Generalized Autoregressive Score Trees and Forests

Andrew Patton
Duke University $\rightarrow$ UNSW Sydney

Yasin Simsek
Duke University

SoFiE 2024 Annual Meeting

Increase your GAS mileage

- The family of “generalized autoregressive score” (GAS) models, proposed by Creal et al. (2013) and Harvey (2013), nests many useful models for capturing time series dynamics:
  - ARMA models, see Box and Jenkins (1970)
  - ARCH/GARCH models, see Engle (1982) and Bollerslev (1986)
  - ACD models, see Engle and Russell (1998)
  - Recent surveys: Artemova et al. (2022a,b) and Harvey (2022)
- These models are good but are of course only approximations to the DGP.

★ We propose using machine learning methods to improve the forecasts from models in this class.

GAS trees and forests

- We propose a "**GAS tree**" that combines the parsimonious structure of the GAS model with the flexible, data-driven learning of decision trees, see Breiman et al. (1984, 2017)
  - The model parameters can vary across "branches" of the tree using a set of state variables
  - The resulting model can incorporate information from *outside* the GAS model, and allows for nonlinearities and interactions, while maintaining interpretability
- We also propose "**GAS forests**," analogous to the "random forests" of Breiman (2001), where many GAS trees are created using bootstrap samples of data and then averaged across trees.
  - In many applications random forests have been found to improve upon regression trees due to the reduction in variance obtained via averaging, see Hastie et al. (2009)

Applications

- We apply the proposed models in four distinct forecasting problems:
    1. Stock return volatility, baseline model is GARCH
    2. Predictive density, baseline model is t-GAS
    3. Stock-bond dependence, baseline model is t-GAS-copula
    4. Trade durations, baseline model is ACD
- In all four applications we find significantly better OOS forecasts using the tree/forest GAS model
- We can also *interpret* the source of the gains
    - The GAS tree/forest uncovers a nonlinearity that is known to improve the baseline model

Related literature

- **GAS models**: Creal et al. (2013), Harvey (2013), Creal et al. (2011), Harvey (2022)
  - www.gasmodel.com
- **ML + Econometrics**:
  - *Volatility forecasting:* Audrino and Bühlmann (2001), Christensen et al. (2022), Nguyen et al. (2022), Reisenhofer et al. (2022),Tetereva and Kleen (2022)
  - *Macro forecasting:* Goulet Coulombe (2024), Huber et al. (2020), Medeiros et al. (2021)
  - *Asset pricing:* Gu et al. (2020), Bianchi et al. (2021), Bryzgalova et al. (2023)
- **Local Estimation**: Tibshirani and Hastie (1987), Fan et al. (1998), Fan et al. (2009), Audrino and Bühlmann (2001), Oh and Patton (2024)

GAS Trees and Forests

GAS models

- The general form of the GAS Models of Creal et al. (2013) is:

$$
\begin{aligned}
y_t &\sim p(\cdot|\mathcal{F}_t; f_t, \theta, \nu) \\
\text{where} \quad f_t &= \omega + \beta f_{t-1} + \alpha s_{t-1} \\
s_t &= S_t \nabla_t
\end{aligned}
$$

where $\theta = [\omega, \beta, \alpha]'$ govern the dynamics of f_t, ν is a static parameter, ∇_t is the gradient of the log-likelihood and S_t is a scaling matrix.

- Parameter estimation is done via maximum likelihood:

$$
\hat{\theta}, \hat{\nu} = \underset{\theta,\nu}{\operatorname{argmax}} \sum_{t=1}^{T} \log p(y_t|\mathcal{F}_t; f_t, \theta, \nu)
$$

GAS Tree

- For a given tree structure with J terminal nodes, $\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_J\}$, the GAS(1,1) tree is based on the evolution equation:

$$f_t = \omega(\mathbf{Z_t}) + \beta(\mathbf{Z_t}) f_{t-1} + \alpha(\mathbf{Z_t}) s_{t-1}$$

where

$$[\ \omega(\mathbf{Z_t}), \beta(\mathbf{Z_t}), \alpha(\mathbf{Z_t})\ ] = \sum_{j=1}^{J} [\ \omega_j, \beta_j, \alpha_j\ ] \times \mathbb{1}(\mathbf{Z_t} \in \mathcal{P}_j)$$

and $[\ \omega_j, \beta_j, \alpha_j\ ]$ are the GAS parameters for partition j.

- Benefits of this approach:
  - Incorporates potential nonlinearities and interactions, Audrino and Bühlmann (2001)
  - Allows the parameters to vary across partitions, i.e. local parameters, Oh and Patton (2024)
  - Brings outside information to the model through $\mathbf{Z_t}$, Engle (2002)

GAS Forest

- Forests have been found to outperform trees (Hastie et al., 2009) in regression applications.
- Similar to bootstrap aggregation ("bagging") the GAS forest fits many trees using bootstrap samples of the original data.
  - Each sample only uses random subset of state variables, increasing variation
- Let $f_t^{(b)}(\mathbf{Z_t})$ denote the forecast from tree b at state variable $\mathbf{Z_t}$. The GAS Forest forecast is obtained simply as

$$f_t(\mathbf{Z_t}) = \frac{1}{B} \sum_{b=1}^{B} f_t^{(b)}(\mathbf{Z_t})$$

GAS Tree estimation overview

1. Estimate GAS model on full sample. Tree depth $\equiv m = 0$, partition $\boldsymbol{\mathcal{P}}^{(0)}$, parameter estimate $\hat{\theta}^{(0)}$
2. Define a new partition: $\boldsymbol{\mathcal{P}}_{j,k}^{(m+1)} = \boldsymbol{\mathcal{P}}_{-j}^{(m)} \cup \{\mathcal{P}_{j,k,L}^{(m)}, \mathcal{P}_{j,k,R}^{(m)}\}$ where $\boldsymbol{\mathcal{P}}_{-j}^{(m)} = \boldsymbol{\mathcal{P}}^{(m)} / \mathcal{P}_j$ contains all the partitions of $\boldsymbol{\mathcal{P}}^{(m)}$ except for the j^{th}, and the j^{th} partition is split into "left" and "right" subpartitions based on the k^{th} state variable and a threshold c
3. Estimate parameters for new subpartitions *taking as fixed* the parameters of the other partitions and evaluate the complete likelihood at $\left(\hat{\theta}_{-j}^{(m)}, \hat{\theta}_{j,k,L}^{(m+1)}, \hat{\theta}_{j,k,R}^{(m+1)}), \hat{\nu}^{(m)}\right)$
   - Massive saving in computation time. Also done by Athey et al. (2019).
4. Maximise step 3 across partitions j, state variables k, split points, c. Denote new set of partitions as $\boldsymbol{\mathcal{P}}^{(m+1)}$. Estimate complete model using these paritions, obtain $\hat{\theta}^{(m+1)}$.
5. Repeat until tree depth, m reaches pre-specified maximum value, M.

GAS Forest estimation overview

- "Just" repeat the previous slide for $B = 200$ trees, based on:
  - Data from a circular block bootstrap with block length 100
  - A randomly selected one-third of the state variables

Empirical Applications

Empirical applications

- We apply our new GAS tree and forest models in four out-of-sample forecasting analyses.
  1. S&P 500 index volatility
  2. S&P 500 index return predictive density
  3. Joint distribution of S&P 500 index and 10-year US govt bond returns
  4. High-frequency SPY trade durations

- We consider three benchmark models:
  - The baseline GAS specification (no tree)
  - "Distributional random forest" of Schlosser et al. (2019) (no dynamics)
  - "Small GAS Tree" similar to Audrino and Bühlmann (2001)

- We compare these models in terms of one-step-ahead predictive performance.
  - For volatility, we use the QLIKE loss function
  - For the others, we use the log-likelihood

- Data runs from 2000–2021, $T \approx 5500$
  - Split 30/30/40 for estimation/validation/testing

State variables

- We consider 10 state variables (at daily frequency) for use in GAS tree and forest model.
  - **Dependent variables:** S&P500 return and 10-year government bond return
  - **Volatility measures:** S&P500 RV (daily and monthly), VIX
  - **Macro variables:** Fed Funds Rate, 10yr-3mth Treasury yield, Default spread, Policy uncertainty index (Baker et al., 2016)
  - **Time**
- For our high-frequency application we also consider:
  - **Dependent variable:** Duration
  - **Market conditions:** Return, Amihud liquidity
- This large set of state variables differentiates us from:
  - Audrino and Bühlmann (2001), which only considers dependent variables as state variables
  - Oh and Patton (2024), which can only handle two state variables at time

Volatility forecasting

- The GARCH model of Bollerslev (1986) is widely used for forecasting asset return volatility, and has been shown to be difficult to beat in a range of applications, see Hansen and Lunde (2005).

$$\begin{aligned} y_t &= \sigma_t \epsilon_t; \quad \epsilon_t \sim \textit{iid } N(0,1) \\ \sigma_t^2 &= \omega + \beta \sigma_{t-1}^2 + \alpha y_{t-1}^2, \end{aligned}$$

- Creal et al. (2013) show that this model can be interpreted as a GAS model for the scale parameter of the Normal distribution.
- The "distributional random forest" (DRF) sets $\beta = \alpha = 0$ and allows the intercept, ω to vary with the forest structure.
- The "small GAS tree" model uses a regression tree with only y_{t-1} as a state variable.

Diebold-Mariano test results

Table 1: Out-of-sample performance of GARCH models using QLIKE loss

	GARCH	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-1.470				
Small Tree	-2.547	-0.414			
GAS Tree	-8.651	-5.577	-8.288		
GAS Forest	-6.409	-3.429	-2.777	4.973	
Avg loss	0.393	0.375	0.367	0.303	0.343

Diebold-Mariano test results

Table 1: Out-of-sample performance of GARCH models using QLIKE loss

	GARCH	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-1.470				
Small Tree	-2.547	-0.414			
GAS Tree	-8.651	-5.577	-8.288		
GAS Forest	-6.409	-3.429	-2.777	4.973	
Avg loss	0.393	0.375	0.367	0.303	0.343

- GAS Tree has lowest average loss, followed by GAS Forest.

Diebold-Mariano test results

Table 1: Out-of-sample performance of GARCH models using QLIKE loss

	GARCH	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-1.470				
Small Tree	-2.547	-0.414			
GAS Tree	-8.651	-5.577	-8.288		
GAS Forest	-6.409	-3.429	-2.777	4.973	
Avg loss	0.393	0.375	0.367	0.303	0.343

- GAS Tree and GAS Forest significantly outperforms all benchmarks.

Diebold-Mariano test results

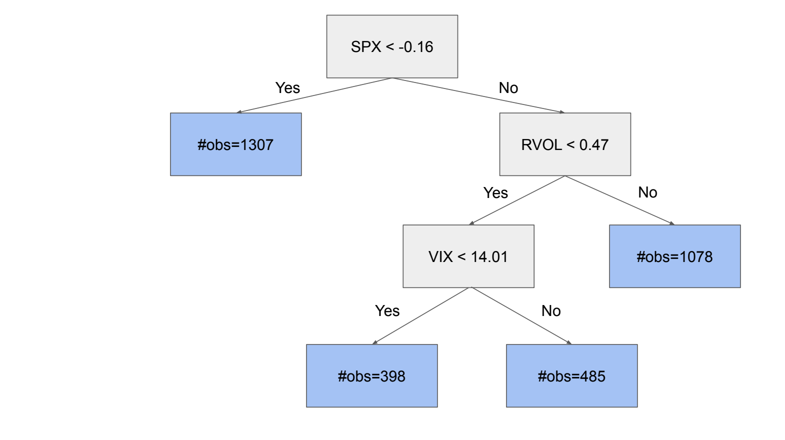
Table 1: Out-of-sample performance of GARCH models using QLIKE loss

	GARCH	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-1.470				
Small Tree	-2.547	-0.414			
GAS Tree	-8.651	-5.577	-8.288		
GAS Forest	-6.409	-3.429	-2.777	4.973	
Avg loss	0.393	0.375	0.367	0.303	0.343

- We find that the GAS Tree outperforms the GAS Forest, in contrast with both the econometrics and the machine learning literature. $\Longrightarrow$ bias-variance tradeoff see Hastie et al. (2009)

Optimal GARCH tree

Figure 1: Estimated GARCH tree



- The optimal tree has 4 "leaves" with 3 state variables:

  SPX: neg vs pos returns

  RVOL: low vs high volatility days

  VIX: low vs high variance risk premium

Application #2: Multivariate density forecasting

- We consider S&P 500 index and 10-year Treasury Bond returns as dependent variables.
- To allow for tail dependence we use the the Student's t copula, as in Janus et al. (2014).

$$\begin{aligned} \mathbf{u}_t &\sim \mathbf{C}_{\text{Student}}(\rho_t, \nu), \quad \rho_t = \frac{\exp\{\tilde{\rho}_t\} - 1}{\exp\{\tilde{\rho}_t\} + 1} \\ \tilde{\rho}_t &= \omega + \beta \tilde{\rho}_{t-1} + \alpha s_{t-1} \end{aligned}$$

  where s_{t-1} is a complicated function of the log-likelihood (see paper for equation).

- We complete the model with t-GAS models for the marginal distributions.

Diebold-Mariano test results

Table 2: Out-of-sample performance of t Copula GAS models using negative log-likelihood loss

	GAS	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	2.598				
Small Tree	-1.451	-2.811			
GAS Tree	-1.451	-2.811	—		
GAS Forest	-3.680	-4.092	-0.795	-0.795	
Avg Loss	-0.079	-0.063	-0.084	-0.084	-0.087

Diebold-Mariano test results

Table 2: Out-of-sample performance of t Copula GAS models using negative log-likelihood loss

	GAS	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	2.598				
Small Tree	-1.451	-2.811			
GAS Tree	-1.451	-2.811	—		
GAS Forest	-3.680	-4.092	-0.795	-0.795	
Avg Loss	-0.079	-0.063	-0.084	-0.084	-0.087

- Benchmark GAS model is beaten by by both the “small tree” and the GAS tree models, and significantly beaten by the GAS forest.

Diebold-Mariano test results

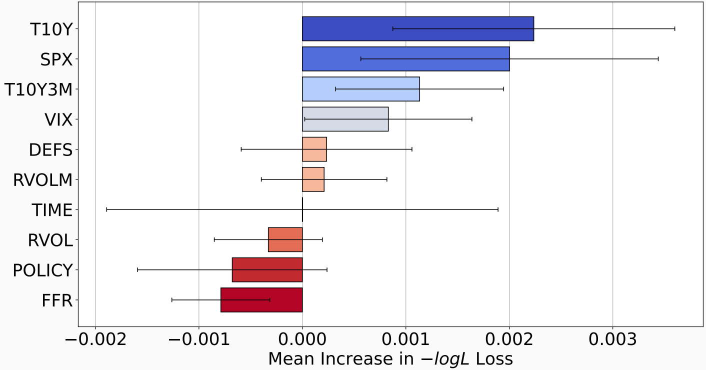
Table 2: Out-of-sample performance of t Copula GAS models using negative log-likelihood loss

	GAS	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	2.598				
Small Tree	-1.451	-2.811			
GAS Tree	-1.451	-2.811	—		
GAS Forest	-3.680	-4.092	-0.795	-0.795	
Avg Loss	-0.079	-0.063	-0.084	-0.084	-0.087

- The best performing model for t-copula is GAS forest, but does not significantly beat either of the tree models (which turn out to be identical).

GAS Forest variable importance

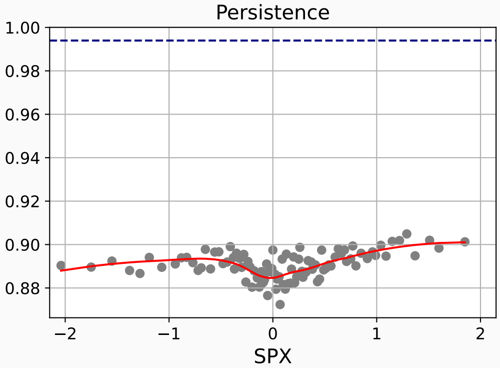
Figure 2: Leave-one-out variable importance for the Student's t copula GAS forest



- Blue means implies omitting the variable hurts performance $\implies$ the variable is important
- Horizontal lines are 95% confidence intervals from D-M test.

GAS Forest persistence parameter (β)

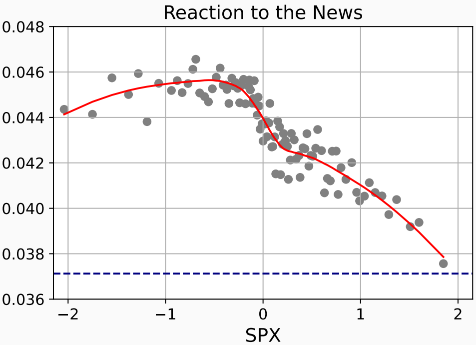
Figure 3: Parameter estimate as a function of S&P 500 returns for the Student's t copula GAS forest



- The persistence of the GAS forest model is roughly unrelated to the stock market return.

GAS Forest reaction to news parameter (α)

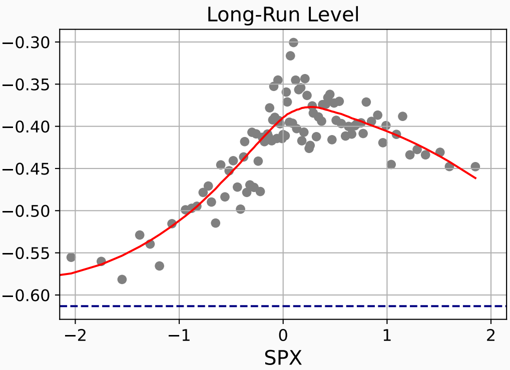
Figure 4: Parameter estimate as a function of S&P 500 returns for the Student's t copula GAS forest



- The α parameter is 0.046 when stocks are down, while it is 0.038 when stocks are up.
- This is consistent with investors paying closer attention to bad news than good news, a finding similar to that of Patton and Sheppard (2015) in a different context.

GAS Forest long-run level parameter ($\omega/(1-\beta)$)

Figure 5: Parameter estimate as a function of S&P 500 returns for the Student's t copula GAS forest



- This is can be interpreted as a "flight-to-quality" effect, with low stock market returns leading to more negative comovements between the stock and bond markets.

Summary

- We propose methods to improve the forecasts from GAS models using ML methods
  - **GAS trees** combine the parsimonious structure of the GAS model with the flexibility of decision trees (Breiman et al., 1984, 2017)
  - **GAS forests**, analogous to the random forests of Breiman (2001), average the forecasts from many GAS trees each produced on a bootstrap sample of the original data.
- We apply the proposed GAS tree and GAS forest models in four distinct forecasting applications
  - Tree/Forest models significantly outperform benchmark models in all cases
  - The source of the improvements is from capturing well-known empirical regularities, e.g. the leverage effect in volatility and the flight-to-quality effect in stock-bond correlations.
  - Trees and forests may be used to uncover new features not yet incorporated into models.

Additional Materials

Estimation Algorithm

- The estimation of GAS trees and GAS forests is computationally demanding.
  - finding the optimal state variables and thresholds from the set of candidate variables
  - given these estimates, estimating the parameters of the GAS model
- We use cluster computing and a "greedy" estimation algorithm related to that of Breiman et al. (1984) and Audrino and Bühlmann (2001).
- Standard tree estimation algorithms involves estimating a regression separately for each terminal node.
- We modify this aspect of the algorithm, and retain the autoregressive nature of the GAS model.

Estimation Algorithm: Step 1

- We use the following estimation algorithm to estimate the tree structure or, equivalently, to find the optimal partition $\mathcal{P}$. Estimation of the GAS tree involves Steps 1–5 below, and the GAS forest additionally uses Step 6.

Step 1: Denote the entire sample as the trivial partition $\mathcal{P}^{(0)}$. Estimate the parameters of the model using MLE since there is no tree structure in this case, and denote these as $(\hat{\theta}^0, \hat{\nu}^0)$:

$$(\hat{\theta}^0, \hat{\nu}^0) \quad = \quad \underset{\theta,\nu}{\operatorname{argmax}} \frac{1}{T} \sum_{t=1}^{T} \log p(y_t; f_t(\theta), \nu)$$

Estimation Algorithm: Step 2

Step 2: Let say $\boldsymbol{\mathcal{P}}^{(m)}$ is the partition leftover from previous step. Define a new partition:

$$\boldsymbol{\mathcal{P}}_j^{(m+1)} = \boldsymbol{\mathcal{P}}_{-j}^{(m)} \cup \{\mathcal{P}_{j,L}^{(m)}, \mathcal{P}_{j,R}^{(m)}\}$$

- $\boldsymbol{\mathcal{P}}_{-j}^{(m)} = \boldsymbol{\mathcal{P}}^{(m)} / \mathcal{P}_j$ contains all the partitions of $\boldsymbol{\mathcal{P}}^{(m)}$ except for the j^{th}
- the j^{th} partition is split into "left" and "right" subpartitions based on the k^{th} state variable and a threshold c

$$\begin{aligned}
\mathcal{P}_{j,L}^{(m)} &= \{\mathbf{Z_t} : \mathbf{Z_t} \in \mathcal{P}_j^{(m)} \quad \text{and} \quad Z_{t,k} \leq c\} \\
\mathcal{P}_{j,R}^{(m)} &= \{\mathbf{Z_t} : \mathbf{Z_t} \in \mathcal{P}_j^{(m)} \quad \text{and} \quad Z_{t,k} > c\}
\end{aligned}$$

Estimation Algorithm: Step 3 and 4

Step 3: Estimate the parameters for new subpartitions, taking the parameters of the other partitions, $\hat{\theta}_{-j}^{(m)}$, as fixed:

$$(\hat{\theta}_{j,L}^{(m+1)}, \hat{\theta}_{j,R}^{(m+1)}) \quad = \quad \underset{\theta_L, \theta_R}{\operatorname{argmax}} \frac{1}{T} \sum_{t=1}^{T} \log p(y_t; f_t(\hat{\theta}_{-j}^{(m)}, \theta_L, \theta_R), \hat{\nu}^{(m)})$$

Compute the log-likelihood value at estimated parameter values.

$$\log p(y; \mathcal{P}_j^{(m+1)}) = \frac{1}{T} \sum_{t=1}^{T} \log p(y_t; f_t(\hat{\theta}_{-j}^{(m)}, \hat{\theta}_{j,L}^{(m+1)}, \hat{\theta}_{j,R}^{(m+1)}), \hat{\nu}^{(m)})$$

Step 4: Maximize the likelihood in step 3 over the partition j, state variable k, and threshold c. Denote the optimized new partition as $\mathcal{P}^{(m+1)}$ and estimate all model parameters and denote $\hat{\theta}^{(m+1)}$.

Estimation Algorithm: Step 5 and 6

Step 5: Repeat steps 2-4 until the depth of the tree, m, reaches a prespecified maximum value, M. The depth of the tree controls the model complexity, and we consider values for M between one and six. We tune this parameter using a validation sample.

Step 6: For the GAS forest, repeat steps 2-5 for B trees. Each tree in the forest uses bootstrap data obtained from a circular block bootstrap (see, e.g., Politis et al., 1999), with block length of 100 observations, and a random selection of one-third of the total state variables. One-third is a common choice in the machine learning literature, see Hastie et al. (2009) for example.

Univariate Density Forecasting

- Our baseline model is the t-GAS model introduced by Creal et al. (2013), which captures both excess kurtosis, (the Student's t distribution), and time-varying volatility (the GAS structure).

$$
\begin{aligned}
y_t &= \sigma_t \epsilon_t; \quad \epsilon_t \sim \ i.i.d.\ t(v) \\
\sigma_t^2 &= \omega + \beta \sigma_{t-1}^2 + \alpha(1+3v^{-1}) \left(\frac{1+v^{-1}}{1-2v^{-1}} \left\{ 1 + \frac{v^{-1}}{1-2v^{-1}} \frac{y_{t-1}^2}{\sigma_{t-1}^2} \right\}^{-1} y_{t-1}^2 - \sigma_{t-1}^2 \right)
\end{aligned}
$$

- The $\{\cdot\}$ term implies a more moderate reaction to a large past return than in the GARCH model, as large returns are more common under the t distribution than the Normal distribution.

Univariate Density Forecasting: Diebold-Mariano Test Results

Table 3: Out-of-sample performance of t-GAS models using negative log-likelihood loss.

	t-GAS	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-5.396				
Small Tree	-3.555	1.571			
GAS Tree	-6.517	-1.240	-4.924		
GAS Forest	-5.485	1.555	-1.048	2.755	
Avg Loss	1.179	1.141	1.153	1.132	1.147

Univariate Density Forecasting: Diebold-Mariano Test Results

Table 3: Out-of-sample performance of t-GAS models using negative log-likelihood loss.

	t-GAS	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-5.396				
Small Tree	-3.555	1.571			
GAS Tree	-6.517	-1.240	-4.924		
GAS Forest	-5.485	1.555	-1.048	2.755	
Avg Loss	1.179	1.141	1.153	1.132	1.147

- t-GAS model is significantly out-performed by all four competing models, with Diebold-Mariano *t*-statistics less than -3.5 in all cases.

Univariate Density Forecasting: Diebold-Mariano Test Results

Table 3: Out-of-sample performance of t-GAS models using negative log-likelihood loss.

	t-GAS	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-5.396				
Small Tree	-3.555	1.571			
GAS Tree	-6.517	-1.240	-4.924		
GAS Forest	-5.485	1.555	-1.048	2.755	
Avg Loss	1.179	1.141	1.153	1.132	1.147

- The GAS tree significantly outperforms the “small GAS tree” and also the GAS forest.

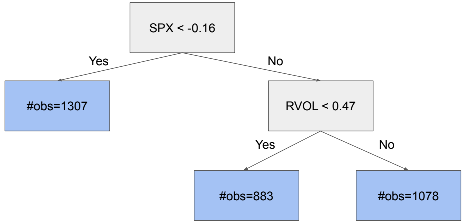
Univariate Density Forecasting: Diebold-Mariano Test Results

Table 3: Out-of-sample performance of t-GAS models using negative log-likelihood loss.

	t-GAS	DRF	Small GAS Tree	GAS Tree	GAS Forest
DRF	-5.396				
Small Tree	-3.555	1.571			
GAS Tree	-6.517	-1.240	-4.924		
GAS Forest	-5.485	1.555	-1.048	2.755	
Avg Loss	1.179	1.141	1.153	1.132	1.147

- The GAS tree also outperforms the DRF forecast, but the difference is not statistically significant at the 5% level.

Figure 6: The estimated t-GAS tree model.



- The optimal tree has 3 subsamples with 2 different state variables.
- These can be interpreted as
  (1) negative returns,
  (2) positive returns and low realized volatility,
  (3) positive returns and high realized volatility.

References

Artemova, M., Blasques, F., van Brummelen, J., and Koopman, S. J. (2022a). Score-driven models: Methodology and theory. In *Oxford Research Encyclopedia of Economics and Finance*.

Artemova, M., Blasques, F., van Brummelen, J., and Koopman, S. J. (2022b). Score-driven models: Methods and applications. In *Oxford Research Encyclopedia of Economics and Finance*.

Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *Annals of Statistics*, 47(2):1148–1178.

Audrino, F. and Bühlmann, P. (2001). Tree-structured generalized autoregressive conditional heteroscedastic models. *Journal of the Royal Statistical Society: Series B*, 63(4):727–744.

Baker, S. R., Bloom, N., and Davis, S. J. (2016). Measuring economic policy uncertainty. *Quarterly Journal of Economics*, 131(4):1593–1636.

Bianchi, D., Büchner, M., and Tamoni, A. (2021). Bond risk premiums with machine learning. *Review of Financial Studies*, 34(2):1046–1089.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327.

Box, G. E. P. and Jenkins, G. M. (1970). *Time series analysis: Forecasting and control*. Holden-Day, San Francisco.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees*. CRC Press.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Bryzgalova, S., Pelger, M., and Zhu, J. (2023). Forest through the trees: Building cross-sections of stock returns. *Journal of Finance*, forthcoming.

Christensen, K., Siggaard, M., and Veliyev, B. (2022). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, forthcoming.

Creal, D., Koopman, S. J., and Lucas, A. (2011). A dynamic multivariate heavy-tailed model for time-varying volatilities and correlations. *Journal of Business & Economic Statistics*, 29(4):552–563.

Creal, D., Koopman, S. J., and Lucas, A. (2013). Generalized autoregressive score models with applications. *Journal of Applied Econometrics*, 28(5):777–795.

Engle, Robert, F. (2002). New frontiers for ARCH models. *Journal of Applied Econometrics*, 17(5):425–446.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007.

Engle, R. F. and Russell, J. R. (1998). Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*, 66(5):1127–1162.

Fan, J., Farmen, M., and Gijbels, I. (1998). Local maximum likelihood estimation and inference. *Journal of the Royal Statistical Society: Series B*, 60(3):591–608.

Fan, J., Wu, Y., and Feng, Y. (2009). Local quasi-likelihood with a parametric guide. *Annals of Statistics*, 37(6B):4153.

Goulet Coulombe, P. (2024). The macroeconomy as a random forest. *Journal of Applied Econometrics*, forthcoming.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5):2223–2273.

Hansen, P. R. and Lunde, A. (2005). A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? *Journal of Applied Econometrics*, 20(7):873–889.

Harvey, A. C. (2013). *Dynamic Models for Volatility and Heavy Tails, with Applications to Financial and Economic Time Series*, volume 52. Cambridge University Press.

Harvey, A. C. (2022). Score-driven time series models. *Annual Review of Statistics and Its Application*, 9:321–342.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data mining, Inference, and Prediction*, volume 2. Springer.

Huber, F., Koop, G., Onorante, L., Pfarrhofer, M., and Schreiner, J. (2020). Nowcasting in a pandemic using non-parametric mixed frequency VARs. *Journal of Econometrics*, 232(1):52–69.

Janus, P., Koopman, S. J., and Lucas, A. (2014). Long memory dynamics for multivariate dependence under heavy tails. *Journal of Empirical Finance*, 29:187–206.

Medeiros, M. C., Vasconcelos, G. F., Veiga, Á., and Zilberman, E. (2021). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business & Economic Statistics*, 39(1):98–119.

Nguyen, T.-N., Tran, M.-N., and Kohn, R. (2022). Recurrent conditional heteroskedasticity. *Journal of Applied Econometrics*, 37(5):1031–1054.

Oh, D. H. and Patton, A. J. (2024). Better the devil you know: Improved forecasts from imperfect models. *Journal of Econometrics*, 242:105767.

Patton, A. J. and Sheppard, K. (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics & Statistics*, 97(3):683–697.

Politis, D. N., Romano, J. P., and Wolf, M. (1999). *Subsampling*. Springer Science & Business Media.

Reisenhofer, R., Bayer, X., and Hautsch, N. (2022). Harnet: A convolutional neural network for realized volatility forecasting. *arXiv preprint arXiv:2205.07719*.

Schlosser, L., Hothorn, T., Stauffer, R., and Zeileis, A. (2019). Distributional regression forests for probabilistic precipitation forecasting in complex terrain. *Annals of Applied Statistics*, 13(3):1564–1589.

Tetereva, A. and Kleen, O. (2022). A forest full of risk forecasts for managing volatility. *SSRN working paper 4161957*.

Tibshirani, R. and Hastie, T. (1987). Local likelihood estimation. *Journal of the American Statistical Association*, 82(398):559–567.