

Orthogonal Gradient Boosting for Interpretable Additive Rule Ensembles

Supplementary Information

Anonymous Author(s)

ACM Reference Format:

Anonymous Author(s). 2023. Orthogonal Gradient Boosting for Interpretable Additive Rule Ensembles Supplementary Information. In *Proceedings of (KDD 2023)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/XXXXXXX>. XXXXXXXX

A PROOF OF THEOREM 4.3

The condition of Theorem 4.3 states that:

Let $\mathbf{Q} \in \mathbb{R}^{n \times (t-1)}$ be the selected query matrix, \mathbf{g} the corresponding gradient vector after a full weight correction, and \mathbf{q}^* be a maximizer of the **orthogonal gradient boosting objective** function defined by

$$\text{obj}_{\text{ogb}}(q) = \frac{|\mathbf{g}^T \mathbf{q}|}{\|\mathbf{q}_\perp\| + \epsilon}$$

where \mathbf{q}_\perp is the projection of q onto the orthogonal complement of range \mathbf{Q} .

A.1 Property a

PROPOSITION A.1. For $\epsilon \rightarrow 0$, $[\mathbf{q}_1, \dots, \mathbf{q}_{t-1}, \mathbf{q}^*]$ is the best approximation to $[\mathbf{q}_1, \dots, \mathbf{q}_{t-1}, \mathbf{g}]$.

PROOF. If $\epsilon \rightarrow 0$, then $\text{obj}_{\text{ogb}}(q) \rightarrow \frac{|\mathbf{g}^T \mathbf{q}|}{\|\mathbf{q}_\perp\|}$. If \mathbf{q}^* is a maximizer of obj_{ogb} , then as shown in Lemma 4.1, \mathbf{q}^* minimises the minimum distance from all

$$\mathbf{f} \in \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_{t-1}, \mathbf{g}\}$$

to the subspace of

$$\text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_{t-1}, \mathbf{q}^*\}.$$

Therefore, the subspace spanned by $[\mathbf{q}_1, \dots, \mathbf{q}_{t-1}, \mathbf{q}^*]$ is the best approximation to the subspace spanned by $[\mathbf{q}_1, \dots, \mathbf{q}_{t-1}, \mathbf{g}]$. \square

A.2 Property b

PROPOSITION A.2. For $\epsilon \rightarrow \infty$, \mathbf{q}^* is also a maximizer of obj_{gs} and any maximizer of obj_{gs} is also a maximizer of obj_{ogb} .

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).
KDD 2023, August 6-10, 2023, Long Beach, CA

© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXXX.XXXXXXX>

PROOF. Let q_1 and q_2 be any two queries and denote by $\text{obj}_{\text{ogb}}^{(\epsilon)}(q)$ the obj_{ogb} -value of q for a specific ϵ . Then

$$\begin{aligned} & \lim_{\epsilon \rightarrow \infty} \epsilon \left(\text{obj}_{\text{ogb}}^{(\epsilon)}(q_1) - \text{obj}_{\text{ogb}}^{(\epsilon)}(q_2) \right) \\ &= \lim_{\epsilon \rightarrow \infty} \epsilon \left(\frac{|\mathbf{g}^T \mathbf{q}_1|}{\|\mathbf{q}_1^\perp\| + \epsilon} - \frac{|\mathbf{g}^T \mathbf{q}_2|}{\|\mathbf{q}_2^\perp\| + \epsilon} \right) \\ &= \lim_{\epsilon \rightarrow \infty} \left(\frac{|\mathbf{g}^T \mathbf{q}_1|}{\|\mathbf{q}_1^\perp\|/\epsilon + 1} - \frac{|\mathbf{g}^T \mathbf{q}_2|}{\|\mathbf{q}_2^\perp\|/\epsilon + 1} \right) \\ &= |\mathbf{g}^T \mathbf{q}_1| - |\mathbf{g}^T \mathbf{q}_2| \\ &= \text{obj}_{\text{gs}}(q_1) - \text{obj}_{\text{gs}}(q_2) \end{aligned}$$

Thus for large enough ϵ , the signs of $\text{obj}_{\text{ogb}}^{(\epsilon)}(q_1) - \text{obj}_{\text{ogb}}^{(\epsilon)}(q_2)$ and $\text{obj}_{\text{gs}}(q_1) - \text{obj}_{\text{gs}}(q_2)$ agree. Therefore, a query q is a obj_{gs} -maximizer, i.e., $\text{obj}_{\text{gs}}(q) \geq \text{obj}_{\text{gs}}(q')$ for all $q' \in \mathbf{Q}$, if and only if q is a obj_{ogb} -maximizer, i.e., $\text{obj}_{\text{ogb}}(q) \geq \text{obj}_{\text{ogb}}(q')$ for all $q' \in \mathbf{Q}$. \square

A.3 Property c

PROPOSITION A.3. For $\epsilon = 0$ and $\|\mathbf{q}_\perp\| > 0$, the ratio $\left(\frac{\text{obj}_{\text{ogb}}(q)}{\text{obj}_{\text{gb}}(q)}\right)^2$ is equal to $1 + \left(\frac{\|\mathbf{q}_\parallel\|}{\|\mathbf{q}_\perp\|}\right)^2$.

PROOF. If $\epsilon = 0$ and $\|\mathbf{q}_\perp\| > 0$, then

$$\begin{aligned} \left(\frac{\text{obj}_{\text{ogb}}(q)}{\text{obj}_{\text{gb}}(q)} \right)^2 &= \frac{\frac{|\mathbf{g}^T \mathbf{q}|^2}{\|\mathbf{q}_\perp\|^2}}{\frac{|\mathbf{g}^T \mathbf{q}|^2}{\|\mathbf{q}\|^2}} = \frac{\|\mathbf{q}\|^2}{\|\mathbf{q}_\perp\|^2} \\ &= \frac{\|\mathbf{q}_\parallel\|^2 + \|\mathbf{q}_\perp\|^2}{\|\mathbf{q}_\perp\|^2} \\ &= 1 + \left(\frac{\|\mathbf{q}_\parallel\|}{\|\mathbf{q}_\perp\|} \right)^2 \end{aligned}$$

\square

A.4 Property d

PROPOSITION A.4. The objective value $\text{obj}_{\text{ogb}}(q)$ is upper bounded by $\|\mathbf{g}\|/(1 + \epsilon/n)$.

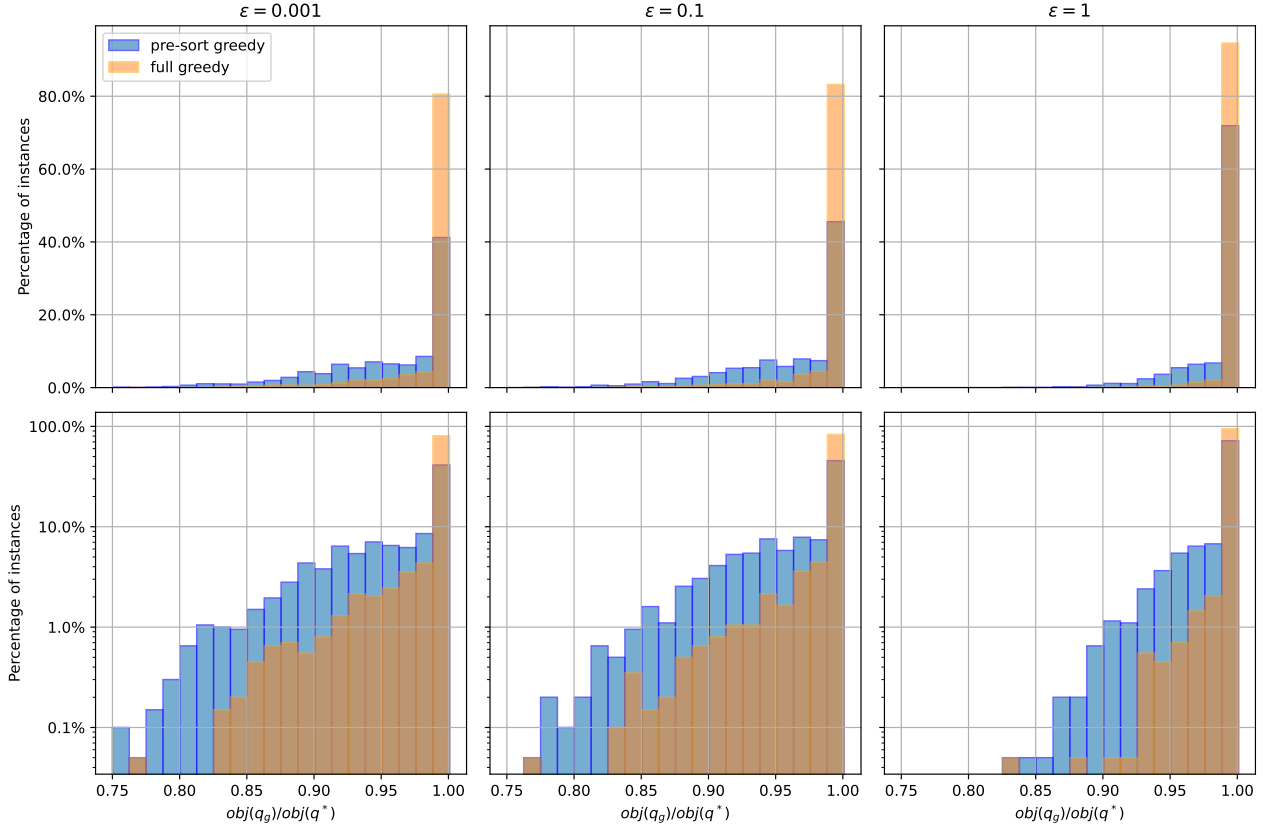


Figure 6: The number of instances of ratios between the best objective values obtained from the greedy search and the true optimal objective value. The upper figures are in linear scales and the lower figures are in log scales. The total variation distances for these three values of ϵ are 0.394, 0.377 and 0.227.

PROOF. If we divide the numerator and denominator of $\text{obj}_{\text{ogb}}(\mathbf{q})$ with $\|\mathbf{q}_{\perp}\|$, then we can get

$$\begin{aligned} \text{obj}_{\text{ogb}}(\mathbf{q}) &= \frac{|\mathbf{g}^T \mathbf{q}|}{\|\mathbf{q}_{\perp}\| + \epsilon} \\ &= \frac{\frac{|\mathbf{g}^T \mathbf{q}_{\perp}|}{\|\mathbf{q}_{\perp}\|}}{1 + \frac{\epsilon}{\|\mathbf{q}_{\perp}\|}} \end{aligned}$$

according to the Cauchy-Schwarz inequality, $\frac{|\mathbf{g}^T \mathbf{q}|}{\|\mathbf{q}_{\perp}\|} \leq \frac{\|\mathbf{g}\| \|\mathbf{q}_{\perp}\|}{\|\mathbf{q}_{\perp}\|} = \|\mathbf{g}\|$, so,

$$\text{obj}_{\text{ogb}}(\mathbf{q}) \leq \frac{\|\mathbf{g}\|}{1 + \frac{\epsilon}{\|\mathbf{q}_{\perp}\|}}$$

as $\|\mathbf{q}_{\perp}\|$ is upper bounded by the number of data points n ,

$$\text{obj}_{\text{ogb}}(\mathbf{q}) \leq \frac{\|\mathbf{g}\|}{1 + \frac{\epsilon}{n}}.$$

□

B GREEDY APPROXIMATION TO BOUNDING FUNCTION

The branch-and-bound search described in Section 3.3 requires an efficient way of calculating the value of $\text{bnd}(\mathbf{q}) = \max\{\text{obj}(\mathbf{q}') : \mathbf{q}' \leq \mathbf{q}, \mathbf{q}' \in \{0, 1\}^n\}$. It is too expensive to enumerate all possible \mathbf{q}' 's as there are 2^n cases in the worst case. One solution to this problem is that we can relax the constraint $\mathbf{q}' \in \{0, 1\}^n$ to $\mathbf{q}' \in [0, 1]^n$ and it can be solved by quadratic programming. However, this relaxation is too loose and inefficient. Instead, we consider relaxing the admission constraint and solve the problem using greedy algorithms.

Table 2: Comparison of Test Risks of Gradient Sum(S), Gradient boosting (G), XGBoost (X) and FCOGB (O) for benchmark datasets of classification (upper), regression (middle) and Poisson regression problems (lower).

DATASET	FEAT	ROW	\bar{R}_O	FCOGB vs. GS			FCOGB vs. GB			FCOGB vs XGBoost		
				Δ_{SO}^{bc}	$\bar{\Delta}_{SO}$	Δ_{SO}^{wc}	Δ_{GO}^{bc}	$\bar{\Delta}_{GO}$	Δ_{GO}^{wc}	Δ_{XO}^{bc}	$\bar{\Delta}_{XO}$	Δ_{XO}^{wc}
TITANIC	7	1043	.712	.074 (17.6)	.035	.000 (2.4)	.147 (2.4)	.025	-.025 (28.2)	.147 (2.4)	.022	-.015 (4.4)
TIC-TAC-TOE	27	958	.751	.174 (23.6)	.101	-.058 (1.4)	.111 (6.2)	.060	.000 (2.4)	.089 (16.8)	.030	-.013 (28.4)
IRIS	4	150	.552	.141 (5.8)	-.089	-.180 (7.2)	.149 (2.4)	-.083	-.294 (5)	.149 (2.4)	-.058	-.147 (25.8)
BREAST	30	569	.352	.024 (7.8)	-.055	-.229 (7)	.009 (7.8)	-.011	-.026 (26)	.100 (7.8)	.031	-.006 (20.2)
WINE	13	178	.368	.314 (2.4)	.020	-.270 (5.8)	.409 (6)	.135	.003 (4)	.433 (6)	.090	-.043 (27)
IBM HR	32	1470	.217	.019 (6.2)	.003	-.005 (8.2)	.034 (15.4)	.004	.000 (1.4)	.065 (4)	.012	.004 (1.4)
TELCO CHURN	18	7043	.688	.058 (2.4)	.005	-.159 (1.4)	.037 (23.4)	.019	-.032 (8.2)	.017 (9.4)	.006	-.009 (12.4)
GENDER	20	3168	.999	.003 (2.4)	.001	.002 (3.2)	.000 (4.2)	.000	.000 (2.4)	.003 (11.4)	.001	.000 (2.4)
BANKNOTE	4	1372	.355	.142 (19.6)	.055	-.075 (8.2)	.133 (19.6)	.024	-.079 (9.4)	.120 (19.6)	.049	-.049 (9.4)
LIVER	6	345	.999	-.012 (3.8)	-.093	-.195 (29.8)	.057 (3.8)	-.024	-.067 (15.4)	.057 (3.8)	-.066	-.164 (29.4)
MAGIC	10	19020	.710	.056 (8.2)	.018	-.037 (4.2)	.018 (15.4)	.007	.000 (5.2)	.017 (18.4)	.007	-.003 (25.4)
ADULT	11	30162	.619	.146 (2.4)	.007	-.191 (1.4)	.059 (10.4)	.018	.000 (2.4)	.058 (4.2)	.011	.004 (20.4)
DIGITS5	64	3915	.381	.030 (4.2)	.014	-.008 (3.2)	.009 (4.2)	-.031	-.058 (19.4)	.070 (4.2)	-.004	-.034 (19.4)
INSURANCE	6	1338	.163	.104 (7.2)	.017	-.567 (1.4)	.172 (4.2)	.018	-.011 (5.2)	.172 (4.2)	.020	-.011 (5.2)
FRIEDMAN1	10	2000	.083	.013 (4)	-.002	-.012 (3.2)	.025 (4)	.006	.000 (1.4)	.025 (4)	.005	.000 (1.4)
FRIEDMAN2	4	10000	.149	.165 (3.2)	-.021	-.612 (1.4)	.084 (10.4)	.019	-.068 (5.2)	.080 (8.2)	.016	-.068 (5.2)
FRIEDMAN3	4	5000	.060	.009 (4.6)	.003	-.012 (4)	.021 (4.6)	.002	.000 (1.4)	.021 (4.6)	.002	.000 (1.4)
WAGE	5	1379	.419	.017 (6.6)	-.021	-.048 (4.6)	.065 (6.6)	.000	-.027 (18.2)	.065 (6.6)	-.002	-.014 (21.4)
DEMOGRAPHICS	13	6876	.229	.011 (5.2)	.003	.000 (1.4)	.007 (3.2)	.002	.000 (1.4)	.007 (3.2)	.002	.000 (1.4)
GDP	1	35	.038	.003 (3.2)	.000	-.001 (5.2)	.003 (3.2)	.000	-.001 (5.2)	.003 (3.2)	.000	-.001 (5.2)
USED CARS	4	1770	.198	.178 (3.2)	-.019	-.549 (1.4)	.113 (14)	.055	.000 (3.2)	.089 (8.8)	.035	.000 (3.2)
DIABETES	10	442	.169	.026 (4.4)	-.004	-.033 (3.4)	.058 (4.4)	-.002	-.011 (29.8)	.058 (4.4)	.002	-.008 (29.4)
BOSTON	13	506	.097	.019 (4.4)	-.001	-.025 (3.8)	.044 (4.4)	.006	-.011 (8.6)	.044 (4.4)	.006	-.011 (9.2)
WORLD HAPPINESS	8	315	.051	.010 (5.2)	-.003	-.012 (3.8)	.013 (5.2)	.002	-.010 (4.4)	.023 (5.2)	.002	-.001 (23.6)
LIFE EXPECTANCY	21	1649	.041	.003 (4.2)	.000	-.001 (8.2)	.007 (4.2)	.001	.000 (20.4)	.007 (4.2)	.001	-.001 (26.4)
MOBILE PRICES	20	2000	.168	.168 (2.4)	-.008	-.648 (1.4)	.058 (3.2)	.002	-.004 (8.2)	.058 (3.2)	.004	.000 (4.2)
SUICIDE RATE	5	27820	.534	.081 (2.4)	.016	-.333 (1.4)	.018 (13.4)	.008	.000 (2.4)	.024 (11.4)	.009	.000 (2.4)
VIDEO GAMES	6	16327	.723	.000 (4.2)	.000	.000 (3.2)	.000 (10.4)	.000	.000 (3.2)	.000 (10.4)	.000	.000 (3.2)
RED WINE	11	1599	.048	.003 (4.2)	.000	-.002 (3.2)	.004 (4.2)	.001	.000 (1.4)	.004 (4.2)	.001	.000 (7.4)
COVID VIC	4	85	.185	.068 (3)	.030	-.062 (10)	.103 (10.6)	-.005	-.692 (2.8)	3.448 (3)	.418	.002 (25.8)
COVID	2	225	.515	.071 (6.8)	-.037	-.399 (2)	.279 (3.8)	.025	-.003 (27)	25.66 (6.8)	5.388	.290 (26.8)
BICYCLE	4	122	.505	.213 (7.8)	-.021	-.333 (3.4)	.054 (4.6)	-.035	-.337 (4)	.011 (7.8)	-.025	-.061 (22.2)
SHIPS	4	34	.338	.076 (3.2)	-.089	-.498 (2.4)	.160 (11)	.015	-.066 (23.2)	993.4 (3.2)	181.3	.605 (29.6)
SMOKING	2	36	.216	-.017 (16.6)	-.046	-.189 (6.6)	.075 (10.2)	.029	-.043 (4.2)	.950 (2.4)	.124	.035 (14.2)

A full greedy approach can be used to approximate the maximum objective value of the subset of data points selected by \mathbf{q} , which is the bounding value $\text{bnd}(\mathbf{q})$. Given a query $\mathbf{q}'^{(t-1)} \leq \mathbf{q}$, we need to find the data point selected by \mathbf{q} which maximise the objective function, and use it with $\mathbf{q}'^{(t-1)}$ to form a $\mathbf{q}'(t)$.

$$i_*^{(t)} = \arg \max_{i \in I(\mathbf{q}) - I(\mathbf{q}'^{(t-1)})} \frac{\mathbf{g}^T(\mathbf{q}'^{(t-1)} + \mathbf{e}_i)}{\|(\mathbf{q}'^{(t-1)} + \mathbf{e}_i)_{\perp}\| + \epsilon}.$$

where $I(\mathbf{q}) = \{i : \mathbf{q}(x_i) = 1, 1 \leq i \leq n\}$, $0 \leq t \leq |I(\mathbf{q})|$, $\mathbf{q}'^{(0)} = \mathbf{0}$ and $\mathbf{q}'^{(t)} = \mathbf{q}'^{(t-1)} + \mathbf{e}_{i_*^{(t)}}$. We use the maximum value of $\text{obj}(\mathbf{q}'^{(t)})$ as the bounding value for query \mathbf{q} . The computation time complexity level of this approach is $O(n^2)$ for each query, which is not as efficient as the presorting greedy approach described in Section 4.3.

The presorting greedy approach of solving the prefix optimization problem described in Section 4.3 leads to another approximation to the optimal objective function value for the queries which

cover subsets of data points covered by \mathbf{q} . As proved in Theorem 4.4, this approach has a time complexity of $O(tn)$.

We test 2000 instances for different initial queries and initial gradient values to see the difference between the approximation of $\text{bnd}(\mathbf{q})$ obtained by the full greedy approach, the pre-sorting greedy approach, and the actual optimal objective values (obtained by a brute-force approach). We choose three different values of ϵ : 0.001, 0.1 and 1.

Figure 6 compares the ratio between the approximations to $\text{bnd}(\mathbf{q})$ obtained by the two greedy approaches and the true optimal objective value. The Y axis of Figure 6 represents the percentage of instances of different ratios.

According to the comparison, the full greedy approach can approximate the true bounding function better than the presorting greedy approach. For smaller ϵ values ($\epsilon = 0.001$), there are 90% instances whose approximation values are more than 90% of the true bounding function values, while 96% of instances approximate more than 90% of the value of $\text{bnd}(\mathbf{q})$ using the full greedy approach.

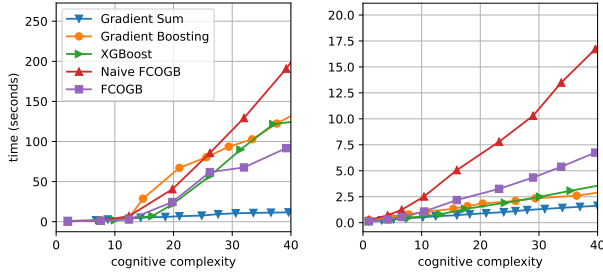


Figure 7: Comparison of the computation time of Gradient boosting, XGBoost and FCOGB for the benchmark datasets breast cancer and diabetes of generating a rule ensemble with cognitive complexity of 40.

For $\epsilon = 0.1$, both algorithms have slight better (both 1% promotion) approximation than $\epsilon = 0.001$. It can be observed that for $\epsilon = 1$, both algorithms have more instances where the approximations are closed to the true bounding values. However, if the value of ϵ is too large, then the calculated objective values cannot be accurate according to Theorem 4.3. Comparing the statistical distances of these two greedy approaches, it is reasonable to use the presorting greedy approach to approximate the bounding values.

To approximate the true bounding function more efficiently and more accurate, we adopt the presorting greedy approach in this research.

C COMPARISON OF TESTING RISKS

Table 2 compares the testing empirical risks of the same benchmark datasets in Section 5. Table 2 has the same format with Table 1 except that it does not contain the comparison of running times.

As discussed in Section 5, the test performance of these algorithms follows similar trends to the training performance.

D COMPARISON OF COMPUTATION TIME IN TERMS OF COGNITIVE COMPLEXITY

Figure 7 shows the computation time of all algorithms generating rule ensembles with a cognitive complexity level of 40 for the benchmark datasets breast cancer and diabetes.

E COMPARISON OF GREEDY FCOGB AND OPTIMAL (BRANCH-AND-BOUND) FCOGB

Table 3 compares the average normalised risks of generating rule ensembles with cognitive complexity levels of 30 by Gradient Sum (GS), Gradient boosting (GB), XGBoost, Greedy FCOGB and FCOGB using Branch-and-bound search (BB FCOGB). This table also compares the computation times of these algorithms of generating 10 rules.

Table 3: Comparison of average normalised risks and computation times of Gradient Sum (GS), Gradient Boosting (GB), XGBoost, FCOGB using Branch-and-bound (BB FCOGB) and FCOGB using Greedy search (Greedy FCOGB)

DATASET	d	n	AVERAGE NORMALISED RISKS					COMPUTATION TIMES				
			GS	GB	XGBoost	BB FCOGB	GREEDY FCOGB	GS	GB	XGBoost	BB FCOGB	GREEDY FCOGB
TITANIC	7	1043	.702	.675	.694	.658	.679	1.660	20.87	26.81	26.44	1.799
TIC-TAC-TOE	27	958	.805	.693	.744	.656	.735	5.570	29.29	23.06	47.68	2.012
IRIS	4	150	.359	.344	.379	.317	.295	.785	2.010	2.075	2.090	.441
BREAST	30	569	.347	.361	.391	.358	.359	8.505	116.2	177.8	257.3	3.818
WINE	13	178	.286	.303	.353	.241	.305	2.015	7.713	1.151	30.09	.935
IBM HR	32	1470	.592	.215	.596	.212	.578	15.67	7.488	20.25	11.44	5.078
TELCO CHURN	18	7043	.707	.714	.703	.695	.694	22.45	5266	5190	5343	28.89
GENDER	20	3168	.997	.997	.997	.997	.997	6.543	42.66	32.38	28.47	7.031
BANKNOTE	4	1372	.394	.354	.406	.337	.356	6.492	10.08	13.83	6.991	3.103
LIVER	6	345	.855	.833	.860	.837	.862	2.015	145.2	301.4	234.5	1.364
MAGIC	10	19020	.744	.729	.737	.731	.733	73.11	71.10	194.2	138.9	78.03
ADULT	11	30162	.620	.626	.614	.608	.600	100.9	392.0	381.7	347.5	156.6
DIGITS5	64	3915	.413	.385	.401	.393	.364	60.38	110.7	37.98	125.8	38.84
INSURANCE	6	1338	.193	.192	.190	.166	.169	10.90	19.50	37.62	28.48	3.110
FRIEDMAN1	10	2000	.096	.084	.102	.080	.093	2.795	5.347	12.38	6.220	1.242
FRIEDMAN2	4	10000	.188	.165	.252	.149	.223	22.89	19.723	30.60	12.65	5.716
FRIEDMAN3	4	5000	.061	.060	.060	.057	.057	8.313	12.766	27.42	12.37	2.836
WAGE	5	1379	.396	.394	.401	.381	.386	9.077	21.263	29.60	20.64	3.158
DEMOGRAPHICS	13	6876	.229	.227	.227	.225	.225	36.37	21.707	49.56	30.39	14.56
GDP	1	35	.038	.038	.038	.037	.037	.086	.263	.429	.351	.094
USED CARS	4	1770	.171	.224	.243	.191	.190	13.23	17.40	34.90	12.46	4.121
DIABETES	10	442	.157	.160	.159	.153	.151	2.598	5.813	8.926	7.529	1.567
BOSTON	13	506	.111	.110	.112	.103	.104	3.942	8.186	16.226	8.591	1.543
HAPPINESS	8	315	.047	.049	.048	.047	.045	.561	13.79	26.37	24.02	.609
LIFE EXPECT.	21	1649	.040	.041	.041	.040	.040	20.28	55.46	144.6	150.6	7.145
MOBILE PRICES	20	2000	.157	.166	.166	.163	.162	21.67	775.4	1736	1101	7.738
SUICIDE RATE	5	27820	.578	.562	.562	.553	.562	74.23	161.1	45.59	178.8	95.55
VIDEOGAME	6	16327	.954	.954	.954	.954	.954	20.21	58.91	14.21	37.72	18.16
RED WINE	11	1599	.048	.048	.049	.048	.048	3.736	30.03	17.72	37.25	3.924
COVID VIC	4	85	.202	.192	.540	.178	.168	.580	.729	1.502	.587	.247
COVID	2	225	.386	.419	4.133	.408	.422	1.032	1.026	2.153	.832	.420
BICYCLE	4	122	.437	.405	.416	.401	.397	.792	1.897	3.536	1.768	.421
SHIPS	4	34	.212	.255	129.2	.223	.203	.265	.367	.597	.362	.141
SMOKING	2	36	.171	.155	.260	.142	.145	.191	.341	.484	.205	.104