Can XGBoost Learn Ratio and Product Features From Raw Inputs Under Distribution Shift?

Anonymous Author

Abstract

Many predictive tasks in tabular data use engineered features such as ratios or products. These features can encode domain invariances. This paper studies whether a modern gradient boosted tree ensemble, XGBoost, learns these structures from raw positive inputs. This paper focuses on robustness under distribution shift.

This paper builds a synthetic benchmark with binary classification tasks of increasing complexity, from Level 1 to Level 7. Each task depends on a latent ratio or product signal, or on a composition of such signals. This paper trains XGBoost models in two modes. The first mode uses only raw inputs. The second mode adds oracle features that contain either intermediate ratio or product coordinates, or the final latent signal.

Across experiments, the raw feature model often fails to recover the intended structure under shifts. The gap is largest on tasks that require multiplicative compositions. For example, on Level 4 tasks, a representative configuration reaches median test PRAUC ≈ 0.11 with raw inputs and PRAUC ≈ 0.24 with oracle features. Under shifts that preserve the true ratio or product while changing input scales, failures can be extreme. In one preserve shift case, the raw feature model drops to PRAUC ≈ 0.08 while the oracle model reaches PRAUC ≈ 0.91. This paper also reports diagnostics that measure invariance error and iso coordinate variance. Oracle features reduce these diagnostics by large factors. These results show that representational capacity does not imply reliable invariant learning in finite sample tabular settings.

1 Introduction

Practitioners often face a design choice. They can rely on a model to discover feature relations. They can also encode known relations as features. Ratios and products are common examples. Finance uses ratios such as debt to income. Physical systems use dimensionless products.

XGBoost is widely used for tabular prediction. It can approximate many nonlinear functions. However, approximation does not guarantee robust generalization. A model can fit training data using incidental cues. This matters under shift. A scale shift can change feature magnitudes while leaving the true ratio unchanged. A robust model should not degrade in that case.

This paper asks a targeted question. Can XGBoost trained on raw features learn ratio or product structure in a way that remains stable under shifts that preserve that structure?

2 Problem setup

Let $x \in \mathbb{R}^d_{>0}$ denote raw features. A task defines a latent signal $s = f(x)$. The label is a noisy function of s. This paper uses a logistic link for most tasks:

$$\mathbb{P}(y = 1 \mid x) = \sigma\big(\beta(s - t)\big), \tag{1}$$

where $\sigma(z) = (1 + e^{-z})^{-1}$, $\beta > 0$, and t controls prevalence.

This paper compares two feature modes. The raw mode uses only x. The oracle mode augments x with features derived from the data generating process. This paper considers two oracle variants. The first variant adds intermediate coordinates, such as $u = x_0/x_1$. The second variant adds the final signal s.

This paper measures performance with test PRAUC. This paper also measures a performance gap:

$$\Delta\text{PRAUC} = \text{PRAUC}_{\text{oracle}} - \text{PRAUC}_{\text{raw}}. \tag{2}$$

3 Synthetic benchmark

3.1 Task families

The benchmark contains task levels that increase structural complexity.

- Level 1 uses a single ratio or a single product, such as $s = x_0/x_1$ or $s = x_0x_1$.
- Level 2 uses ratios or products of sums, such as $s = (x_0 + x_1)/(x_2 + x_3)$.
- Level 3 uses ratios or products of differences, such as $s = (x_0 - x_1)/(x_2 - x_3)$.
- Level 4 uses multiplicative compositions of coordinates, such as $s = (x_0/x_1)/(x_2/x_3)$ or $s = (x_0x_1)(x_2x_3)$.
- Level 5 mixes ratios and products, such as $s = (x_0/x_1)/(x_2x_3)$.
- Level 6 applies nonmonotone transforms of a basic ratio or product, such as a band pass function of x_0/x_1.
- Level 7 uses gated combinations, where a gate selects between two latent coordinates.

Appendix A gives concrete task definitions.

3.2 Distribution regimes and shifts

This paper generates features under several regimes. Some regimes are static and only change tails or correlations. Other regimes impose explicit train test shifts.

This paper includes a preserve shift that changes raw magnitudes while preserving the ratio or product signal. For ratio tasks, this shift multiplies both numerator and denominator features by a common factor. For product tasks, this shift applies inverse scaling that keeps the product constant. Appendix B defines each regime.

3.3 Experimental protocol

This paper uses a fixed train validation test split within each dataset. The main experiment uses $n = 30{,}000$ training examples and a similarly sized test set. This paper repeats each configuration over three random seeds. Across tasks, regimes, and capacity settings, this yields 1,944 trained models in the main run.

This paper uses three feature modes. The first mode uses raw features only. The second mode adds intermediate oracle coordinates. The third mode adds the final oracle signal.

Level	Raw med PRAUC	Oracle med PRAUC	Median ΔPRAUC	10 to 90 pct range	Runs
1	0.205	0.239	0.008	[-0.010, 0.099]	36
2	0.132	0.157	0.014	[0.000, 0.067]	36
3	0.211	0.330	0.047	[0.006, 0.153]	36
4	0.116	0.246	0.052	[0.009, 0.196]	36
5	0.376	0.477	0.062	[0.018, 0.415]	18
6	0.054	0.065	0.008	[0.000, 0.022]	36
7	0.333	0.346	0.016	[-0.001, 0.041]	18

Table 1: Main experiment results by task level for the baseline model with the oracle signal setting. Each row reports medians across runs. The paired ΔPRAUC is computed per run and then aggregated. Therefore, the median ΔPRAUC can differ from the difference between the two medians.

3.4 Models

This paper trains XGBoost classifiers with a small set of capacity settings. The main settings vary the maximum tree depth. This paper uses a validation set for early stopping. Appendix C lists the training configuration.

4 Results

4.1 Overall performance gap

Oracle features improve performance in most settings. The gap grows with task complexity. Level 4 tasks show the largest median gap.

In the main experiment, a representative configuration reaches median test PRAUC ≈ 0.11 in the raw mode and PRAUC ≈ 0.24 in the oracle mode on Level 4 tasks. This corresponds to a relative drop of about 53% for the raw feature model under this aggregation.

Table 1 summarizes median gaps by level for one baseline configuration.

4.2 Preserve shifts expose noninvariant learning

A preserve shift should not hurt a model that uses the correct ratio or product. However, preserve shifts cause the largest failures for the raw feature model.

In one Level 4 preserve shift case, the raw feature model reaches PRAUC ≈ 0.08 while the oracle model reaches PRAUC ≈ 0.91. This gap indicates that the raw feature model relied on scale dependent cues. Those cues did not transfer.

Across tasks, preserve shifts produce the largest median ΔPRAUC. Static regimes such as heavy tail and correlation regimes produce much smaller gaps. This paper also observes rare cases where the oracle mode is slightly worse. These cases are small in magnitude. Appendix E includes a breakdown by regime family.

4.3 Diagnostics align with performance gaps

This paper uses diagnostics that test invariance directly. This paper also tests iso coordinate variation. For a ratio, this paper scales the numerator and denominator together. This keeps the ratio fixed. A perfectly invariant predictor should not change.

In the main experiment, the median invariance error for raw models is on the order of 10^{-2}. The median invariance error for oracle models is on the order of 10^{-3}. The iso coordinate variance shows a similar pattern. Raw models show median values around 2.8×10^{-4} for ratio tasks and 3.0×10^{-4} for product tasks. Oracle models reduce this to about 10^{-5} or lower.

These diagnostics explain the failure under preserve shifts. A model can fit the training distribution without learning the invariant coordinate. A preserve shift then removes shortcut cues while keeping the signal fixed.

5 Discussion

The results support three practical points.

First, XGBoost does not reliably learn ratio and product invariances from raw inputs in finite sample settings. This is most visible on multiplicative compositions. Level 4 tasks show the largest gaps.

Second, preserve shifts form a targeted stress test. They isolate whether a model has learned the invariant structure. Large failures under preserve shifts indicate shortcut learning.

Third, oracle features improve interpretability. When the model uses the intended coordinate, feature attribution aligns with domain reasoning. The invariance diagnostics also become small.

6 Limitations

This study has several limitations.

This paper evaluates only one model class. The experiments focus on XGBoost. Other architectures may behave differently.

The experiments use synthetic tasks. They offer ground truth structure. They may not capture all complexities of real data.

The experiments use a limited set of regimes and random seeds. The experiments also fix the main sample size in most experiments. Appendix **??** expands this discussion.

7 Future work

Several extensions can strengthen the evidence.

A dataset size sweep is a natural next step. This can separate data efficiency from asymptotic learnability.

Evaluating other model classes is also a natural next step. Neural networks, random forests, and symbolic approaches are natural candidates.

Testing invariance inducing training methods is also a natural next step. Data augmentation and explicit invariance penalties can reduce reliance on oracle features.

8 Conclusion

This paper studies ratio and product feature learning in XGBoost under distribution shift. This paper introduces a synthetic benchmark that isolates known invariances. Results show that raw feature models often fail on higher complexity tasks. They also fail under preserve shifts that should be harmless. Oracle features close large gaps in PRAUC and reduce invariance diagnostics.

These findings support a conservative modeling practice. When domain knowledge suggests a ratio or product feature, it is risky to assume that a tree ensemble will learn it robustly from raw inputs. Explicit feature construction remains a strong baseline for robustness.

A Task definitions

This appendix lists representative task definitions. All tasks use positive raw features.

A.1 Level 1

Ratio:

$$s = \frac{x_0}{x_1}. \tag{3}$$

Product:

$$s = x_0 x_1. \tag{4}$$

A.2 Level 2

Ratio of sums:

$$s = \frac{x_0 + x_1}{x_2 + x_3}. \tag{5}$$

Product of sums:

$$s = (x_0 + x_1)(x_2 + x_3). \tag{6}$$

A.3 Level 3

Ratio of differences:

$$s = \frac{x_0 - x_1}{x_2 - x_3 + \epsilon}, \tag{7}$$

where $\epsilon > 0$ avoids division by zero.

A.4 Level 4

Ratio of ratios:

$$s = \frac{(x_0/x_1)}{(x_2/x_3)}. \tag{8}$$

Product of products:

$$s = (x_0 x_1)(x_2 x_3). \tag{9}$$

A.5 Level 7

A simple gated task uses a gate $g \in \{0, 1\}$ and two latent coordinates u_1 and u_2. The final signal is:

$$s = (1 - g)u_1 + g u_2. \tag{10}$$

B Distribution regimes

This paper generates x from distributions that vary tails, correlation, and train test shift. This paper uses lognormal features in most regimes:

$$x_i = \exp(z_i), \quad z \sim \mathcal{N}(\mu, \Sigma). \tag{11}$$

This paper varies the marginal variance through the diagonal of Σ. This paper induces correlation through off diagonal entries.

This paper uses a mixture regime that samples from a low variance component and a high variance component.

This paper defines two shift families.

Naive shift changes raw features in a way that changes the ratio or product distribution. Preserve shift changes raw features while keeping the latent signal fixed by construction.

C XGBoost configuration

This paper uses standard binary logistic objectives. This paper varies tree depth across a small grid. This paper uses early stopping on a validation split. This paper keeps other settings fixed within an experiment.

D Diagnostics

This appendix defines invariance error and iso coordinate variance.

D.1 Invariance error

Let $p(x)$ denote the model predicted probability for class 1. Let T_c denote a transformation that preserves the latent signal. For a ratio x_0/x_1, one example is:

$$T_c(x)_0 = cx_0, \quad T_c(x)_1 = cx_1, \tag{12}$$

with other coordinates unchanged.

This paper defines an invariance error at scale c as:

$$e_{\text{inv}}(x; c) = |p(T_c(x)) - p(x)|. \tag{13}$$

This paper reports the median of $e_{\text{inv}}(x; c)$ over a test set and over a small grid of c values.

D.2 Iso coordinate variance

Fix a latent coordinate value. For a ratio, this paper can move along a curve that keeps x_0/x_1 constant:

$$x'_0 = cx_0, \quad x'_1 = cx_1. \tag{14}$$

This paper computes the variance of $p(x')$ over several c values. This paper then aggregates over the test set.

XGB config	Oracle setting	Raw med PRAUC	Oracle med PRAUC	Median ΔPRAUC	10 to 90 pct range
baseline	coords only	0.157	0.230	0.014	[-0.004, 0.132]
baseline	signal only	0.157	0.236	0.020	[0.000, 0.135]
depth 4 light	coords only	0.166	0.231	0.015	[-0.002, 0.135]
depth 4 light	signal only	0.166	0.235	0.019	[-0.004, 0.142]
depth 7 high capacity	coords only	0.152	0.218	0.013	[-0.001, 0.132]
depth 7 high capacity	signal only	0.152	0.226	0.019	[-0.001, 0.138]

Table A1: Overall robustness summary across model configurations and oracle settings. Each row aggregates all runs in the experiment.

Level	Ratio inv err raw	Ratio inv err oracle	Product inv err raw	Product inv err oracle	Iso var ratio raw	Iso var ratio oracle	Iso var product raw	Iso var product oracle
Level 1	0.010923	0.002065	0.013462	0.004033	0.00019406	0.00001061	0.00052064	0.00004963
Level 2	0.009471	0.001931	0.013817	0.004652	0.00013698	0.00000880	0.00042177	0.00006944
Level 3	0.014673	0.002937	0.016298	0.002368	0.00061243	0.00001854	0.00053737	0.00001905
Level 4	0.002637	0.000536	0.012852	0.002605	0.00002746	0.00000286	0.00161728	0.00003784
Level 5	0.012526	0.002334	0.006157	0.000719	0.00057113	0.00002592	0.00017782	0.00000472
Level 6	0.000210	0.000010	0.003098	0.000585	0.00000039	0.00000011	0.00001350	0.00000326
Level 7	0.028697	0.031254	0.018731	0.016002	0.00321148	0.00300330	0.00110973	0.00111752

Table A2: Invariance diagnostics by level for the baseline model with the oracle signal setting. Ratio and product invariance errors are computed under scale transformations. Iso variance is computed from the variance of predicted probabilities over a transformation grid.

E Additional tables and figures

E.1 Compact result summaries

E.2 Figures

E.3 Supplementary CSV artifacts

The full run level results and derived deltas are provided as CSV artifacts. This paper does not typeset them because they exceed the table length constraint. The filenames are exp_default__runs.csv and exp_default__deltas.csv.

Regime family	Raw med PRAUC	Oracle med PRAUC	Median ΔPRAUC	10 to 90 pct range	Runs
mixture extremes	0.278	0.321	0.025	[0.004, 0.070]	36
shift naive	0.295	0.362	0.023	[-0.008, 0.092]	36
shift preserve	0.126	0.364	0.151	[0.019, 0.418]	36
tail corr	0.137	0.169	0.011	[-0.004, 0.057]	108

Table A3: Results by regime family for the baseline model with the oracle signal setting.

Task	Level	n	Raw med PRAUC	Oracle med PRAUC	Median ΔPRAUC	10 to 90 pct range	Runs
l1 product	1	30000	0.442	0.455	0.005	[-0.010, 0.303]	18
l1 ratio	1	30000	0.148	0.213	0.012	[-0.007, 0.087]	18
l2 product of sums	2	30000	0.394	0.417	0.012	[-0.003, 0.152]	18
l2 ratio of sums	2	30000	0.091	0.129	0.022	[0.000, 0.054]	18
l3 product diff	3	30000	0.197	0.305	0.040	[0.006, 0.154]	18
l3 ratio diff	3	30000	0.227	0.341	0.050	[0.009, 0.151]	18
l4 product x product	4	30000	0.804	0.903	0.018	[0.005, 0.348]	18
l4 ratio x ratio	4	30000	0.068	0.202	0.112	[0.025, 0.165]	18
l5 ratio x product	5	30000	0.376	0.477	0.062	[0.018, 0.415]	18
l6 nonmonotone product	6	30000	0.061	0.067	0.005	[-0.002, 0.015]	18
l6 nonmonotone ratio	6	30000	0.052	0.064	0.012	[0.003, 0.024]	18
l7 gated ratio vs product	7	30000	0.333	0.346	0.016	[-0.001, 0.041]	18

Table A4: Results by task for the baseline model with the oracle signal setting.

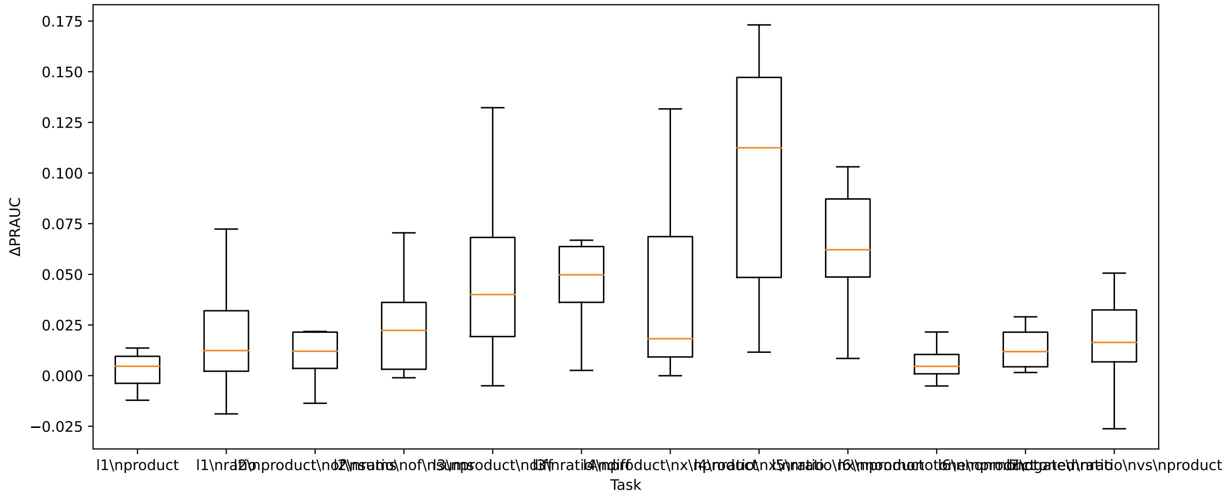

Figure A1: Distribution of paired ΔPRAUC across runs, grouped by task, for the baseline model with the oracle signal setting.

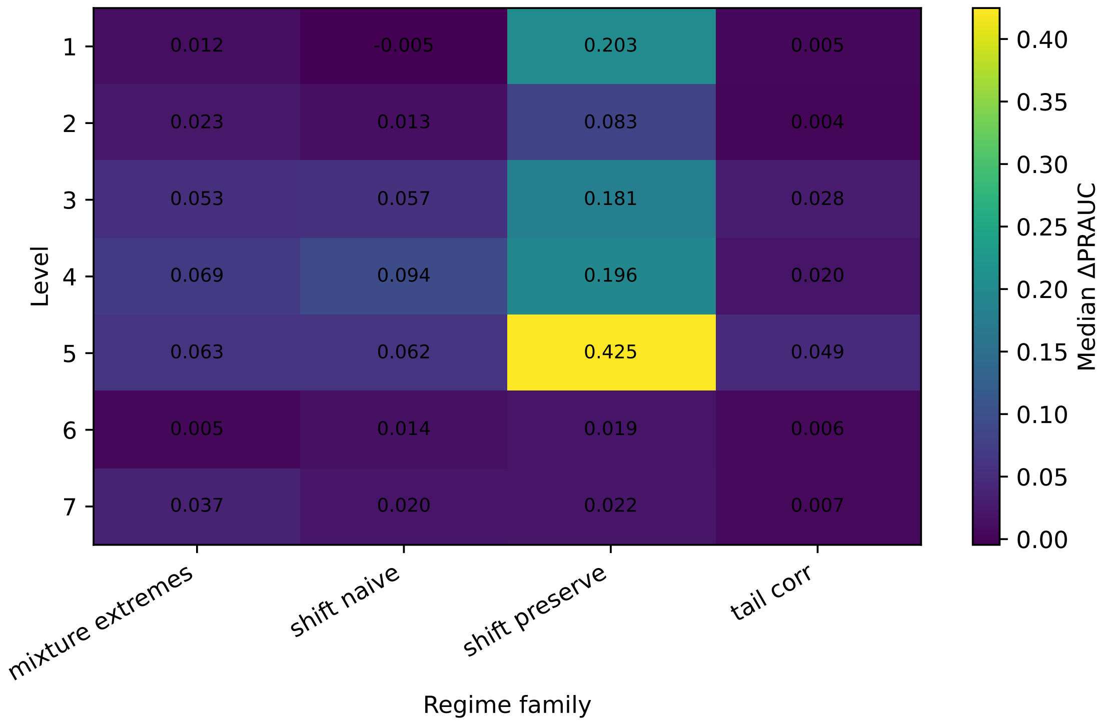

Figure A2: Median paired ΔPRAUC by task level and regime family for the baseline model with the oracle signal setting.

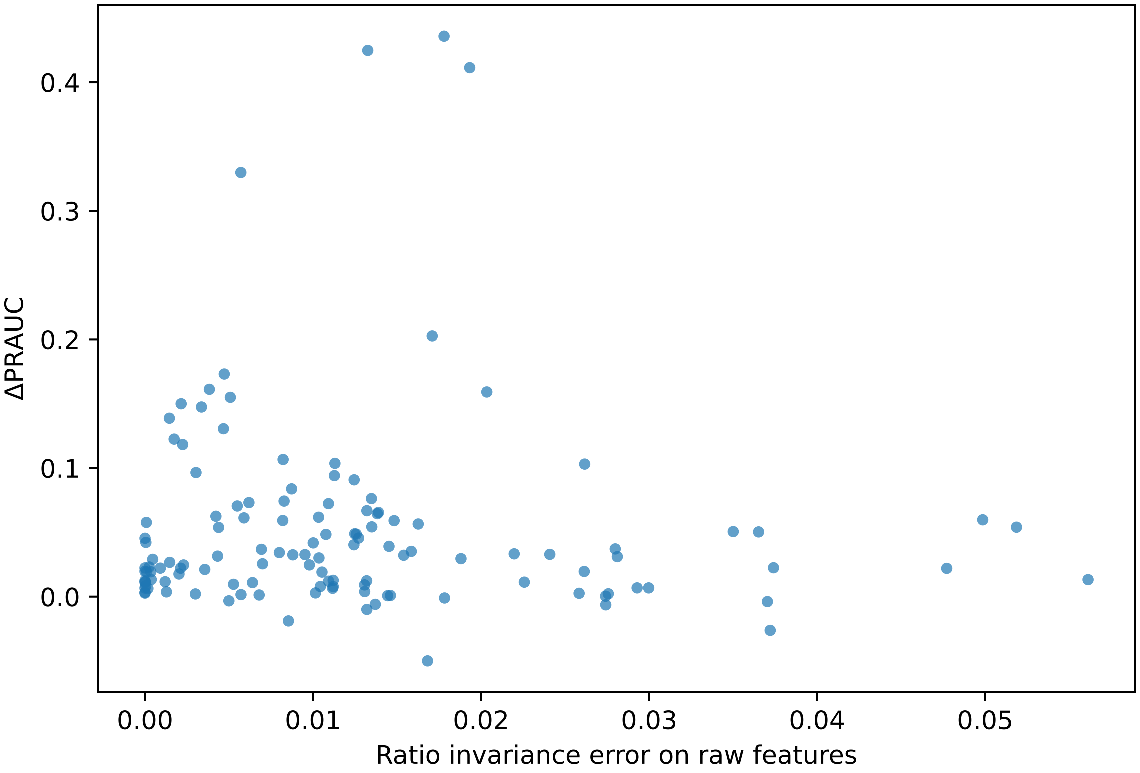

Figure A3: Relationship between ratio invariance error on raw features and paired ΔPRAUC across runs for the baseline model with the oracle signal setting.