: 0.73, pval: 1.8E-03 | SyRBo: 128.38s, SR: 25.71s |
| 620_fri_c1_1000_25  | SyRBo: 0.66, SR: 0.74, pval: 0.0E+00 | SyRBo: 128.35s, SR: 25.35s |
| 621_fri_c0_100_10  | SyRBo: 0.39, SR: 0.47, pval: 0.0E+00 | SyRBo: 125.76s, SR: 24.98s |
| 622_fri_c2_1000_50  | SyRBo: 0.7, SR: 0.73, pval: 0.0E+00 | SyRBo: 130.39s, SR: 26.39s |
| 623_fri_c4_1000_10  | SyRBo: 0.58, SR: 0.69, pval: 0.0E+00 | SyRBo: 137.99s, SR: 26.69s |
| 624_fri_c0_100_5  | SyRBo: 0.41, SR: 0.47, pval: 0.0E+00 | SyRBo: 132.62s, SR: 25.69s |
| 626_fri_c2_500_50  | SyRBo: 0.71, SR: 0.73, pval: 8.3E-02 | SyRBo: 127.3s, SR: 25.7s |
| 627_fri_c2_500_10  | SyRBo: 0.58, SR: 0.69, pval: 0.0E+00 | SyRBo: 129.13s, SR: 25.24s |
| 628_fri_c3_1000_5  | SyRBo: 0.57, SR: 0.66, pval: 0.0E+00 | SyRBo: 108.94s, SR: 21.41s |
| 631_fri_c1_500_5  | SyRBo: 0.54, SR: 0.68, pval: 0.0E+00 | SyRBo: 136.54s, SR: 26.34s |
| Dataset         | Mean absolute error and pval | Run times           |
|-----------------|------------------------------|---------------------|
| 633_fri_c0_500_25 | SyRBo: 0.33, SR: 0.43, pval: 0.0E+00 ! | SyRBo: 101.34s, SR: 20.79s |
| 634_fri_c2_100_10 | SyRBo: 0.64, SR: 0.7, pval: 8.9E-03 ! | SyRBo: 135.04s, SR: 26.73s |
| 635_fri_c0_250_10 | SyRBo: 0.36, SR: 0.51, pval: 0.0E+00 ! | SyRBo: 132.75s, SR: 26.19s |
| 637_fri_c1_500_50 | SyRBo: 0.75, SR: 0.76, pval: 1.6E-01 | SyRBo: 132.3s, SR: 26.57s |
| 641_fri_c1_500_10 | SyRBo: 0.57, SR: 0.73, pval: 0.0E+00 ! | SyRBo: 128.61s, SR: 25.01s |
| 643_fri_c2_500_25 | SyRBo: 0.72, SR: 0.75, pval: 8.3E-03 ! | SyRBo: 105.24s, SR: 20.85s |
| 644_fri_c4_250_25 | SyRBo: 0.7, SR: 0.74, pval: 2.5E-03 ! | SyRBo: 129.54s, SR: 25.53s |
| 645_fri_c3_500_50 | SyRBo: 0.69, SR: 0.71, pval: 2.4E-02 ! | SyRBo: 125.64s, SR: 24.97s |
| 646_fri_c3_500_10 | SyRBo: 0.58, SR: 0.69, pval: 0.0E+00 ! | SyRBo: 134.8s, SR: 26.19s |
| 647_fri_c1_250_10 | SyRBo: 0.6, SR: 0.74, pval: 0.0E+00 ! | SyRBo: 105.11s, SR: 20.46s |
| 648_fri_c1_250_50 | SyRBo: 0.73, SR: 0.74, pval: 5.4E-01 | SyRBo: 128.78s, SR: 26.5s |
| 649_fri_c0_500_5 | SyRBo: 0.34, SR: 0.46, pval: 0.0E+00 ! | SyRBo: 129.74s, SR: 25.21s |
| 650_fri_c0_500_50 | SyRBo: 0.34, SR: 0.39, pval: 0.0E+00 ! | SyRBo: 128.54s, SR: 27.19s |
| 651_fri_c0_100_25 | SyRBo: 0.51, SR: 0.53, pval: 4.6E-02 ! | SyRBo: 131.91s, SR: 26.73s |
| 653_fri_c0_250_25 | SyRBo: 0.35, SR: 0.41, pval: 0.0E+00 ! | SyRBo: 127.19s, SR: 26.07s |
| 654_fri_c0_500_10 | SyRBo: 0.34, SR: 0.46, pval: 0.0E+00 ! | SyRBo: 108.92s, SR: 21.67s |
| 656_fri_c1_100_5 | SyRBo: 0.55, SR: 0.64, pval: 0.0E+00 ! | SyRBo: 141.71s, SR: 28.99s |
| 657_fri_c2_250_10 | SyRBo: 0.62, SR: 0.68, pval: 0.0E+00 ! | SyRBo: 129.88s, SR: 25.33s |
| 658_fri_c3_250_25 | SyRBo: 0.73, SR: 0.74, pval: 1.2E-01 | SyRBo: 125.15s, SR: 24.64s |
| 659_sleuth_ex1714 | SyRBo: 6464.12, SR: 7876.73, pval: 5.2E-02 | SyRBo: 976.99s, SR: 183.76s |
| 663_rabe_266 | SR: 19.33, SyRBo: 20.57, pval: 1.4E-02 | SyRBo: 212.59s, SR: 60.57s |
| 665_sleuth_case2002 | SR: 5.18, SyRBo: 5.37, pval: 4.0E-01 = | SyRBo: 148.57s, SR: 37.37s |
| 666_rmftsa_ladata | SyRBo: 1.59, SR: 1.63, pval: 1.6E-01 | SyRBo: 134.26s, SR: 32.33s |
Acknowledgements This work was supported by National Institutes of Health (USA) Grants LM010098, LM012601, AI116794. We thank Hagai Ravid for spotting an error in an earlier version of the code.

References

1. T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16. ACM, New York, NY, USA, pp. 785–794 (2016). https://doi.org/10.1145/2939672.2939785
2. M. Fink, P. Perona, Mutual boosting for contextual inference, in Advances in Neural Information Processing Systems. ed. by S. Thrun, L.K. Saul, B. Schölkopf, vol. 16, pp. 1515–1522 (2004). https://proceedings.neurips.cc/paper/2003/file/070dbb6024b5ef93784428af71f2146-Paper.pdf
3. Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55(1), 119–139 (1997)
4. J.H. Friedman, Greedy function approximation: a gradient boosting machine. Ann. Stat. 29, 1189–1232 (2001)
5. GPLearn. https://gplearn.readthedocs.io/ (2020). Accessed 20 Nov 2020
6. M.B. Harries, Boosting a strong learner: evidence against the minimum margin, in Proceedings of the 16th International Conference on Machine Learning, ICML ’99. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 171–180 (1999)
7. H. Iba, Bagging, boosting, and bloating in genetic programming, in Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation, vol. 2, pp. 1053–1060 (1999)
8. S. Karakatič, V. Podgorelec, Building boosted classification tree ensemble with genetic programming, in Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 165–166 (2018)
9. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, in Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS ’17. Curran Associates Inc., Red Hook, NY, USA, pp. 3149–3157 (2017)
10. E. Oliveira, A. Pozo, S.R. Vergilio, Using boosting techniques to improve software reliability models based on genetic programming, in 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’06). IEEE, pp. 643–650 (2006)
11. L.O.V. Oliveira, F.E. Otero, G.L. Pappa, J. Albinati, Sequential symbolic regression with genetic programming, in Genetic Programming Theory and Practice XII. ed. by R. Riolo, W.P. Worzel, M. Kotanchek (Springer International Publishing, Cham, 2015), pp. 73–90
12. P. Orzechowski, W. La Cava, J.H. Moore, Where are we now? a large benchmark study of recent symbolic regression methods, in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1183–1190 (2018)

13. G. Paris, D. Robilliard, C. Fonlupt, Applying boosting techniques to genetic programming, in *International Conference on Artificial Evolution (Evolution Artificielle)*. Springer, pp. 267–278 (2001)

14. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830 (2011)

15. R.E. Schapire, The strength of weak learnability. *Mach. Learn.* 5(2), 197–227 (1990)

16. Scikit-learn: machine learning in python. https://scikit-learn.org/ (2020). Accessed 20 Nov 2020

17. J. Wickramaratna, S. Holden, B. Buxton, Performance degradation in boosting, in *International Workshop on Multiple Classifier Systems*. Springer, pp. 11–21 (2001)