Multi-Target XGBoostLSS Regression

Alexander März^ℵ.

Abstract—Current implementations of Gradient Boosting Machines are mostly designed for single-target regression tasks and commonly assume independence between responses when used in multivariate settings. As such, these models are not well suited if non-negligible dependencies exist between targets. To overcome this limitation, we present an extension of XGBoostLSS that models multiple targets and their dependencies in a probabilistic regression setting. Empirical results show that our approach outperforms existing GBMs with respect to runtime and compares well in terms of accuracy.

 $Keywords: Compositional\ Data\ Analysis \cdot Multi-Target\ Distributional\ Regression \cdot Probabilistic\ Modelling \cdot XGBoostLSS$

I. Introduction

The recent M5 forecasting competition demonstrated that tree-based models are highly competitive beyond cross-sectional tabular data. Yet, despite their wide-spread use, when applied in a multivariate setting, current implementations of Gradient Boosting Machines (GBMs) commonly assume (conditional) independence between target variables. However, modelling of mutual dependencies in a probabilistic regression setting has been shown to increase accuracy and to also lead to added insight into the data generating process (Schmid et al., 2023). Therefore, models tailored to single-target regression tasks are not well suited when applied in environments where non-negligible inter-target co-relations exist.

While high-dimensional multivariate forecasting is an active area of research in Deep Learning (see, e.g., Kan et al. (2022); Rasul et al. (2021a,b); Wu et al. (2020); Salinas et al. (2019)) with applications ranging

from anomaly detection to causal analysis or retail sales forecasting, where modelling of dependencies between articles is crucial to account for cannibalization effects, tree-based approaches have received comparatively few multivariate extensions. Recent advances include Pande et al. (2022) who introduce a gradient boosting approach to model multivariate longitudinal responses, as well as Nespoli and Medici (2022) who present a computationally efficient algorithm for fitting multivariate boosted trees. To address the problem of low generalization ability and tree redundancy when dependencies between several targets are ignored, Zhang and Jung (2021) propose a general method to train GBMs with multiple targets. O'Malley et al. (2021) extend the NGBoost model of Duan et al. (2020) to a multivariate Gaussian setting and Cevid et al. (2021) introduce a distributional regression forest for multivariate responses. Based on Bayesian additive regression trees (BARTs) of Chipman et al. (2010), Clark et al. (2021) develop multivariate time series models and Um et al. (2020) extend BARTs to allow modelling of multivariate skewed responses. Lang et al. (2020) introduce multivariate distributional regression forests to probabilistically predict wind profiles. Quan and Valdez (2018) use multivariate trees to model insurance claims data with correlated responses and Miller et al. (2016) introduce a multivariate extension of gradient boosted regression trees for continuous multivariate responses.

With this paper, we contribute to the emerging literature on multi-target probabilistic GBMs and present a multivariate extension of the univariate XGBoostLSS introduced by März (2019). Our approach leverages automatic differentiation using PyTorch (Paszke et al., 2019), which facilitates implementation of distributions for which Gradients and Hessians are difficult to derive analytically.

The remainder of this paper is organized as follows: Section II introduces our multivariate XGBoostLSS framework and Section III presents both a simulation study and real world examples. Section IV concludes. 4

II. MULTI-TARGET XGBOOSTLSS

In its original formulation, distributional modelling relates all parameters of a univariate response distribution to covariates \mathbf{x} . In particular, it assumes the response to follow a distribution $\mathcal{D}(\boldsymbol{\theta}(\mathbf{x}))$ that depends on up

^ℵ₀Author for correspondence: alex.maerz@gmx.net

¹For details on the M5 competition, see Makridakis et al. (2021a,b). For a good overview of tree-based methods and their use in the M5 competition, see Januschowski et al. (2021).

²While XGBoost (Chen and Guestrin, 2016) and CatBoost (Prokhorenkova et al., 2018; Dorogush et al., 2017) allow modelling of several responses, with a separate model trained for each target, LightGBM (Ke et al., 2017) currently does not support multi-target regression. A workaround often suggested for extending models that do not natively support multi-target regression is to use scikit-learn's Multi-Output-Regressor. However, since a separate model is trained per target, this does not allow modelling of dependencies between multiple responses.

³In the following, we use the terms multi-target and multivariate regression interchangeably for denoting environments where \mathbf{y} is a $N \times D$ response matrix $\mathbf{y} = (y_{i1}, \dots, y_{iD})^T, i = 1, \dots, N$ with D denoting the target dimension.

⁴The code of the implementation will be made available on OstatMixedML/XGBoostLSS at the time of the final publication of the paper.

to four parameters, i.e., $y_i \stackrel{ind}{\sim} \mathcal{D}(\mu_{i\mathbf{x}}, \sigma_{i\mathbf{x}}^2, \nu_{i\mathbf{x}}, \tau_{i\mathbf{x}}), i = 1, \ldots, N$, where $\mu_{i\mathbf{x}}$ and $\sigma_{i\mathbf{x}}^2$ are often location and scale parameters, respectively, while $\nu_{i\mathbf{x}}$ and $\tau_{i\mathbf{x}}$ correspond to shape parameters such as skewness and kurtosis. More generally, univariate distributional modelling can be formulated as follows

$$y_i \stackrel{ind}{\sim} \mathfrak{D} \begin{pmatrix} h_1(\theta_{i1}(\mathbf{x}_i)) = \eta_{i1} \\ h_2(\theta_{i2}(\mathbf{x}_i)) = \eta_{i2} \\ \vdots \\ h_K(\theta_{iK}(\mathbf{x}_i)) = \eta_{iK} \end{pmatrix}$$

for $i=1,\ldots,N$, where $\mathcal{D}(\cdot)$ denotes a parametric distribution for the response $\mathbf{y}=(y_1,\ldots,y_N)^T$ that depends on K distributional parameters $\theta_k, k=1,\ldots,K$, and with $h_k(\cdot)$ denoting a known function relating distributional parameters to predictors $\boldsymbol{\eta}_k$. The predictor specification $\boldsymbol{\eta}_k = f_k(\mathbf{x}), k=1,\ldots,K$ is generic enough to use either GAM-type, Deep Learning or GBMs as in our case.

To allow for a more flexible framework that explicitly models the dependencies of a D-dimensional response $\mathbf{y} = (y_{i1}, \dots, y_{iD})^T, i = 1, \dots, N$, Klein et al. (2015) introduce a multivariate version of distributional regression. Similar to the univariate case, multivariate distributional regression relates all K parameters of a multivariate density $f_i(y_{i1}, \dots, y_{iD} | \theta_{i1}(\mathbf{x}), \dots, \theta_{iK}(\mathbf{x}))$ to a set of covariates \mathbf{x} .

A. Multivariate Gaussian Regression

A common choice for multivariate probabilistic regression is to assume a multivariate Gaussian distribution, with the density given as 7

$$f\!\left(\mathbf{y}|\boldsymbol{\theta}_{\mathbf{x}}\right) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_{\mathbf{x}}|}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_{\mathbf{x}})^T\boldsymbol{\Sigma}_{\mathbf{x}}^{-1}(\mathbf{y}-\boldsymbol{\mu}_{\mathbf{x}})\right)$$

where $\boldsymbol{\mu}_{\mathbf{x}} \in \mathbb{R}^D$ represents a vector of conditional means, $\boldsymbol{\Sigma}_{\mathbf{x}}$ is a positive definite symmetric $D \times D$ covariance matrix and $|\cdot|$ denotes the determinant. For the bivariate case D = 2, $\boldsymbol{\Sigma}_{\mathbf{x}}$ can be expressed as

 $^6\mathrm{To}$ improve on the convergence and stability of XGBoostLSS estimation, unconditional Maximum Likelihood estimates of the parameters $\theta_k,\ k=1,\ldots,K$ are used as offset values. Also, since XGBoostLSS updates the parameters by optimizing Gradients and Hessians, it is important that these are comparable in magnitude for all distributional parameters. Due to variability regarding the ranges, the estimation of Gradients and Hessians might become unstable so that XGBoostLSS might not converge or might converge very slowly. To mitigate these effects, we have implemented a stabilization of Gradients and Hessians.

⁷In the further course of this section, we follow the notation of Muschinski et al. (2022); O'Malley et al. (2021); Salinas et al. (2019).

$$\boldsymbol{\Sigma}_{i\mathbf{x}} = \begin{bmatrix} \sigma_{i,1}^2(\mathbf{x}) & \rho_i(\mathbf{x})\sigma_{i,1}(\mathbf{x})\sigma_{i,2}(\mathbf{x}) \\ \rho_i(\mathbf{x})\sigma_{i,2}(\mathbf{x})\sigma_{i,1}(\mathbf{x}) & \sigma_{i,2}^2(\mathbf{x}) \end{bmatrix}$$

with the variances on the diagonal and the covariances on the off-diagonal, for i = 1, ..., N.

Cholesky Decomposition of Covariance Matrix

To ensure positive definiteness of Σ , the D(D+1)/2 entries of the covariance matrix must satisfy specific conditions.⁸ For the bivariate case, this can be ensured by applying exponential functions to the variances and a suitable transformation to restrict the coefficient of correlation $\rho \in [-1,1]$. However, in high-dimensional settings, where all moments are modelled as functions of covariates, ensuring positive definiteness of the covariance matrix becomes challenging, since joint restrictions for the elements of Σ are necessary (Muschinski et al., 2022). A computationally more tractable approach to ensure positive definiteness is based on the Cholesky-decomposition, that uniquely decomposes the covariance matrix as follows

$$\Sigma = \mathbf{L}\mathbf{L}^T$$

where $\mathbf{L} \in \mathbb{R}^{D \times D}$ is a lower triangular matrix. To ensure Σ to be positive definite, the D diagonal elements ℓ_{ii} of \mathbf{L} need to be strictly positive, whereas all D(D-1)/2 off diagonal elements ℓ_{ij} can take on any value in \mathbb{R} , leaving them untransformed. Illustrative for the bivariate case, the Cholesky factor \mathbf{L} is given as follows \mathbf{L}

$$\mathbf{L} = \begin{bmatrix} \exp(\ell_{11}) & 0 \\ \ell_{21} & \exp(\ell_{22}) \end{bmatrix}$$

In addition to reparameterizing the covariance matrix, the Cholesky decomposition is also computationally efficient, since only the determinant of a triangular matrix needs to be calculated (Salinas et al., 2019).

Low-Rank Covariance Approximation

While efficient for low to medium dimensions of D, the computational cost of the Cholesky-decomposition becomes prohibitive in high-dimensional settings. To reduce the computational overhead, the covariance matrix Σ can be approximated via the sum of a diagonal matrix $\mathbf{K} \in \mathbb{R}^{D \times D}_+$ and a unrestricted low-rank matrix $\mathbf{V} \in \mathbb{R}^{D \times r}$

$$\begin{split} \mathbf{\Sigma} &= \mathbf{K} + \mathbf{V}\mathbf{V}^T \\ &= \begin{bmatrix} \exp(\mathbf{K}_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \exp(\mathbf{K}_D) \end{bmatrix} + \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_D \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_D \end{bmatrix}^T \end{split}$$

⁵While the original formulation of GAMLSS in Rigby and Stasinopoulos (2005) suggests that any distribution can be described by location, scale and shape parameters, it is not necessarily true that the observed data distribution can actually be characterized by all of these parameters. Hence, we prefer to use the term distributional modelling.

⁸Without loss of generality, we notationally omit the explicit dependency of all parameters on \mathbf{x} and $i=1,\ldots,N$ in the following. ⁹For the precision matrix, the Cholesky-decomposition as given as $\mathbf{\Sigma}^{-1} = (\mathbf{L}^{-1})^T \mathbf{L}^{-1}$.

¹⁰In contrast to the original formulation of Σ , the elements in L do not have any direct interpretation.

where $\exp(\cdot)$ ensures all diagonal entries of \mathbf{K} to be strictly positive and the rank parameter r governs the quality of the approximation. The computational efficiency of this approach results from the fact that the rank parameter r << D can typically be chosen much smaller than the number of target variables D (Salinas et al., 2019). Showing the relationship between the response dimension D and the number of parameters K, Table 1 indicates that the number of parameters increases exponentially for the Cholesky-decomposition, while the relationship is only linear for the low-rank approximation, making it more suitable for high-dimensional settings.

[Table 1 about here.]

B. Multivariate Student-T Regression

As a generalization of the multivariate Gaussian, the multivariate Student-T is suitable when modelling heavy-tailed data, i.e., when there is more mass in the tails of the distribution. The density is given as

$$\begin{split} f \big(\mathbf{y} | \boldsymbol{\theta}_{\mathbf{x}} \big) &= \\ \frac{\Gamma \big[\frac{\boldsymbol{\nu}_{\mathbf{x}} + D}{2} \big]}{\Gamma \big[\frac{\boldsymbol{\nu}_{\mathbf{x}}}{2} \big] (\pi \boldsymbol{\nu}_{\mathbf{x}})^{D/2} |\boldsymbol{\Sigma}_{\mathbf{x}}|^{1/2}} \left[1 + \frac{(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{x}})^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} (\mathbf{y} - \boldsymbol{\mu}_{\mathbf{x}})}{\boldsymbol{\nu}_{\mathbf{x}}} \right]^{-\frac{\boldsymbol{\nu}_{\mathbf{x}} + D}{2}} \end{split}$$

with covariance matrix $\boldsymbol{\nu}_{\mathbf{x}}(\boldsymbol{\nu}_{\mathbf{x}}-2)^{-1}\boldsymbol{\Sigma}_{\mathbf{x}}$ and $\Gamma[\cdot]$ denoting the gamma function. $\boldsymbol{\mu}_{\mathbf{x}} \in \mathbb{R}^D$ and $\boldsymbol{\Sigma}_{\mathbf{x}}$ are defined as for the multivariate Gaussian. The multivariate Student-T distribution has an additional degrees of freedom parameter $\boldsymbol{\nu}_{\mathbf{x}} > 2$ that governs the tail behaviour, where for $\boldsymbol{\nu}_{\mathbf{x}} \to \infty$ the Student-T converges in distribution to the multivariate Normal. Similar to the multivariate Gaussian, we use the Cholesky-decomposition $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbf{L}_{\mathbf{x}} \mathbf{L}_{\mathbf{x}}^T$ to ensure the covariance matrix to be positive definite. 11

C. Dirichlet Regression

While the multivariate Gaussian and Student-T are defined for $\mathbf{y} \in \mathbb{R}^D$, the Dirichlet distribution is commonly used for modelling non-negative compositional data, i.e., data that consist of sub-sets that are fractions of some total. Compositional data are typically represented as proportions or percentages summing to 100%, so that the Dirichlet extends the univariate betadistribution to the multivariate case (Klein et al., 2015). Dating back to the seminal paper of Aitchison (1982), compositional data analysis (CoDa) is a branch of statistics that deals with multivariate observations carrying relative information and finds widespread use in ecology (Douma and Weedon, 2019), economics Fry et al. (2000) or political science (Katz and King, 1999). As a result of the unit-sum constraint, models that use distributions designed for unconstrained data typically suffer from

the problem of spurious correlation when applied to compositional data (Aitchison, 2003).

The density of the Dirichlet distribution with parameters $\boldsymbol{\alpha}_{\mathbf{x}} = (\alpha_{\mathbf{x},1},\dots,\alpha_{\mathbf{x},D}) \in \mathbb{R}_+^D$ with $\sum_{d=1}^D y_d = 1$ for all $y_d \in [0,1]$ is given by

$$f(\mathbf{y}|\boldsymbol{\theta}_{\mathbf{x}}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha}_{\mathbf{x}})} \prod_{d=1}^{D} y_d^{\alpha_{\mathbf{x},d-1}}$$

where the normalizing constant is expressed as the multinomial beta-function

$$B(\boldsymbol{\alpha}_{\mathbf{x}}) = \frac{\prod_{d=1}^{D} \Gamma(\alpha_{\mathbf{x},d})}{\Gamma\left(\sum_{d=1}^{D} \alpha_{\mathbf{x},d}\right)}$$

To ensure positivity, we use $\exp(\alpha_{\mathbf{x},d})$ for all $d=1,\ldots,D$. The estimated parameters have the interpretation of providing the probability of an event falling into category d, i.e., $\mathbb{E}(y_d) = \frac{\alpha_d}{\alpha_0}$, with $\alpha_0 = \sum_{d=1}^D \alpha_d$ (Klein et al., 2015).

III. APPLICATIONS

In this section, we present simulation studies and real-world examples to illustrate the functionality of our approach. All hyper-parameters of the models presented in this paper are selected using Optuna of Akiba et al. (2019). For all models, Table 2 shows the space of the tuneable hyper-parameter search.

[Table 2 about here.]

A. Simulation

Multivariate Gaussian Regression

We start with a trivariate Gaussian scenario (N=10,000), where all moments of the distribution are allowed to vary with covariates \mathbf{x} .¹²

Figure 1 shows that the estimated parameters of the multivariate Gaussian, with the covariance matrix being parametrized using either the Cholesky-decomposition (Panel 1a) or the low-rank approximation (Panel 1b), closely match the true parameters, even for a rank as low as 2. Yet, for ρ_{21} and ρ_{32} specifically, the low-rank approximation shows some deviations and generally a somewhat more erratic behaviour of the estimates as compared to the Cholesky-decomposition.

 $^{^{11}{\}rm Unlike}$ the multivariate Normal, a low-rank approximation of the covariance matrix is not yet implemented for the multivariate Student-T, but is planned in future releases.

 $^{^{12}}$ Hyper-parameters for each of the models in the simulation study are optimized running 100 hyper-parameter trails each.

Multivariate Student-T Regression

Figure 2 presents the simulation results for the trivariate Student-T distribution, where the covariance matrix is parametrized via the estimated Cholesky-factors $\hat{\mathbf{L}}\hat{\mathbf{L}}^T.^{13}$

[Figure 2 about here.]

The estimates presented in Figure 2 are still quite close to the true shapes, but not as accurate as for the multivariate Gaussian. In general, the Student-T estimates exhibit a somewhat more erratic behaviour than the Gaussian Cholesky-decomposition. This is especially true for the degrees of freedom parameter ν , which does not approximate the U-shape well. Also, parameter estimates of μ_3 and ρ_{21} deviate slightly from the true values.

Dirichlet Regression

To evaluate the ability of our approach of inferring the relationship between covariates and a set of Dirichlet-distributed responses, we apply our model to the widely used Arctic-Lake dataset of (Aitchison, 2003) that contains information on sediment composition (sand, silt, clay) of an Arctic lake. The data are shown in Figure 3.

[Figure 3 about here.]

The dataset includes 39 measurements and we model the sediment composition as a function of water depth in meters. Figure 4 compares the results of our model to a scatter-smooth estimate. To facilitate visual comparison, we use smoothed estimates of our model.

[Figure 4 about here.]

The scatter-smooth and our model estimates are in close agreement, showing that with increasing depth, the relative frequency of sand decreases while the proportion of silt and clay increases.

In summary, despite some deviations, our multivariate XGBoostLSS models approximate well the conditional moments of the underlying data generating processes for all distributions studied. For the simulated datasets, one could further improve the accuracy by increasing the trails of the hyper-parameter search or the number of boosting iterations, which we set to 100.

B. Regression Datasets

We benchmark our model, which we refer to as mXG-BoostLSS, against the multivariate NGBoost (mNG-Boost) model of O'Malley et al. (2021) and our univariate XGBoostLSS implementation (März, 2019) using a subset of the datasets of Spyromitros-Xioufis et al. (2016), as well as the California housing dataset of Pace and Barry (1997):¹⁴

- Airline Ticket Price (1d): This dataset is concerned with the prediction of airline ticket prices and the target variables are the next day price for 6 flight preferences: (1) any airline with any number of stops, (2) any airline non-stop only, (3) Delta Airlines, (4) Continental Airlines, (5) Airtrain Airlines, and (6) United Airlines.
- California Housing: This dataset was derived from the 1990 U.S. census and contains information with respect to demography, location, as well as general information regarding the house in Californian districts. As responses, we use the median income and the median house value.
- Jura: The dataset consists of measurements of concentrations of seven heavy metals recorded at 359 locations in the topsoil of a region of the Swiss Jura. The type of land use and rock type are also recorded for each location. Cadmium, copper and lead are treated as targets, while the remaining metals along with land use type, rock type and the coordinates of each location are used as features.
- Occupational Employment Survey (2010): The Occupational Employment Survey data were obtained from the annual Occupational Employment Survey compiled by the US Bureau of Labor Statistics in 2010. Each target denotes the estimated number of full-time equivalent employees across many employment types (e.g., doctor, dentist, car repair technician, etc.) across 403 cities.
- River Flow 1: The dataset, obtained from the US National Weather Service and collected between September 2011 and September 2012, contains data from hourly river flow observations for 8 sites in the Mississippi River network and the task is to predict river network flows at specific locations.
- Supply Chain Management (1d): This dataset is derived from the Trading Agent Competition in Supply Chain Management (TAC SCM) tournament from 2010 and contains 16 targets, each corresponding to the next day mean price for each product in the competition.
- Slump: This dataset is concerned with the prediction of three properties of concrete (slump, flow and compressive strength) as a function of the content of seven concrete ingredients.

We also include a subset of the simulated trivariate Gaussian and Student-T datasets presented in Section III-A, with additional noise features added. Table 3 presents the data and its characteristics.

[Table 3 about here.]

From Table 3 it follows that there is a non-negligible dependence between targets in all datasets. It is interesting to see how the univariate baseline, which assumes independence between targets, compares to the

 $^{^{13}\}mathrm{Similar}$ to the Gaussian scenario, we set N=10,000.

¹⁴We restrict the datasets in Spyromitros-Xioufis et al. (2016) to continuous or close-to continuous responses only. We use the California housing dataset of scikit-learn. All other datasets are publicly available at http://mulan.sourceforge.net/datasets-mtr.html. We run all experiments on a 8-core Intel(R) i7-7700T CPU with 32 GB of RAM.

multivariate models. For conducting the experiments, we create 11 randomly shuffled folds for all datasets. Each is split into train and test, where 80% is used for training and the remaining 20% for evaluation. The first fold is used for hyper-parameter tuning only, where we run 100 hyper-parameter trails for each model-dataset combination and select the best set of hyper-parameters. All models are initialized with 500 boosting rounds, with the optimal number of iterations based on early-stopping. Once optimized, we keep the optimal set of hyper-parameters constant and use the remaining 10 folds for model evaluation. All models except the Student-T assume a Gaussian distribution, with the accuracy being evaluated using the negative log-likelihood (NLL). Table 4 reports the results.

[Table 4 about here.]

Table 4 shows that our models compare well with existing implementations. With an average rank of 2.2, mXGBoostLSS-G-C has the highest overall accuracy, closely followed by mXGBoostLSS-T-C and mNGBoost-G-C with an average rank of 2.4 each. For kurtotic datasets, the mXGBoostLSS-T-C outperforms its Gaussian counterparts due to its additional degrees of freedom parameter. Probably due to its fixed and nonoptimized rank parameter r, the LRA model ranks last in our comparison. For the comparatively high-dimensional scm1d and oes10-datasets, the LRA model shows some diverging behaviour. We will further investigate the effect of the rank on the LRA model in Section III-C. The comparison between models also shows that explicitly modelling dependencies between targets tends to increase accuracy: for 7 out of 9 datasets, the multivariate models have a higher accuracy than the univariate model. This is consistent with the results of Schmid et al. (2023) who report that the performances of multivariate approaches were substantially better than the univariate ones for some of the simulation settings considered.

To further benchmark our models, Table 5 reports the variability $(NLL_{max} - NLL_{min})$ of the NLL-scores across folds and datasets, demonstrating that explicit modelling of dependencies tends to stabilize estimation.

[Table 5 about here.]

Compared to its multivariate counterparts, the univariate model shows the highest average variability across datasets and folds. As an example, consider the sl-dataset, for which the univariate XGBoostLSS model shows some divergent predictions. The lower overall variability might be attributed to the fact that joint modelling of all targets tends to stabilize the estimation in the multivariate setting. Within the class of multivariate

models, mXGBoostLSS-G-C shows the least variation, closely followed by mXGBoostLSS-T-C.

In addition to assessing the accuracy, Table 6 presents an overview of normalized runtimes. To ensure a fair comparison, we set all hyper-parameters of the models to the same values. All models are estimated using CPUs and all XGBoostLSS models, both univariate and multivariate, are trained without leveraging its fast histogram tree-growing method. We exclude the univariate XGBoostLSS model from the analysis since it has less parameters compared to its multivariate counterparts and therefore always lower runtimes.

[Table 6 about here.]

From Table 6 it follows that, while mNGBoost-G-C has the lowest runtime for the smallest sl-dataset, the Cholesky-based mXGBoostLSS models scale well with the number of observations and with the mXGBoostLSS-G-C model being the most efficient in terms of runtime across datasets. The efficiency is likely to increase further if we leverage XGBoostLSS's GPU-histogram training. Also, for fairly large datasets, one can use distributed training with Dask for even better scalability. Table 6 also shows that the LRA-model benefits from its linear increase in parameters which results in the lowest runtime for the high-dimensional oes10-dataset.

C. Ablation Study

Following the discussion in the previous section, it remains to be investigated why the LRA-model does not perform as well as the other models, especially for datasets with small D and N. One reason might be the low number of observations relative to the number of estimated parameters. Recall that for the Gaussian, the Cholesky factorization of the covariance matrix requires estimation of D(D+3)/2 parameters, while D(2+r) need to be estimated for the low-rank covariance approximation. As an example, take the relatively small sl-dataset with D=3: while only 9 parameters have to be estimated for the Cholesky model, there are already 21 parameters for the LRA-model with r=5.

Another reason might be related to the choice of the rank parameter r. While all hyper-parameters of the Cholesky models are optimized, we set r=5 for all datasets, mainly to keep the computational cost low while still maintaining a reasonable fit. To further investigate the effect of r on the accuracy, we run additional experiments using a small subset of the datasets with varying values of r. Table 7 reports the results.

[Table 7 about here.]

Table 7 shows that the quality of the covariance approximation varies with the rank parameter. However, there is no general recommendation that higher values of r tend to increase accuracy. This can be seen from the atp1d-dataset, where the model tends to overfit with increasing values of r, resulting in lower accuracy. For the relatively

¹⁵For the high-dimensional scm1d and oes10-datasets, we were not able to complete the hyper-parameter search for mNGBoost, mainly due to out-of-memory issues, non-positive definite covariance matrices, or unrealistically long runtimes for a single iteration. For these reasons, we had to re-initialize the hyper-parameter search several times and also reduce the number of hyper-parameter trails to 20.

low-dimensional ju-dataset, higher values of r increase accuracy, while for the sl-dataset, a moderately low rank gives the best results. Since the order of r is depending on the size of the dataset and the characteristics determining the covariance structure, our recommendation would be to add the rank parameter as a tuneable hyperparameter.

IV. CONCLUSION, LIMITATIONS AND FUTURE RESEARCH

While most implementations of Gradient Boosting Machines are tailored to single-target settings, this paper presents an extension of XGBoostLSS that models multi-targets and their dependencies in a probabilistic regression environment. Using simulation studies and real-world data, we have shown that our approach outperforms existing GBMs with respect to runtime and is competitive in terms of accuracy. We have also demonstrated that explicit modelling of dependencies between targets can lead to an increase in accuracy.

Despite its flexibility, we acknowledge some limitations of our approach that require additional research. Although the base XGBoost implementation accepts a $N \times D$ array of responses, model training is still optimized for single-target models. This implies that for our XGBoostLSS approach, which is based on multiparameter training, with a separate tree grown for each parameter, estimating many parameters for a large dataset can become computationally expensive, with the computational cost growing $\mathcal{O}(K^2)$. We would like to emphasize, though, that high-dimensional multi-target regression and multiclass-classification is a known scaling problem of current GBMs implementations and therefore not a unique limitation of our approach. An interesting scope for future implementation and research would be a more runtime efficient version of our framework, where multiple parameters can be estimated with a single tree. In addition, we consider the extension of our framework to allow for a more flexible choice of multivariate response distributions beyond the Gaussian, Dirichlet and Student-T to be an interesting refinement.

References

- Aitchison, J. (1982). The Statistical Analysis of Compositional Data. *Journal of the Royal Statistical Society:* Series B (Methodological), 44(2):139–160.
- Aitchison, J. (2003). The statistical analysis of compositional data. Blackburn Press, Caldwell, N.J.
- Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. In Teredesai, A., editor, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM Digital Library, pages 2623–2631, New York,NY,United States. Association for Computing Machinery.
- Ćevid, D., Michel, L., Näf, J., Meinshausen, N., and Bühlmann, P. (2021). Distributional Random Forests:

- Heterogeneity Adjustment and Multivariate Distributional Regression. arXiv Pre-Print.
- Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pages 785– 794, New York, NY, USA. Association for Computing Machinery.
- Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.
- Clark, T. E., Huber, F., Koop, G., Marcellino, M., and Pfarrhofer, M. (2021). Tail Forecasting with Multivariate Bayesian Additive Regression Trees. Federal Reserve Bank of Cleveland, Working Paper No. 21-08.
- Dorogush, A. V., Ershov, V., and Gulin, A. (2017). CatBoost: gradient boosting with categorical features support. Workshop on ML Systems at NIPS.
- Douma, J. C. and Weedon, J. T. (2019). Analysing continuous proportions in ecology and evolution: A practical introduction to beta and Dirichlet regression. *Methods in Ecology and Evolution*, 10(9):1412–1430.
- Duan, T., Anand, A., Ding, D. Y., Thai, K. K., Basu, S., Ng, A. Y., and Schuler, A. (2020). NGBoost: Natural Gradient Boosting for Probabilistic Prediction.
 In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research, pages 2690–2700. PMLR.
- Fry, J. M., Fry, T. R. L., and McLaren, K. R. (2000). Compositional data analysis and zeros in micro data. Applied Economics, 32(8):953–959.
- Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., and Gasthaus, J. (2021). Forecasting with trees. *International Journal of Forecasting*.
- Kan, K., Aubet, F.-X., Januschowski, T., Park, Y., Benidis, K., Ruthotto, L., and Gasthaus, J. (2022). Multivariate Quantile Function Forecaster. arXiv Pre-Print.
- Katz, J. N. and King, G. (1999). A Statistical Model for Multiparty Electoral Data. The American Political Science Review, 93(1):15–32.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, pages 3149–3157, Red Hook, NY, USA. Curran Associates Inc.
- Klein, N., Kneib, T., Klasen, S., and Lang, S. (2015). Bayesian structured additive distributional regression for multivariate responses. *Journal of the Royal Statis*tical Society: Series C (Applied Statistics), 64(4):569– 591.
- Lang, M. N., Mayr, G. J., Schlosser, L., Simon, T., Stauffer, R., and Zeileis, A. (2020). Multivariate Distributional Regression Forests for Probabilistic Nowcasting of Wind Profiles. In Irigoien, I., Lee, D.-J.,

- Martínez-Minaya, J., and Rodríguez-Álvarez, M. X., editors, *Proceedings of the 35th International Workshop on Statistical Modelling*, pages 142–147, Bilbao.
- Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2021a). The M5 competition: Background, organization, and implementation. *International Journal of Forecasting*.
- Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen,
 Z., Gaba, A., Tsetlin, I., and Winkler, R. L. (2021b).
 The M5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting*.
- März, A. (2019). XGBoostLSS An extension of XG-Boost to probabilistic forecasting. arXiv Pre-Print, pages 1–23.
- Miller, P. J., Lubke, G. H., McArtor, D. B., and Bergeman, C. S. (2016). Finding structure in data using multivariate tree boosting. *Psychological methods*, 21(4):583–602.
- Muschinski, T., Mayr, G. J., Simon, T., Umlauf, N., and Zeileis, A. (2022). Cholesky-based multivariate Gaussian regression. *Econometrics and Statistics*.
- Nespoli, L. and Medici, V. (2022). Multivariate Boosted Trees and Applications to Forecasting and Control. arXiv Pre-Print.
- O'Malley, M., Sykulski, A. M., Lumpkin, R., and Schuler, A. (2021). Multivariate Probabilistic Regression with Natural Gradient Boosting. arXiv Pre-Print, pages 1–19.
- Pace, K. R. and Barry, R. (1997). Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297
- Pande, A., Ishwaran, H., and Blackstone, E. (2022). Boosting for Multivariate Longitudinal Responses. *SN Computer Science*, 3(3):186.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc, Red Hook, NY, USA.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush,
 A. V., and Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In S. Bengio,
 H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc.
- Quan, Z. and Valdez, E. A. (2018). Predictive analytics of insurance claims using multivariate decision trees. Dependence Modeling, 6(1):377–407.
- Rasul, K., Seward, C., Schuster, I., and Vollgraf, R. (2021a). Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting.
 In Meila, M. and Zhang, T., editors, Proceedings of the 38th International Conference on Machine Learn-

- ing, volume 139 of Proceedings of Machine Learning Research, pages 8857–8868. PMLR.
- Rasul, K., Sheikh, A.-S., Schuster, I., Bergmann, U. M., and Vollgraf, R. (2021b). Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *International Conference on Learning Rep*resentations.
- Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized additive models for location, scale and shape. Journal of the Royal Statistical Society: Series C (Applied Statistics), 54(3):507–554.
- Salinas, D., Bohlke-Schneider, M., Callot, L., Medico, R., and Gasthaus, J. (2019). High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.
- Schmid, L., Gerharz, A., Groll, A., and Pauly, M. (2023). Tree-based ensembles for multi-output regression: Comparing multivariate approaches with separate univariate ones. *Computational Statistics & Data Analysis*, 179:107628.
- Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., and Vlahavas, I. (2016). Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98.
- Um, S., Linero, A., Sinha, D., and Bandyopadhyay, D. (2020). Bayesian Additive Regression Trees for Multivariate Skewed Responses. In American Statistical Association, editor, 2020 JSM Proceedings: Papers Presented at the Virtual Joint Statistical Meetings, August 2-6, 2020, and Other ASA-sponsored Conferences. American Stastistical Association.
- Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. (2020). Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In Gupta, R., editor, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM Digital Library, pages 753–763, New York,NY,United States. Association for Computing Machinery.
- Zhang, Z. and Jung, C. (2021). GBDT-MO: Gradient-Boosted Decision Trees for Multiple Outputs. IEEE Transactions on Neural Networks and Learning Systems, 32(7):3156–3167.

TABLES 8

TABLE 1: Number of parameters for Cholesky and Low-Rank Approximation (LRA)

Y_D	Cholesky	LRA(r=5)	LRA(r=10)	LRA(r=20)
2	5	14	24	44
5	20	35	60	110
10	65	70	120	220
50	1,325	350	600	1,100
100	$5,\!150$	700	1,200	2,200
500	125,750	3,500	6,000	11,000
1,000	$501,\!500$	7,000	12,000	22,000
10,000	50,015,000	70,000	120,000	220,000

The table shows the number of parameters K to estimate for a Multivariate Gaussian for the Cholesky D(D+3)/2 and the low-rank covariance matrix approximation D(2+r) as functions of the response dimension \mathbf{Y}_D .

TABLE 2: Hyper-Parameter Search-Space

	Range
learning-rate	[0.001, 1.0]
max-depth	[2, 10]
gamma	[0, 100]
sub-sample	[0.4, 1.0]
col-sample	[0.4, 1.0]
min-child-weight	[0, 500]
boosting-iterations	[500]
early-stopping-rounds	[2]

TABLE 3: Dataset Overview

Dataset	Abbreviation	Observations	Y_D	Features	Dependency-Strength
Airline Ticket Price (1d)	atp1d	337	6	411	0.8013 [0.7305, 0.9166]
California Housing	ch	20,640	2	7	0.6881 [0.6881, 0.6881]
Jura	ju	359	3	15	0.1907 [0.1567, 0.2452]
Occupational Employment Survey (2010)	oes10	403	16	298	0.549 [0.3195, 0.6928]
River Flow 1	rf1	9,125	7	65	0.5028 [0.0799, 0.8208]
Supply Chain Management (1d)	scm1d	9,803	16	280	0.6256 [0.4857, 0.8386]
Simulated Trivariate Gaussian	stg	2,000	3	5	0.4585 [0.4542, 0.5035]
Simulated Trivariate Student-T	stt	3,000	3	6	0.5524 [0.5261, 0.6780]
Slump	sl	103	3	7	-0.124 [-0.2035, 0.7001]

Due to its high level of skewness that caused instabilities for all models, we removed one target (target_NASI2_48H_0) from the rf1 dataset. Further, also for stability reasons, we applied a Box-Cox transformation to all responses of the oes10 dataset. The last column measures the median strength of dependency between responses using the Pearson coefficient of correlation, with additional quantiles in parentheses, i.e., $q_{0.5}(q_{0.1}, q_{0.9})$.

TABLES 9

TABLE 4: NLL scores

	mNGBoost-G-C	mXGBoostLSS-G-C	mXGBoostLSS-G-LRA(5)	mXGBoostLSS-T-C	${\bf uXGBoostLSS\text{-}G}$
atp1d	32.1131 [31.1064, 33.3593]	33.1507 [32.4934, 33.8251]	35.2945 [34.7998, 37.9187]	34.2416 [32.8878, 35.8722]	36.4295 [35.6846, 37.9606]
ch	1.5875 [1.5602, 1.656]	1.6921 [1.6512, 1.7364]	2.2747 [2.2289, 2.3244]	1.6679 [1.6375, 1.7526]	1.5608 [1.5369, 1.6244]
ju	6.9913 [6.054, 7.6863]	6.9372 [6.6957, 7.489]	7.0017 [6.5767, 7.1895]	7.4975 [7.1694, 7.8141]	7.5602 [7.1841, 8.068]
oes10	4.9139 [3.8811, 5.8206]	5.763 [5.3078, 6.6205]	9.1977 [8.3772, 10.0111]	4.0237 [3.5981, 5.3306]	4.4383 [3.798, 6.6723]
rf1	26.5513 [26.3026, 26.8497]	21.3728 [21.2832, 21.4263]	24.9533 [24.337, 25.1423]	23.3522 [23.1668, 24.2478]	19.1992 [19.0538, 20.0179]
scm1d	97.5695 [97.3239, 97.9193]	95.5774 [95.4184, 95.8474]	128.814 [128.1463, 129.0811]	95.124 [95.014, 95.3738]	107.1911 [107.0608, 107.3768]
stg	3.706 [3.5851, 3.8114]	3.664 [3.5365, 3.7215]	3.9296 [3.7848, 4.0291]	3.8374 [3.7553, 3.898]	4.6199 [4.5147, 4.7021]
stt	5.1247 [4.7432, 5.6858]	4.8221 [4.7306, 5.7158]	5.1441 [4.8861, 5.4495]	4.4243 [4.3407, 4.4598]	5.7544 [5.5818, 6.8563]
sl	$10.0841 \ [9.4764, 10.9282]$	10.4098 [10.1858, 11.065]	10.7407 [10.5641, 11.2718]	10.6563 [10.3614, 11.573]	12.0785 [11.4485, 13.5503]
Average Rank	2.4	2.2	4.2	2.4	3.7

The table shows median NLL scores across models, datasets and folds, with additional quantiles in parentheses, i.e., $q_{0.5}(q_{0.1}, q_{0.9})$. Lower is better, with best results in bold. At the bottom, we also report average ranks across datasets. Again, lower is better. The columns are to be read as follows: Model-Distribution-Covariance Approximation, where G: Gaussian, T: Student-T, C: Cholesky, LRA: Low-Rank Approximation(r), m: multivariate, u: univariate.

TABLE 5: NLL Variability

	mNGBoost-G-C	mXGBoostLSS-G-C	mXGBoostLSS-G-LRA(5)	mXGBoostLSS-T-C	uXGBoostLSS-G
atp1d	2.6763	1.8471	5.4083	3.8626	3.3171
$^{\mathrm{ch}}$	0.1212	0.1120	0.1290	0.1227	0.1106
ju	2.2576	1.2288	1.0296	0.6547	1.0135
oes10	2.6402	1.7893	2.6149	2.3389	3.8587
rf1	1.3828	0.3413	1.1937	2.4426	1.1768
scm1d	0.8980	0.6472	1.4308	0.6590	0.5307
stg	0.3263	0.2021	0.3171	0.1691	0.3044
stt	1.7744	1.0664	1.0759	0.2368	1.5770
$_{ m sl}$	1.6586	1.2184	0.9302	1.3211	12.1062
Average	1.5262	0.9392	1.5699	1.3119	2.6661

The table shows the distance between maximum and minimum NLL scores $(NLL_{max} - NLL_{min})$ across models, dataset and folds. Lower is better, with best results in bold. At the bottom, we also report average distances across datasets. Again, lower is better. The columns are to be read as follows: Model-Distribution-Covariance Approximation, where G: Gaussian, T: Student-T, C: Cholesky, LRA: Low-Rank Approximation(r), m: multivariate, u: univariate.

TABLE 6: Relative Median Runtimes

	mNGBoost-G-C	${\rm mXGBoostLSS\text{-}G\text{-}C}$	mXGBoostLSS-G-LRA(5)	mXGBoostLSS-T-C
atp1d	5.4236	1.0000	2.1106	1.0881
ch	6.5707	1.0000	10.8340	1.4833
ju	1.2184	1.0733	3.8502	1.0000
oes10	5.5625	1.1406	1.0000	1.4297
rf1	10.0112	1.0413	2.6890	1.0000
scm1d	45.1237	1.1741	2.0985	1.0000
stg	3.1539	1.0352	7.4486	1.0000
stt	3.5399	1.0000	9.0137	1.3316
sl	1.0000	2.0964	8.9662	3.0045
Average Rank	3.2	1.7	3.3	1.8

The table shows relative median runtimes, with entries normalized to the model with the lowest runtime. Lower is better, with best results in bold. The following hyper-parameters are used: learning-rate=0.1, max-depth=6, iterations=100. All other hyper-parameters are set to their default values. The columns are to be read as follows: Model-Distribution-Covariance Approximation, where G: Gaussian, T: Student-T, C: Cholesky, LRA: Low-Rank Approximation(r), m: multivariate, u: univariate.

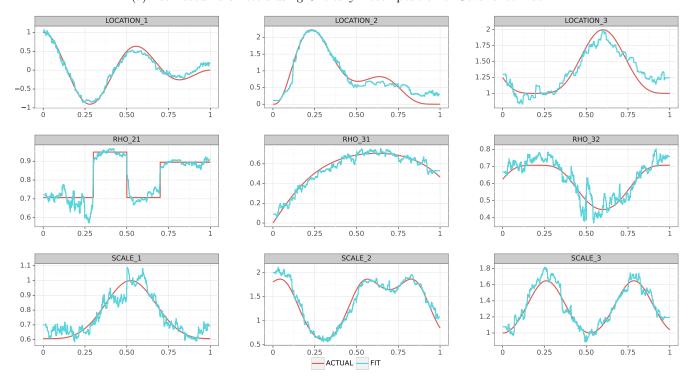
TABLE 7: Ablation Results of the LRA(r) model

Rank (r)	atp1d	ju	sl
2	34.9707 [34.3182, 35.9444]	7.2583 [6.9075, 7.6592]	14.9027 [14.6824, 16.4902]
4	35.0910 [34.3140, 36.8900]	7.5869 [6.9829, 7.9232]	12.8737 [12.2232, 12.9751]
5	35.2945 [34.7998, 37.9187]	7.0017 [6.5767, 7.1895]	10.7407 [10.5641, 11.2718]
6	48.0852 [47.1251, 48.4539]	8.1454 [7.8824, 8.3371]	16.8487 [14.7109, 23.4578]
8	37.8567 [37.1523, 38.7453]	$6.9445 \; [6.2847, 7.4682]$	13.4723 [13.2427, 13.6850]
10	70.3046 [48.6155, 79.8918]	7.2275 [6.8551, 7.5264]	14.7997 [14.4707 , 18.2046]

The table shows median NLL scores across datasets, folds and varying values of r for the LRA model, with additional quantiles in parentheses, i.e., $q_{0.5}(q_{0.1}, q_{0.9})$. Lower is better, with best results in bold.

Fig. 1: Estimated Parameters of Trivariate Gaussian.

(a) Estimated Parameters using Cholesky-Decomposition of Covariance-Matrix.



(b) Estimated Parameters using a Low-Rank-Approximation (r=2) of Covariance-Matrix.

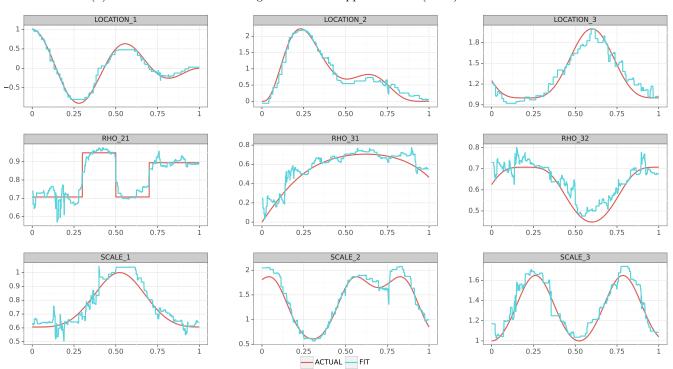


Fig. 2: Estimated Parameters of trivariate Student-T.

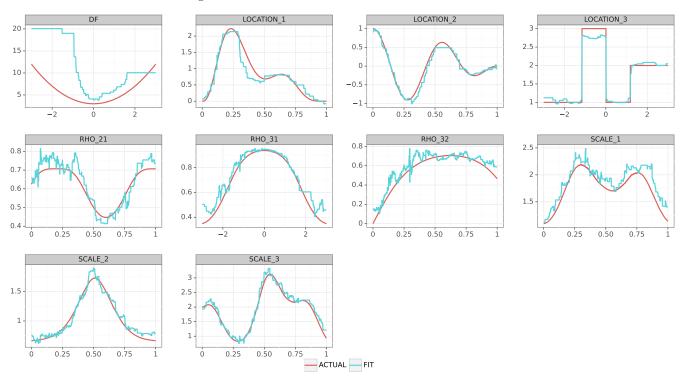


Fig. 3: Relative Frequencies of Sand, Silt, and Clay in Arctic-Lake Data.

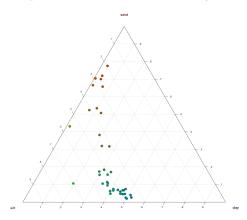


Fig. 4: Sediment Composition of Arctic-Lake Data.

