Symbolic Regression of Confidence Intervals for Conformal Prediction - Additional Materials

Alberto Tonda[1,2] [0000−0001−5895−4809],
Alejandro Lopez-Rincon[3] [0000−0003−4491−5889],
David Rojas-Velazquez[3,4] [0000−0003−4402−4736], and
Evelyne Lutton[1,2] [0000−0003−0889−4427]

[1] UMR 518 MIA-PS, INRAE, Université Paris-Saclay, Palaiseau France
[2] UAR 3611 ISC-PIF, CNRS, Paris, France
{alberto.tonda,evelyne.lutton}@inrae.fr
[3] Division of Pharmacology, University of Utrecht, The Netherlands
[4] Julius Center for Health Sciences and Primary Care, University Medical Center Utrecht, The Netherlands
{a.lopezrincon,e.d.rojasvelazquez}@uu.nl

Abstract. This document contains the additional materials for the paper *Symbolic Regression of Confidence Intervals for Conformal Prediction*. The main document can be found on the website of the EA 2024 conference, https://ea2024.inria.fr/.

Keywords: Confidence intervals · Conformal prediction · Conformal regression · Machine learning · Regression · Symbolic regression.

References

1. Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)
2. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794. ACM (8 2016)
3. Fischer, S.F., Feurer, M., Bischl, B.: OpenML-CTR23–a curated tabular regression benchmarking suite. In: AutoML Conference 2023 (Workshop) (2023)

A Additional materials

A.1 Results of the experimental run on OpenML-CTR23

OpenML-CTR23 [3] is a benchmark suite of hand-picked data sets from OpenML, chosen for several desirable characteristics. All data sets have been selected according to desirable properties, such as: containing between 500 and 100,000 samples and less than 5,000 features; having a referenced source; containing no artificial data; presenting a regression task is not trivially solvable by a linear model, or in other words, a linear regression algorithm cannot attain a $R2 = 1.00$ in a cross-validation; and so on.

We performed a trial run on the data sets included in OpenML-CTR23, to (i) compare the performance of RF [1] against XGBoost [2], which is commonly considered the state of the art for regression on tabular data; (ii) assess whether the proposed split (50% training, 25% calibration, 25% test) can still deliver good predictions; (iii) identify data sets for which the computation of confidence intervals does not make sense, as the performance of RF is nearly perfect ($R2 > 0.95$) or extremely poor ($R2 < 0.40$), using two arbitrary thresholds. The complete results are reported in Table 1.

A.2 Full experimental results

Table 2 shows the detailed performance of each method on each data set, for coverage and median width of the confidence intervals. Interestingly, $SRCP$ often delivers the tightest confidence intervals, with the lowest empirical coverage on the test set.

A.3 Pareto fronts

As we are employing a multi-objective approach (evaluating coverage and median width of the confidence intervals) to compare the different CP methods, each run on a different data set produces a Pareto front. The Pareto fronts are reported in Figures 1-3.

Table 1. Data sets in OpenML-CTR23, with mean performance of Random Forest and XGBoost regressors in a 10-fold cross-validation. Rows highlighted in grey represent data sets that have been discarded from the experimental evaluation, due to poor ($R2 < 0.40$) or exceptional ($R2 > 0.95$) performance.

Task ID	Data set name	Samples	Features	Missing values	Categorical features	RF R2 (test) mean +/- std	XGB R2 (test) mean +/- std
361234	abalone	4,177	8	0	1	0.53 +/- 0.02	0.47 +/- 0.02
361235	airfoil_self_noise	1,503	5	0	0	0.90 +/- 0.01	0.96 +/- 0.01
361236	auction_verification	2,043	7	0	2	0.99 +/- 0.00	1.00 +/- 0.00
361267	brazilian_houses	10,692	9	0	4	0.52 +/- 0.24	0.59 +/- 0.36
361255	california_housing	20,640	8	0	0	0.81 +/- 0.00	0.83 +/- 0.01
361622	cars	804	17	0	0	0.94 +/- 0.01	0.94 +/- 0.02
361237	concrete_compressive_strength	1,030	8	0	0	0.88 +/- 0.01	0.94 +/- 0.02
361261	cps88wages	28,155	6	0	4	0.15 +/- 0.02	0.21 +/- 0.04
361256	cpu_activity	8,192	21	0	0	0.98 +/- 0.00	0.98 +/- 0.01
361257	diamonds	53,940	9	0	3	0.98 +/- 0.00	0.98 +/- 0.00
361617	energy_efficiency	768	8	0	0	1.00 +/- 0.00	1.00 +/- 0.00
361272	fifa	19,178	28	0	1	0.77 +/- 0.01	0.75 +/- 0.04
361618	forest_fires	517	12	0	2	-0.95 +/- 1.26	-10.35 +/- 17.06
361268	fps_benchmark	24,624	39	69,696	13	0.99 +/- 0.00	1.00 +/- 0.00
361243	geographical_origin_of_music	1,059	116	0	0	0.23 +/- 0.02	0.22 +/- 0.09
361251	grid_stability	10,000	12	0	0	0.89 +/- 0.00	0.93 +/- 0.01
361269	health_insurance	22,272	11	0	7	0.32 +/- 0.01	0.31 +/- 0.02
361258	kin8nm	8,192	8	0	0	0.68 +/- 0.01	0.78 +/- 0.01
361266	kings_county	21,613	21	0	4	0.87 +/- 0.01	0.88 +/- 0.04
361260	miami_housing	13,932	15	0	0	0.90 +/- 0.00	0.91 +/- 0.01
361616	Moneyball	1,232	10	3,600	4	0.92 +/- 0.00	0.93 +/- 0.00
361247	naval_propulsion_plant	11,934	14	0	0	0.99 +/- 0.00	1.00 +/- 0.00
361241	physiochemical_protein	45,730	9	0	0	0.64 +/- 0.00	0.63 +/- 0.01
361259	pumadyn32nh	8,192	32	0	0	0.64 +/- 0.00	0.58 +/- 0.03
361621	QSAR_fish_toxicity	908	6	0	0	0.61 +/- 0.03	0.58 +/- 0.08
361250	red_wine	1,599	11	0	0	0.44 +/- 0.02	0.44 +/- 0.09
361254	sarcos	48,933	21	0	0	0.97 +/- 0.00	0.98 +/- 0.00
361264	socmob	1,156	5	0	4	0.75 +/- 0.03	0.78 +/- 0.21
361244	solar_flare	1,066	10	0	8	-0.09 +/- 0.12	-0.42 +/- 0.38
361623	space_ga	3,107	6	0	0	0.63 +/- 0.01	0.69 +/- 0.04
361619	student_performance_por	649	30	0	17	0.23 +/- 0.08	0.17 +/- 0.19
361242	superconductivity	21,263	81	0	0	0.91 +/- 0.00	0.92 +/- 0.01
361252	video_transcoding	68,784	18	0	2	0.98 +/- 0.00	0.99 +/- 0.00
361253	wave_energy	72,000	48	0	0	0.83 +/- 0.00	0.97 +/- 0.00
361249	white_wine	4,898	11	0	0	0.46 +/- 0.01	0.50 +/- 0.05

Table 2. Detailed results of the experimental run for each data set, reporting values of coverage and median width of the confidence intervals for each method. Cells in green show the best value obtained in the row, cells in magenta show the worst value.

Data set name	Metric	SCP	NCP_d	NCP_{std}	NCP_{oob}	NCP_{var}	MCP	$SRCP$
abalone	Coverage	0.9474	0.9502	0.9512	0.9455	0.9311	0.9483	0.9522
	Median	2.9545	3.2021	3.6647	3.3293	2.7972	2.6755	2.6284
airfoil_self_noise	Coverage	0.9362	0.9388	0.9468	0.9441	0.8989	0.9282	0.9441
	Median	1.1607	1.4426	1.6024	1.5573	1.0324	1.1157	0.9472
brazilian_houses	Coverage	0.9577	0.9562	0.9581	0.9540	0.9600	0.9727	0.9633
	Median	0.4947	0.4813	0.3420	0.2937	0.4858	0.3453	0.2594
california_housing	Coverage	0.9531	0.9547	0.9521	0.9514	0.9554	0.9663	0.9525
	Median	1.9542	2.3276	1.8447	1.6705	1.4844	1.7037	1.3536
cars	Coverage	0.9303	0.9751	0.9502	0.9453	0.9701	0.9602	0.9254
	Median	1.0866	5.2368	1.2290	0.7567	0.9936	1.0866	0.6614
concrete_compressive_strength	Coverage	0.9264	0.9574	0.9535	0.9651	0.9806	0.9302	0.9147
	Median	1.2917	2.0088	1.7034	1.6180	1.6667	1.2881	0.9751
fifa	Coverage	0.9545	0.9495	0.9395	0.9395	0.9452	0.9568	0.9395
	Median	1.8810	1.6280	0.5688	0.5400	0.5754	0.5692	0.5399
grid_stability	Coverage	0.9552	0.9424	0.9432	0.9560	0.9360	0.9520	0.9540
	Median	1.3653	1.5484	1.4467	1.3390	1.4780	1.3759	1.2266
health_insurance	Coverage	0.9481	0.9582	0.9510	0.9515	0.9542	0.9623	0.9481
	Median	3.3151	5.7595	3.6869	4.2658	5.0377	3.6209	3.2132
kin8nm	Coverage	0.9482	0.9526	0.9424	0.9507	0.9458	0.9463	0.9551
	Median	2.1733	2.7016	2.3705	2.0131	2.3174	2.1190	2.0097
kings_county	Coverage	0.9524	0.9556	0.9617	0.9563	0.9574	0.9684	0.9552
	Median	1.4712	1.1274	1.0826	0.9722	0.7327	0.9059	0.9615
miami_housing	Coverage	0.9526	0.9518	0.9475	0.9500	0.9388	0.9701	0.9440
	Median	1.1928	0.8502	0.5071	0.5355	0.3968	0.5779	0.5633
Moneyball	Coverage	0.9253	0.9805	0.9351	0.9448	0.9448	0.9448	0.9156
	Median	1.0988	3.3098	1.4842	1.4414	1.7583	1.1847	1.0084
physiochemical_protein	Coverage	0.9484	0.9480	0.9508	0.9496	0.9493	0.9580	0.9514
	Median	2.6304	2.4417	2.9550	2.3881	1.9352	2.3144	2.2510
pumadyn32nh	Coverage	0.9468	0.9458	0.9580	0.9639	0.9526	0.9526	0.9448
	Median	2.3860	2.7065	2.8763	3.0634	2.9467	2.5839	2.3213
QSAR_fish_toxicity	Coverage	0.9868	0.9604	0.9824	0.9648	0.9559	0.9780	0.9207
	Median	3.4230	3.2027	3.4439	4.5222	3.5589	3.7275	1.9548
red_wine	Coverage	0.9550	0.9625	0.9550	0.9525	0.9575	0.9600	0.9500
	Median	3.4234	4.5759	3.2935	3.4862	3.5041	3.7128	3.0421
socmob	Coverage	0.9412	0.9585	0.9550	0.9377	0.9550	0.9550	0.9550
	Median	1.5951	1.5285	0.5835	0.5806	0.3293	0.7478	0.3265
space_ga	Coverage	0.9614	0.9575	0.9601	0.9678	0.9511	0.9665	0.9562
	Median	2.4853	3.5980	2.9231	3.0588	2.8941	2.4853	2.0549
superconductivity	Coverage	0.9528	0.9492	0.9464	0.9481	0.9466	0.9626	0.9518
	Median	1.3232	4.9142	0.7834	0.7384	0.5328	0.9699	0.9111
wave_energy	Coverage	0.9529	0.9479	0.9489	0.9519	0.9516	0.9531	0.9535
	Median	1.7184	2.9430	1.9980	1.6872	1.8567	1.6691	1.5973
white_wine	Coverage	0.9592	0.9584	0.9461	0.9502	0.9665	0.9608	0.9600
	Median	2.9805	3.2859	3.6857	3.2506	2.9687	3.1553	2.7576

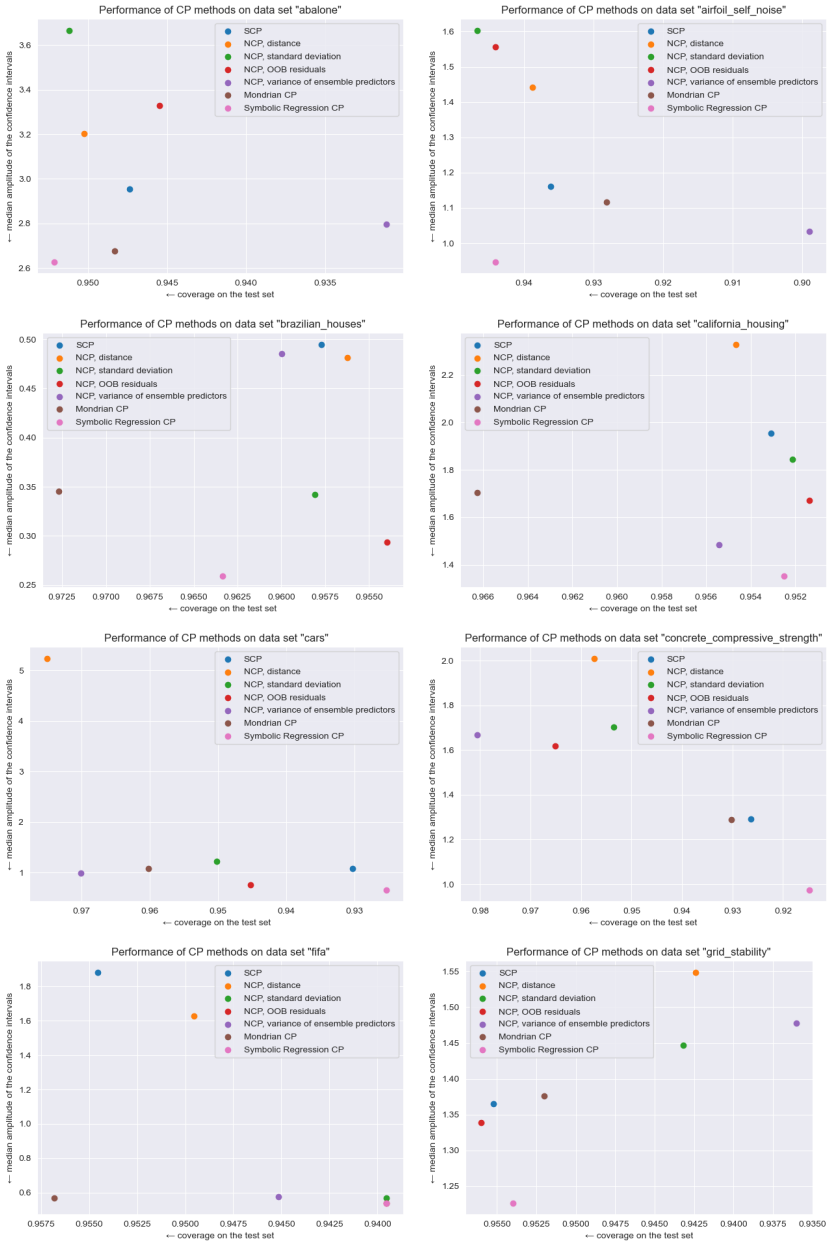

Fig. 1. Pareto fronts obtained for the different data sets (1).

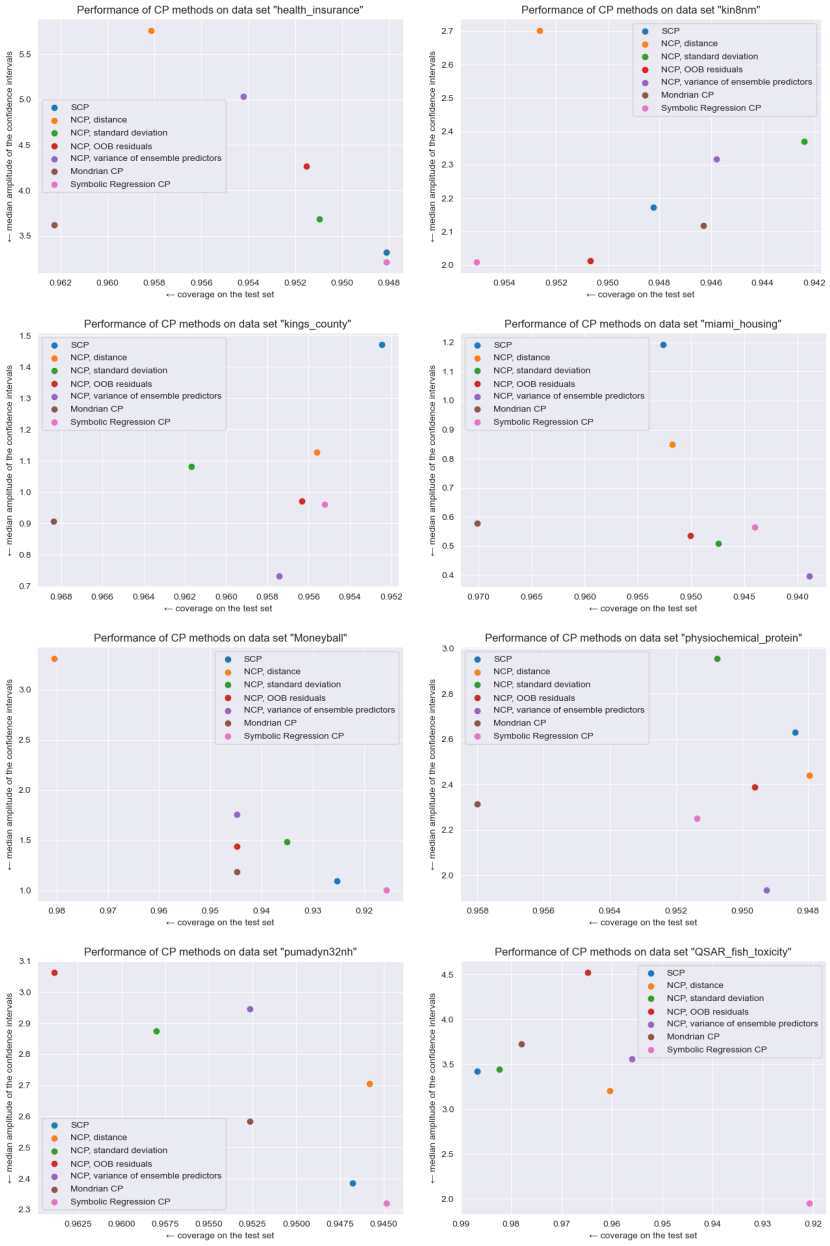

Fig. 2. Pareto fronts obtained for the different data sets (2).

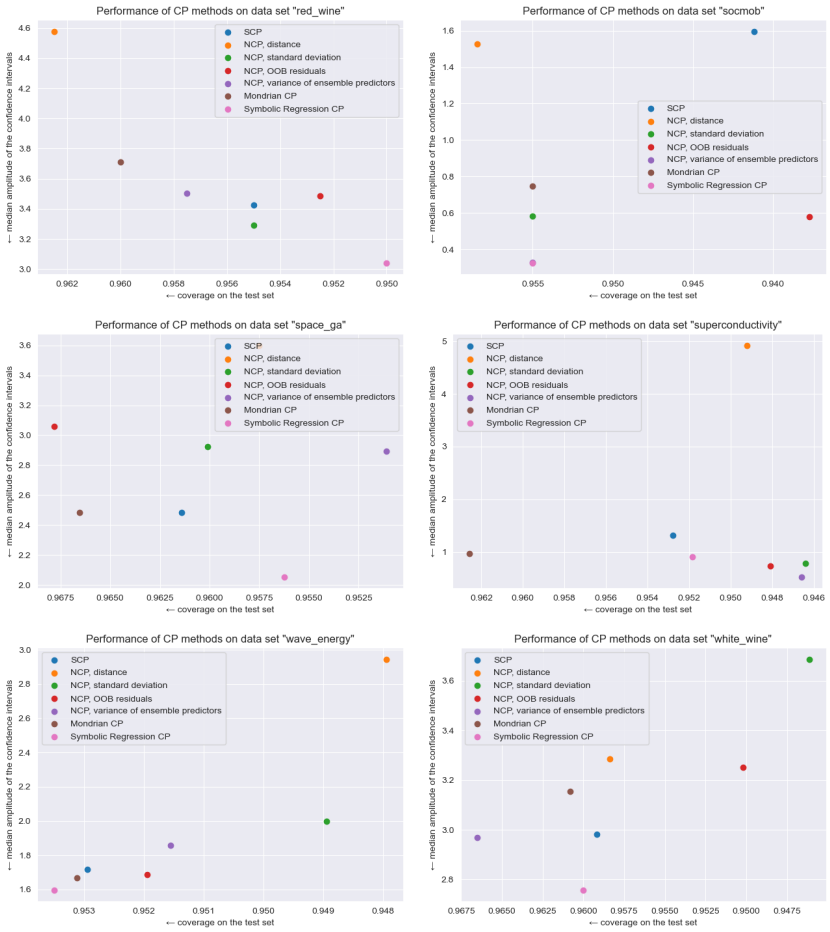

Fig. 3. Pareto fronts obtained for the different data sets (3).