The Journal of Open Source Software

quantile-forest: A Python Package for Quantile Regression Forests

Reid A. Johnson [1]

1 Zillow Group, USA

DOI: 10.21105/joss.05976

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @jncraton
- @oparisot

Submitted: 02 August 2023
Published: 19 January 2024

Summary

Quantile regression forests (QRF) is a non-parametric, tree-based ensemble method for estimating conditional quantiles (Meinshausen, 2006). It is a generalization of the random forests algorithm, a versatile ensemble learning algorithm, originally proposed in (Breiman, 2001), that has proven extremely popular and useful as a general-purpose machine learning method (Athey et al., 2019; Biau & Scornet, 2016; Hengl et al., 2018; Wager & Athey, 2018). Instead of outputting the weighted mean value of training labels like random forests regressors, QRF employs the weighted empirical distribution of training labels to obtain the predictive distribution. This feature enables QRF to output probabilistic predictions for regression problems, which are widely useful for constructing estimates of uncertainty (Petropoulos et al., 2022).

`quantile-forest` provides a fast, feature-rich QRF implementation. The estimators provided in this package are optimized using Cython (Behnel et al., 2010) for training and inference speed, and can estimate arbitrary quantiles at prediction time without retraining. They provide methods for out-of-bag estimation, calculating quantile ranks, and computing proximity counts. The provided QRF estimators are also compatible with and can serve as drop-in replacements for the widely used forest regressors available in scikit-learn (Kramer, 2016). The package is designed to be used by a broad array of researchers and in production business settings. It has already been cited in scholarly work (Althoff, 2023; Prinzhorn, 2023; Saporta, 2023) and used in a production setting at the real estate technology company Zillow. The combination of speed, design, and functionality in `quantile-forest` enables exciting scientific explorations in academia and industry alike.

Statement of Need

Quantile regression is useful for understanding relationships between variables outside of the mean of the data, which can be particularly useful for understanding outcomes that are non-normally distributed or that have nonlinear relationships with predictor variables. It can be used to understand an outcome at its various quantiles and to compare groups or levels of an exposure on those quantiles (Koenker, 2005).

QRF, an extension of the random forests algorithm, provides a flexible, nonlinear and nonparametric way of performing quantile regression on the predictive distributions for high-dimensional data. Unlike traditional machine learning algorithms that focus solely on point estimation, QRF enables researchers to obtain a more comprehensive understanding of the underlying data distribution. It provides predictions not only for the expected outcome but also for various quantiles, allowing researchers to quantify uncertainties and capture the full spectrum of potential outcomes. QRF has become a standard method for probabilistic prediction in machine learning and has been applied to many areas requiring reliable probabilistic predictions.

Traditional prediction intervals often rely on assumptions such as normality, which may not hold in many real-world scenarios (Gyamerah & Moyo, 2020). QRF, on the other hand, allows researchers to generate prediction intervals that are non-parametric, flexible, and adaptive to different data distributions. This capability is invaluable for quantifying uncertainties in a wide range of research areas, including finance (Córdoba et al., 2021), environmental sciences (Fang et al., 2018; Francke et al., 2008; Zhang et al., 2018), healthcare (Dean et al., 2022; Molinder et al., 2020), and more. A crucial difference between QRF and many other quantile regression approaches is that after training a QRF once, one has access to all the quantiles at inference time, whereas most approaches require retraining separately for each quantile.

As a cutting-edge statistical modeling technique, QRF holds enormous potential for researchers across many domains, providing them with a powerful tool to address complex problems involving quantile regression and uncertainty estimation. The QRF algorithm is broadly available in R, which is host to the canonical QRF implementation (Meinshausen, 2017) as well as established alternative implementations (Athey et al., 2019; Wright & Ziegler, 2017). However Python has emerged as a prevailing standard programming language within the scientific community, making it a popular option for researchers and practitioners. The absence of a comprehensive Python QRF implementation severely hampers researchers' ability to utilize and benefit from its wide-ranging applications.

We seek to fill this need by providing a comprehensive Python-based QRF implementation. While other Python-based implementations of the QRF algorithm exist and are meaningful additions to the Python ecosystem (Kumar & others, 2017; Roebroek & others, 2022), none currently provide performance and functionality comparable to the implementations available in R, such as specifying quantiles at inference time or scaling to large datasets without approximation. By contrast, the QRF implementation provided in this package has been optimized for training and inference speed, enabling it to scale to millions of samples with a runtime that is orders of magnitude faster than less-optimized solutions. It also allows specifying prediction quantiles after training, permitting a trained model to be reused to estimate conditional quantiles as needed. Beyond this, the package includes utilities that enhance the algorithm's applicability and usefulness for researchers and practitioners. These utilities include:

- Out-of-bag scoring: The QRF algorithm can utilize the out-of-bag (OOB) samples, which are the data points that are not used during the construction of a specific decision tree in the forest. OOB scoring can be used to obtain unbiased estimates of prediction errors and quantile-specific metrics, enabling researchers to assess the performance and reliability of the QRF model without the need for additional validation datasets.
- Quantile rank calculation: The QRF algorithm can be leveraged to facilitate the calculation of quantile ranks. Quantile ranks provide a measure of relative standing for each data point in the distribution. This capability allows researchers to compare and rank observations based on their position within the quantile distribution, providing valuable insights for various applications, such as risk assessment and anomaly detection.
- Proximity and similarity estimation: The QRF algorithm can compute proximity measures that quantify the similarity between pairs of observations based on their paths through the forest. These proximity measures capture the notion of closeness or similarity between data points and enable researchers to perform tasks such as clustering, anomaly detection, and identifying influential observations.

By incorporating these utilities into a Python-based QRF implementation, researchers gain a comprehensive and versatile toolkit for quantile regression and uncertainty estimation. Researchers can now harness the power of Python's ecosystem to seamlessly integrate QRF into their existing workflows, perform thorough model evaluation through OOB scoring, assess quantile ranks, and leverage proximity and similarity measures for a wide range of data analysis tasks. Altogether, this package enables researchers to estimate conditional quantiles accurately, thereby empowering them to gain deeper insights into complex data.

Acknowledgements

We thank Andy Krause and Andrew Martin for support throughout the development and release of this work, and Ravi Ashar-Bryant for help during its early development. We also thank the many other members of Zillow Group who have shown support for this project.

Examples

Training and Predicting

```python
from quantile_forest import RandomForestQuantileRegressor
from sklearn import datasets
from sklearn.model_selection import train_test_split

X, y = datasets.fetch_california_housing(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

qrf = RandomForestQuantileRegressor().fit(X_train, y_train)

y_pred = qrf.predict(X_test, quantiles=[0.025, 0.5, 0.975])
y_pred_oob = qrf.predict(X_train, quantiles=[0.025, 0.5, 0.975], oob_score=True)
```

Estimating Quantile Ranks

```python
from quantile_forest import RandomForestQuantileRegressor
from sklearn import datasets
from sklearn.model_selection import train_test_split

X, y = datasets.fetch_california_housing(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

qrf = RandomForestQuantileRegressor().fit(X_train, y_train)
y_ranks = qrf.quantile_ranks(X_test, y_test)
```

Computing Proximities

```python
from quantile_forest import RandomForestQuantileRegressor
from sklearn import datasets
from sklearn.model_selection import train_test_split

X, y = datasets.fetch_california_housing(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

qrf = RandomForestQuantileRegressor().fit(X_train, y_train)
proximities = qrf.proximity_counts(X_test)
```

References

Althoff, S. (2023). *Conform with the wind* [Master's Thesis]. Lund University.

Athey, S., Tibshirani, J., & Wager, S. (2019). *Generalized random forests*. https://doi.org/10.1214/18-aos1709

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2010). Cython: The best of both worlds. *Computing in Science & Engineering*, *13*(2), 31–39.

https://doi.org/10.1109/mcse.2010.118

Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, *25*, 197–227. https://doi.org/10.1007/s11749-016-0481-7

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Córdoba, M., Carranza, J. P., Piumetto, M., Monzani, F., & Balzarini, M. (2021). A spatially based quantile regression forest model for mapping rural land values. *Journal of Environmental Management*, *289*, 112509. https://doi.org/10.1016/j.jenvman.2021.112509

Dean, A., Meisami, A., Lam, H., Van Oyen, M. P., Stromblad, C., & Kastango, N. (2022). Quantile regression forests for individualized surgery scheduling. *Health Care Management Science*, *25*(4), 682–709. https://doi.org/10.1007/s10729-022-09609-0

Fang, Y., Xu, P., Yang, J., & Qin, Y. (2018). A quantile regression forest based method to predict drug response and assess prediction reliability. *PLoS One*, *13*(10), e0205155. https://doi.org/10.1371/journal.pone.0205155

Francke, T., López-Tarazón, J., & Schröder, B. (2008). Estimation of suspended sediment concentration and yield using linear models, random forests and quantile regression forests. *Hydrological Processes*, *22*(25), 4892–4904. https://doi.org/10.1002/hyp.7110

Gyamerah, S. A., & Moyo, E. (2020). Long-term exchange rate probability density forecasting using Gaussian kernel and quantile random forest. *Complexity*, *2020*, 1–11. https://doi.org/10.1155/2020/1972962

Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B., & Gräler, B. (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ*, *6*, e5518. https://doi.org/10.7717/peerj.5518

Koenker, R. (2005). *Quantile regression*. Cambridge University Press. https://doi.org/10.1017/CBO9780511754098

Kramer, O. (2016). scikit-learn. *Machine Learning for Evolution Strategies*, 45–53. https://doi.org/10.1007/978-3-319-33383-0_5

Kumar, M., & others. (2017). *Scikit-garden*. https://github.com/scikit-garden/scikit-garden

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(35), 983–999. http://jmlr.org/papers/v7/meinshausen06a.html

Meinshausen, N. (2017). *quantregForest: Quantile regression forests*. https://cran.r-project.org/web/packages/quantregForest/index.html

Molinder, J., Scher, S., Nilsson, E., Körnich, H., Bergström, H., & Sjöblom, A. (2020). Probabilistic forecasting of wind turbine icing related production losses using quantile regression forests. *Energies*, *14*(1), 158. https://doi.org/10.3390/en14010158

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., & others. (2022). Forecasting: Theory and practice. *International Journal of Forecasting*, *38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001

Prinzhorn, D. W. E. (2023). *Benchmarking conformal prediction methods for time series regression* [Bachelor's Thesis]. University of Amsterdam.

Roebroek, J., & others. (2022). *Sklearn-quantile*. https://github.com/jasperroebroek/sklearn-quantile

Saporta, J. (2023). *Statistical tools for causal inference and forensic science* [PhD Thesis]. Iowa State University.

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242. https://doi.org/10.1080/01621459.2017.1319839

Wright, M. N., & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, *77*(1), 1--17. https://doi.org/10.18637/jss.v077.i01

Zhang, W., Quan, H., & Srinivasan, D. (2018). Parallel and reliable probabilistic load forecasting via quantile regression forest and quantile determination. *Energy*, *160*, 810–819. https://doi.org/10.1016/j.energy.2018.07.019