SETAR-Tree: A Novel and Accurate Tree Algorithm for Global Time Series Forecasting

ECML PKDD 2023

Presented by

Christoph Bergmeir

Rakshitha Godahewa
Geoffrey I. Webb
Daniel Schmidt

Faculty of Information Technology, Monash University, Australia

Motivation

- Tree-based algorithms dominated the M5 forecasting competition, in particular, globally (cross-series) trained LightGBM models (Makridakis et al., 2022).
- These algorithms are general-purpose methods, not specifically designed for forecasting.
  - No time series-specific splitting.
  - They calculate the average of the training outputs at leaf nodes.
    $->$ No extrapolation, weak capability to model trends.

Motivation (2)

- **Less noticed in the M5:**
  - Also globally trained linear models (Pooled Regression, PR) perform very well, if used with many lags to make them sufficiently complex (Montero-Manso and Hyndman, 2021).
  - A PR model with 400 lags would have been 17th place in the M5 Accuracy Track (Bandara et al., 2021).
- **Idea:**
  - If we use linear regression in the leaf, we have effectively a generalisation of a PR model: such a tree with only one node is a PR model.
  - If we build the tree, we get a hierarchical piecewise linear model.

Trees with Linear Models in Their Leafs: Linear Model Trees

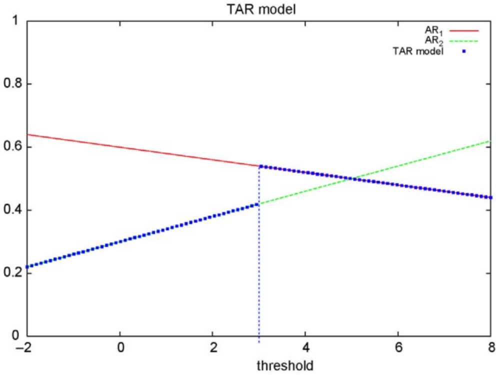
- Model trees (Quinlan, 1992)
- Cubist (Kuhn and Johnson, 2013)
- ctree (Hothorn et al., 2006)
- Gradient Boosting with Piece-Wise Linear Regression Trees (Shi et al., 2019)
  - `linear_tree` parameter in LightGBM
  - since 24/12/2020 in LightGBM master branch on github – after the M5

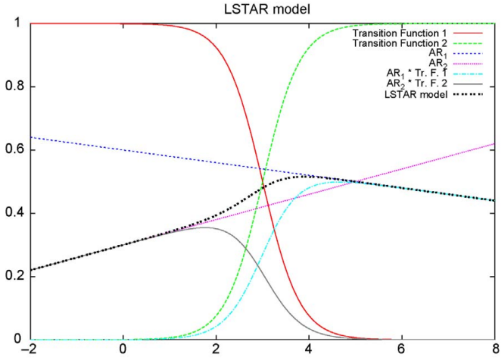
Piecewise Linear Models in Forecasting: Threshold Autoregressive Models

Piece-wise linear models that model the state space of a given prediction problem using multiple Autoregressive (AR) models.

Type	Characteristics
SETAR (Tong, 1993)	The simplest version of the TAR models. Defines regimes based on a particular lagged value of the time series itself. Requires estimates for the optimal lag and threshold.
STAR (Terasvirta, 1994)	Transition happens smoothly either using a past value of the series or an external variable. Transition functions: exponential (ESTAR) and logistic (LSTAR). Requires estimates for the parameters of the transition function.

TAR and STAR models





(Image source: Aznarte and Benitez (2010))

STR-Tree (da Rosa et al., 2008)

- A linear model tree that combines the concepts of regression trees and STAR models.
- **Shortcomings of STR-Tree**
  - No cross-learning / global modelling.
  - While STAR is a generalisation of SETAR, it is more complicated to implement and a lot slower: Each leaf needs to still operate on the full dataset, in a weighted way.
  - Stopping criteria can be improved.
  - No particular focus on time series forecasting (though those authors deem that a trivial extension).
  - Not widely noticed at the time (34 citations since 2008).

Our Research Contributions: SETAR-Tree and SETAR-Forest

- **Introduce an accurate, automated and publicly available tree-based algorithm for global time series forecasting.**
  - Incorporates forecasting-specific splitting and stopping criteria.
  - Uses the underlying concept of SETAR models (Tong, 1993) in defining the splits.
  - Trains a global PR (Gelman and Hill, 2006) model at leaf nodes.
  - A cross-series, piece-wise hierarchical linear model.
  - Requires minimal external hyperparameter tuning.
- **Introduce an accurate forest algorithm by extending the tree algorithm.**
- **The models can be simply executed using our R package, setartree:** `https://cran.r-project.org/web/packages/setartree`**.**

1 Introduction

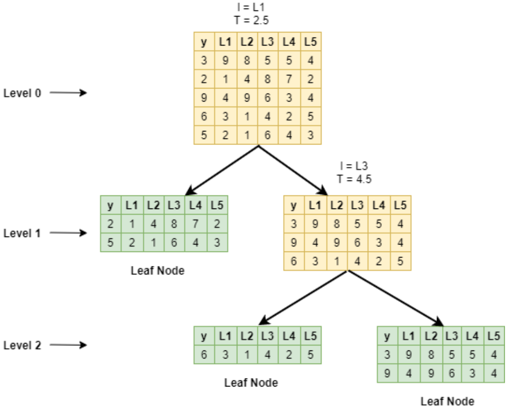
2 Methodology

3 Experiments

4 Results & Conclusions

SETAR-Tree: Splitting Criterion

- Same as the splitting in SETAR models.
- Finds the **optimal lag,** l and the **optimal threshold,** T that should be used to split each node using a grid search approach.
- **Left child:** instances with **Lag** $l < T$
  Right child: instances with **Lag** $l >= T$
- Trains a global PR model at each leaf node.
- Uses a leaf-wise tree growth approach.



SETAR-Tree: Grid Search

- To split a node into two children, its each lag and a set of thresholds (grid) are considered.
- For each lag and threshold pair in the grid, separate linear models are fitted to the two subsets of data formed by partitioning the chosen lag at the specified threshold.
- To speed up the method, the linear models are incrementally estimated.
- The optimal lag and threshold provide the split with the minimum sum of squared errors at the child nodes.

SETAR-Tree: Stopping Criteria

- **General linear F-test (Box, 1953)**
  - Determines whether there exists a remaining non-linearity of a set of training instances at a particular tree node.
  - The node is only further split if there exists a significant remaining non-linearity.
  - Uses sequence significance when defining the significance level (α) at each level of the tree.
- **Error reduction in node splitting**
  - A node is only split if the error reduction percentage between parent and child nodes is greater than or equal to a particular error threshold (3%).
- As the tree depth is internally controlled, this model requires a minimal amount of external hyperparameter tuning.

SETAR-Tree: Forecasting

- Identifies the leaf node corresponding with a given test instance by following the same optimal lags and thresholds that are previously used during node splitting starting from the root node.
- The prediction of the test instance is obtained using the trained PR model corresponding with its leaf node.

Training with Covariates

- A big strength of tree-based algorithm is the simple incorporation of covariates.
- This is also true for the SETAR-Tree.
- It can also be trained with external numerical and categorical covariates.
- The numerical covariates are treated in the same way as the time series lagged values.
- The categorical covariates are converted into a numerical format by applying one-hot encoding and are treated in the same way as numerical attributes.
- The past lags, numerical covariates and categorical covariates are all considered together when determining the optimal attributes and thresholds during node splitting.

SETAR-Forest

- Uses bagging.
- The goal is to get the most accurate, most diverse base-model pool.
- A collection of parallelly executed diverse SETAR-Trees.
- **Methods of diversifying SETAR-Trees**
  - Varying the initial significance level.
  - Varying the significance divider used to calculate the sequence significance.
  - Varying the error reduction percentage threshold.
- The forecasts provided by all trees are averaged to obtain the final forecasts.

1 Introduction

2 Methodology

3 Experiments

4 Results & Conclusions

Datasets

Dataset Name	No. of Time Series	Forecast Horizon	Frequency	Minimum Length	Maximum Length
Rossmann	1115	48	Daily	894	894
Kaggle Web Traffic	1000	59	Daily	744	744
Favorita	1000	16	Daily	1668	1668
M5	1490	28	Daily	1941	1941
Tourism Monthly	366	24	Monthly	67	309
Tourism Quarterly	427	8	Quarterly	22	122
Chaotic Logistic	100	8	-	592	592
Mackey-Glass	100	8	-	592	592

Performance Measures

- **Modified Symmetric Mean Absolute Percentage Error** (Suilin, 2017)

$$msMAPE = \frac{100\%}{N} \sum_{k=1}^{N} \frac{|F_k - Y_k|}{max(|Y_k| + |F_k| + \epsilon, 0.5 + \epsilon)} \quad where\ \epsilon = 0.1$$

- **Mean Absolute Scaled Error** (Hyndman and Koehler, 2006)

$$MASE = \frac{\sum_{k=M+1}^{M+h} |F_k - Y_k|}{\frac{h}{M-S} \sum_{k=S+1}^{M} |Y_k - Y_{k-S}|}$$

Benchmarks

- **Traditional univariate forecasting models**
  - Exponential Smoothing (ETS, Hyndman et al., 2008)
  - Auto-Regressive Integrated Moving Average (ARIMA, Box et al., 2015)
  - SETAR (Tong, 1993)
  - STAR (Terasvirta, 1994)
- **GFMs**
  - PR model (Gelman and Hill, 2006)
  - Cubist (Kuhn and Johnson, 2013)
  - Feed-Forward Neural Network (FFNN, Goodfellow et al., 2016)
  - Regression Tree (Loh, 2011)
  - CatBoost (Prokhorenkova et al., 2018)
  - LightGBM (Ke et al., 2017)
  - XGBoost (Chen and Guestrin, 2016)
  - Random Forest (RF, Breiman, 2001)

SETAR-Tree Variants

Variant	Stopping Criteria
Tree.Lin.Test	Significance of the linearity test.
Tree.Error.Red	Error reduction percentage of node splitting.
Tree.Lin.Test.Error.Red	Significance of the linearity test. Error reduction percentage of node splitting.

SETAR-Forest Variants

Variant	Tree Diversifying Attributes
Forest.Significance	Significance level and sequence significance divider of linearity test.
Forest.Error.Red	Error threshold used to measure the error reduction percentage.
Forest.Significance.Error.Red	Significance level and sequence significance divider of linearity test. Error threshold used to measure the error reduction percentage.

1 Introduction

2 Methodology

3 Experiments

4 Results & Conclusions

Mean msMAPE Results - Tree and Forest Variants

	Without Covariates								With Covariates		
	Ross-mann	Kag-gle	Favo-rita	M5	Tour (M)	Tour (Q)	Cha-otic	Mackey-Glass	Ross-mann	Kag-gle	Favo-rita
Tree.Lin.Test	**39.05**	71.04	83.01	**45.28**	23.94	17.04	41.83	0.00372	**11.53**	66.64	**94.77**
Tree.Error.Red	54.93	44.74	85.04	53.92	22.62	19.04	49.36	0.00661	15.20	45.01	97.01
Tree.Lin.Test.Error.Red	41.90	44.74	85.04	53.92	21.52	**15.59**	41.98	0.00372	12.09	**44.88**	97.01
Forest.Significance	41.65	48.13	85.06	53.91	**21.17**	**15.59**	41.30	**0.00296**	12.06	46.85	96.67
Forest.Error.Red	43.03	43.97	82.44	54.13	25.61	16.57	41.55	0.00307	12.07	47.94	95.32
Forest.Significance.Error.Red	40.73	**43.80**	**82.36**	54.13	22.16	15.97	**41.14**	**0.00296**	11.93	47.83	95.28

Mean msMAPE Results - All Models

	Without Covariates								With Covariates		
	Ross-mann	Kag-gle	Favo-rita	M5	Tour (M)	Tour (Q)	Cha-otic	Mackey-Glass	Ross-mann	Kag-gle	Favo-rita
ETS	43.98	46.24	87.67	78.22	**19.02**	**15.07**	50.33	1.02983	-	-	-
ARIMA	45.34	47.96	87.82	77.81	19.73	16.58	48.71	11.12100	-	-	-
SETAR	62.20	46.75	94.56	58.18	31.30	36.14	52.93	0.04079	-	-	-
STAR	72.89	46.82	96.30	95.01	32.58	34.08	44.82	0.02094	-	-	-
PR	64.45	111.48	85.04	53.92	21.56	17.07	52.27	0.01949	43.02	68.78	99.22
Cubist	**38.77**	55.69	85.75	146.12	19.96	16.02	43.03	0.26995	13.07	55.67	**85.63**
FFNN	197.35	164.74	119.40	94.97	199.47	199.77	42.78	60.53347	197.35	164.74	115.14
Regression Tree	55.48	61.88	101.21	65.94	64.34	115.02	44.72	3.42950	46.58	61.88	101.21
CatBoost	49.39	49.66	90.73	57.33	23.75	25.37	42.09	0.65735	39.91	47.97	90.90
LightGBM	56.16	55.63	96.83	**32.60**	22.18	19.72	42.53	0.56777	42.73	59.25	98.06
XGBoost	48.29	69.73	86.93	54.99	23.48	18.84	44.40	0.45676	48.41	65.20	89.41
RF	61.95	49.63	103.02	104.49	32.55	27.13	42.62	2.85191	46.53	49.90	101.62
Tree.Lin.Test.Error.Red	41.90	44.74	85.04	53.92	21.52	15.59	41.98	0.00372	12.09	**44.88**	97.01
Forest.Significance.Error.Red	40.73	**43.80**	**82.36**	54.13	22.16	15.97	**41.14**	**0.00296**	**11.93**	47.83	95.28

Main Observations

- **Overall, our proposed models outperform the benchmarks across most datasets (both with and without covariates).**
  - **Best SETAR-Tree variant:** Tree.Lin.Test.Error.Red
  - **Best SETAR-Forest variant:** Forest.Significance.Error.Red
- **Compared to Tree.Lin.Test and Tree.Lin.Test.Error.Red, Tree.Error.Red shows worse performance.**
  - Using the significance of the statistical linearity test individually or together with the error reduction percentage gained by node splitting are better options rather than using the error reduction percentage on its own as the stopping criterion of the SETAR-Tree.
- **The results of PR, Tree.Error.Red and Tree.Lin.Test.Error.Red are the same across the Favorita dataset.**
  - SETAR-Tree variants have only one node which is the parent node.

Conclusions

- **The proposed SETAR-Tree algorithm** provides more accurate forecasts compared to the state-of-the-art tree-based algorithms such as LightGBM, CatBoost and XGBoost across eight experimental datasets.
- **Training a global model at leaf nodes** often leads to better prediction accuracy compared to simple averaging of training instances.
- Considering the **significance of the statistical linearity test and the error reduction percentage gained by node splitting together** is a good option in automatically determining the maximum tree depth.

Conclusions Contd.

- SETAR-Forest shows the overall best performance as it **minimises the data, model and parameter uncertainties compared to individual trees**.
- SETAR-Forest provides more accurate results when **the individual trees are more diversified**.
- **Future work:**
  - Use lasso or other variants of linear models in the leaves.
  - Use other splitting criteria, such as AIC, cross-validation.

R Package: setartree

- Released **version 0.2.0** a month ago.
- **New Features**
  - Predictions intervals
  - Flexibility of normalising time series before model training.
    - Mean normalisation
    - Per-window normalisation
  - Faster execution of SETAR-Forest using parallelising the execution of SETAR-Trees.

Thank you

christoph.bergmeir@monash.edu

References I

Aznarte, J.L., Benitez, J.M., 2010. Equivalences between neural-autoregressive time series models and fuzzy systems. IEEE Transactions on Neural Networks 21, 1434–1444.

Bandara, K., Hewamalage, H., Godahewa, R., Gamakumara, P., 2021. A fast and scalable ensemble of global models with long memory and data partitioning for the m5 forecasting competition. International Journal of Forecasting doi:`https://doi.org/10.1016/j.ijforecast.2021.11.004`.

Box, G.E.P., 1953. Non-normality and tests on variances. Biometrika 40, 318–335.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time Series Analysis: Forecasting and Control. John Wiley and Sons.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

References II

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA. p. 785–794.

Gelman, A., Hill, J., 2006. Data analysis using regression and multilevel/hierarchical models. Analytical Methods for Social Research, Cambridge University Press.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Hothorn, T., Hornik, K., Zeileis, A., 2006. Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics 15, 651–674.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22, 679–688.

References III

Hyndman, R.J., Koehler, A.B., Ord, J.K., Snyder, R.D., 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T., 2017. LightGBM: A highly efficient gradient boosting decision tree, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA. p. 3149–3157.

Kuhn, M., Johnson, K., 2013. Applied predictive modeling. Springer, New York, NY, USA.

Loh, W.Y., 2011. Classification and regression trees. WIREs Data Mining and Knowledge Discovery 1, 14–23.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2022. The M5 accuracy competition: Results, findings and conclusions. International Journal of Forecasting .

References IV

Montero-Manso, P., Hyndman, R.J., 2021. Principles and algorithms for forecasting groups of time series: Locality and globality. International Journal of Forecasting 37, 1632–1653.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A., 2018. CatBoost: Unbiased boosting with categorical features, in: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.

Quinlan, J.R., 1992. Learning with continuous classes, in: 5th Australian Joint Conference on Artificial Intelligence, World Scientific. pp. 343–348.

da Rosa, J.C., Veiga, A., Medeiros, M.C., 2008. Tree-structured smooth transition regression models. Computational Statistics and Data Analysis 52, 2469–2488.

References V

Shi, Y., Li, J., Li, Z., 2019. Gradient boosting with piece-wise linear regression trees, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 3432–3438.

Suilin, A., 2017. kaggle-web-traffic. `https://github.com/Arturus/kaggle-web-traffic`.

Terasvirta, T., 1994. Specification, estimation and evaluation of smooth transition autoregresive models. Journal of the American Statistical Association 89, 208–218.

Tong, H., 1993. Non-linear time series: A dynamical system approach. Clarendon Press, Oxford.

Analysis of SETAR-Forest Size

No: of Trees	Mean msMAPE	Median msMAPE	Mean MASE	Median MASE
5	41.65	36.57	0.676	**0.644**
10	**41.14**	36.38	**0.667**	0.646
20	41.34	35.65	0.670	**0.644**
50	41.25	**35.60**	0.668	0.647

Computational Performance (in Minutes)

	Without Covariates								With Covariates		
	Rossmann	Kaggle	Favorita	M5	Tour (M)	Tour (Q)	Chaotic	Mackey-Glass	Rossmann	Kaggle	Favorita
ETS	7.52	5.59	8.08	14.43	5.00	1.06	0.07	0.22	-	-	-
ARIMA	163.80	11.16	123.00	25.41	47.00	6.78	0.18	0.30	-	-	-
SETAR	7.55	0.90	2.01	5.17	0.59	0.27	0.38	1.29	-	-	-
STAR	99.68	70.33	133.87	140.29	10.04	1.86	4.81	5.11	-	-	-
PR	0.55	0.63	0.65	1.58	0.03	0.02	0.02	0.02	0.90	0.25	0.55
Cubist	1.51	7.40	2.55	7.44	0.52	0.27	0.10	0.06	1.86	2.16	2.73
FFNN	7.57	18.88	223.80	24.01	0.74	0.25	25.00	4.61	10.16	24.22	352.20
Regression Tree	0.98	0.48	1.31	1.84	0.02	0.02	0.03	0.02	1.22	0.37	1.60
CatBoost	2.13	2.17	1.12	2.07	0.24	0.13	0.15	0.43	6.07	1.20	2.99
LightGBM	3.24	12.73	5.25	8.91	0.48	5.03	3.84	6.27	48.02	5.44	12.22
XGBoost	81.60	45.00	231.60	70.26	5.23	3.75	7.37	4.72	341.40	21.93	59.53
RF	1.49	58.30	2.76	4.71	6.89	0.10	0.28	0.12	10.43	6.44	14.29
Tree.Lin.Test.Error.Red	13.85	5.25	0.24	0.53	0.34	0.12	0.09	19.25	38.78	14.52	3.22
Forest.Significance.Error.Red	124.16	68.22	35.56	5.53	7.14	1.79	0.77	103.82	310.60	175.88	300.15

Prediction Times (in Seconds)

	Without Covariates								With Covariates		
	Ross-mann	Kag-gle	Favo-rita	M5	Tour (M)	Tour (Q)	Cha-otic	Mackey-Glass	Ross-mann	Kag-gle	Favo-rita
Tree.Lin.Test.Error.Red	1.20	0.65	0.01	0.01	0.29	0.32	0.06	0.16	1.52	0.96	0.87
Forest.Significance.Error.Red	12.00	6.50	0.10	0.10	2.90	3.20	0.60	1.60	15.20	9.60	8.70

Table 1: Results of Statistical Testing

Model	p_{Hoch}
Forest.Significance.Error.Red	-
Tree.Lin.Test.Error.Red	0.003
LightGBM	$< 10^{-30}$
ARIMA	$< 10^{-30}$
ETS	$< 10^{-30}$
CatBoost	$< 10^{-30}$
SETAR	$< 10^{-30}$
XGBoost	$< 10^{-30}$
STAR	$< 10^{-30}$
Cubist	$< 10^{-30}$
PR	$< 10^{-30}$
Regression Tree	$< 10^{-30}$
RF	$< 10^{-30}$
FFNN	$< 10^{-30}$