Generalized Autoregressive Score Trees and Forests[*]

Andrew J. Patton[a] and Yasin Simsek[a]

[a] *Department of Economics, Duke University*

This version: May 26, 2023

Abstract

We propose methods to improve the forecasts from generalized autoregressive score (GAS) models (Creal et al., 2013; Harvey, 2013) by localizing their parameters using decision trees and random forests. These methods avoid the curse of dimensionality faced by kernel-based approaches, and allow one to draw on information from multiple state variables simultaneously. We apply the new models to four distinct empirical analyses, and in all applications the proposed new methods significantly outperform the baseline GAS model. In our applications to stock return volatility and density prediction, the optimal GAS tree model reveals a leverage effect and a variance risk premium effect. Our study of stock-bond dependence finds evidence of a flight-to-quality effect in the optimal GAS forest forecasts, while our analysis of high-frequency trade durations uncovers a volume-volatility effect.

Keywords: Forecasting, machine learning, random forest, regression tree, volatility, copula, durations.

J.E.L. Codes: C22, C32, C53.

[*] We thank Saketh Aleti, Tim Bollerslev, Anna Bykhovskaya, Anna Cieslak, Peter Reinhard Hansen and Haozhe Zhang for valuable comments. Email: andrew.patton@duke.edu.

1 Introduction

Models for economic time series data that can capture time variation in features of the predictive density are widely used for policy making, investment decisions, risk management, and in many other applications. Such models include the autoregressive-moving average model of Box and Jenkins (1970), the ARCH/GARCH models of Engle (1982) and Bollerslev (1986), and many others. The family of "generalized autoregressive score" (GAS) models, proposed by Creal et al. (2013) and Harvey (2013), nests these time series models and others, and has been applied to a wide range of problems. Artemova et al. (2022a,b) and Harvey (2022) provide recent surveys of this large and growing literature.

Despite their success, score-driven models are inevitably only approximations to the true data generating process. We propose to use data mining methods from the machine learning literature to improve the performance of these models. Specifically, we propose a "GAS tree," that combines the parsimonious structure of the GAS model with the flexible, data-driven learning of decision trees Breiman et al. (1984, 2017). A GAS tree allows the parameters of the model to vary across "branches" of the tree, which are formed using a possibly large collection of state variables. This leads to a model that can incorporate information from outside the GAS model, and that allows for potentially complicated non-linearities and interactions. We further propose "GAS forests," analogous to the "random forests" of Breiman (2001) for linear regression, where we create many GAS trees using bootstrap samples of the original data and then average the forecasts from these trees. In many applications random forests have been found to improve upon regression trees due to the reduction in variance obtained via averaging, see e.g. Hastie et al. (2009).

The estimation of GAS trees and GAS forests is computationally demanding. It involves finding the optimal state variables and thresholds from the set of candidate variables, as well as estimating the parameters of the GAS model. We use cluster computing and a "greedy" estimation algorithm related to that of Breiman et al. (1984) for regression and Audrino and Bühlmann (2001) for GARCH trees. This algorithm finds a near-optimal solution and converges quickly. A key hyper-parameter in tree and forest models is the maximum depth of the tree (essentially, how many subsamples of the data will be considered) and we tune this parameter using a validation sample, separate from our forecast evaluation sample.

We apply the proposed GAS tree and GAS forest models in four empirically relevant problems: forecasting stock return volatility, the distribution of stock returns, the joint distribution of stock and bond returns, and high-frequency trade durations. As baseline models for these applications we use the GARCH model of Bollerslev (1986), the t-GAS model of Creal et al. (2011), a joint distribution model with Student's t margins and a Student's t copula, as in Janus et al. (2014), and the ACD model of Engle and Russell (1998). We then consider tree and forest extensions of these models, and in all four cases we find that the baseline model is significantly out-performed. For the two stock return applications, we find that the GAS tree provides the best out-of-sample forecasts. The estimated tree structures provide significantly better forecasts, and turn out to be relatively simple: we find evidence of a leverage effect, where the GAS model parameters differ depending on whether the lagged stock return was positive or negative, and a variance risk premium effect, where the model parameters differ depending on whether the difference between option-implied and historical volatilities is large or small.

In our study of the joint predictive distribution of stock and bond returns, we find that the GAS forest produces the best out-of-sample forecasts. Variable importance analyses indicate that the most important variables for the GAS forest are the lagged stock and bond returns themselves, indicating omitted nonlinearity in the baseline GAS model. We find evidence of a flight-to-quality effect, where higher bond returns or lower stock returns are associated with even more negative long-run correlations between the stock and bond markets. In our analysis of trade durations, defined as the time taken for 10,000 shares of the S&P 500 exchange traded fund, SPY, to be transacted, we again find the forest-based extension to be the preferred model. In this application the most important state variables are both measures of volatility, consistent with the well-known volume-volatility relationship (see, e.g., Karpoff, 1987).

This paper is part of the fast-growing literature using tools from machine learning in econometrics, see Varian (2014) and Athey and Imbens (2019) for recent surveys. Various studies have found that machine learning techniques bring significant gains over traditional econometric methods for forecasting applications. For example, Medeiros et al. (2021), Goulet Coulombe (2020) and Huber et al. (2020) show that tree-based methods, including random forests, can produce more accurate forecasts of important macroeconomic variables like unemployment and inflation. Gu et al. (2020) and Bianchi et al. (2021) show how machine learning methods can improve forecasts of stock and bond returns.

In addition to macroeconomic and financial forecasting, some recent papers have found success applying machine learning methods to volatility models such as the GARCH model of Bollerslev (1986) and the HAR model of Corsi (2009). For instance, Christensen et al. (2022) shows that neural networks and random forests significantly improve over HAR model, and Nguyen et al. (2022a,b) create hybrid stochastic volatility and GARCH models with recurrent neural networks. Reisenhofer et al. (2022) and Tetereva and Kleen (2022) use convolutional neural networks and random forests, respectively, combined with the HAR model to obtain improved out-of-sample forecasts.

This study also relates to a broadly defined "local estimation" literature. Tibshirani and Hastie (1987), Fan et al. (1998) and Fan et al. (2009) use kernel-based methods to localize (quasi-) maximum likelihood models. A more recent strand of this literature includes Breiman (2001), Schlosser et al. (2019) and Athey et al. (2019), who use decision trees and random forests to localize regressions, parametric distributions, and GMM models respectively. Our paper is related to Oh and Patton (2021), which is part of the first strand of this literature. That paper's approach suffers from the curse of dimensionality, due to its use of kernel-based methods, and it additionally requires that all (or none) parameters of the baseline model are localized. In the next section, we show that our proposed approach can deal with a large number of state variables and permits a subset of parameters to be localized, allowing the researcher to impose more or less structure on the model as needed.

The remainder of the paper is structured as follows. In Section 2 we review the class of generalized autoregressive score models of Creal et al. (2013) and Harvey (2013) and introduce our new GAS tree and GAS forest models. Section 2 also includes computational details on the implementation of these models. Section 3 presents four empirical analyses, applying the new methods to forecasting volatility, correlation, and univariate and bivariate distributions. Section 4 concludes, and the appendix presents details on the derivations for the third application. A supplemental appendix contains additional results.

2 GAS Trees and Forests

The class of generalized autoregressive score (GAS) models of Creal et al. (2013) and Harvey (2013) provide a parsimonious and powerful way to capture time variation in the parameter(s) of a given probability density function. We describe this model below, and in Sections 2.2 and 2.3 we introduce tree- and forest-based extensions of this class of models.

2.1 GAS models

Let the dependent variable be denoted $\mathbf{y}_t \in \mathbb{R}^K$. Conditional on the information set $\mathcal{F}_t$, this variable is assumed to have a parametric predictive density p, with d-dimensional time-varying parameter f_t, and potentially a static parameter ν. The GAS(p,q) model specifies the evolution of f_t as:

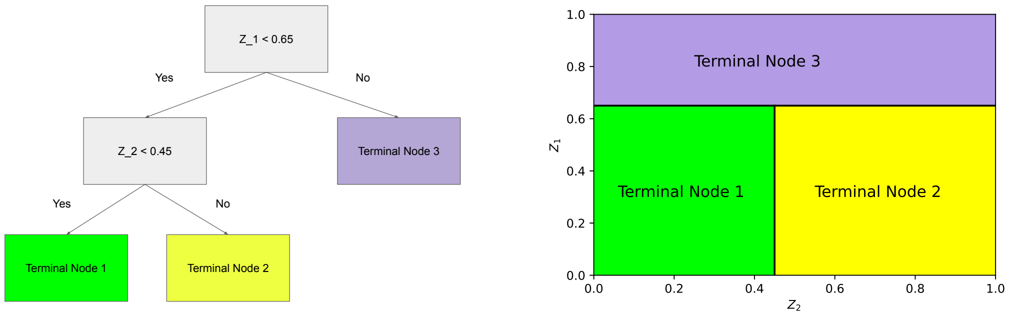
$$\begin{aligned} f_t &= \omega + \sum_{j=1}^{q} B_j f_{t-j} + \sum_{i=1}^{p} A_i s_{t-i} \qquad (1) \\ \text{where} \quad s_t &= S_t \cdot \nabla_t \\ \nabla_t &= \frac{\partial \log p(\mathbf{y}_t; f_t, \nu)}{\partial f_t'} \\ S_t &= \mathbb{E}_{t-1}[\nabla_t \nabla_t']^{-1} \end{aligned}$$

It is the appearance of the score, ∇_t, in the evolution equation for f_t that gives this class of models its name.[1] Similar to the well-known Newton-Raphson algorithm for numerical optimization, at each date t, f_t moves in the direction that most improves the model fit.

Let $\theta = (\omega, \text{vec}(B_1), ..., \text{vec}(B_q), \text{vec}(A_1), ..., \text{vec}(A_p))$ denote the vector of all GAS parameters of this model, making (θ, ν) the full set of unknown parameters. Since GAS models are "observation driven," as opposed to "parameter-driven", the likelihood function is available in closed form, and (θ, ν) can be estimated by maximum likelihood with low computational cost. This feature makes it feasible to consider tree- and forest-based extensions of this class of models, which we introduce below.

[1] We follow Creal et al. (2011) and use the inverse information matrix to scale the score in all of our applications, though other choices for this matrix are possible, such as the square root of this matrix, or simply the identity matrix.

Figure 1: A decision tree example



2.2 GAS Trees

Regression trees (Breiman et al., 1984; Breiman et al., 2017) are a type of nonparametric regression based on sequentially splitting the available data into partitions. The partitions are formed using one or more state variables, Z_t, and estimated threshold value(s), c. Figure 1 illustrates a simple tree structure. The left panel shows a tree with two state variables and specific thresholds, and the right panel shows the corresponding partition of the support of state variables. This hypothetical tree has three "terminal nodes" and implies a specific partition of the data, denoted $\boldsymbol{\mathcal{P}} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3\}$. Given a tree structure, a "regression tree" is obtained by estimating a linear regression separately for each of the terminal nodes in the tree. In so doing, regression trees allow for nonlinearities and multi-way interactions, greatly generalizing the baseline regression model. Naturally this flexibility makes trees prone to overfit the training data, and therefore, trees must be regularized, or "pruned." We describe the estimation and regularization methods we use for GAS trees and forests in Section 2.4.

We adapt the idea of regression trees for application to generalized autoregressive score (GAS) models. For a given tree structure with J terminal nodes, $\boldsymbol{\mathcal{P}} = \{\mathcal{P}_1, ..., \mathcal{P}_J\}$, the GAS(1,1) tree is based on the evolution equation:

$$\begin{aligned} f_t &= \omega(\mathbf{Z_t}) + \beta(\mathbf{Z_t}) f_{t-1} + \alpha(\mathbf{Z_t}) s_{t-1} \qquad (2) \\ \text{where} \quad \theta(\mathbf{Z_t}) &= \sum_{j=1}^{J} \theta_j \mathbb{1}(\mathbf{Z_t} \in \mathcal{P}_j) \end{aligned}$$

where $\theta_j \equiv [\omega_j, \beta_j, \alpha_j]$ are the GAS parameters for partition j, and s_t is as in equation (1). By allowing the parameters of the GAS model to vary across partitions we greatly increase the flexibility of this class of models to fit the data. Furthermore, by retaining the GAS structure for each partition, we can more easily interpret *how* the tree structure improves the fit of the model, in contrast with more "black box" machine learning algorithms.

The parameters of the predictive density that are assumed constant in the baseline GAS model, denoted ν above, can either be held constant across partitions or can be allowed to vary.[2] In our description below we impose they are fixed across partitions.

2.3 GAS Forests

"Random forests" (Breiman, 2001) are an extension of regression trees designed to reduce the estimation error in predictions, while retaining the information contained in the tree-based forecast, see Hastie et al. (2009) for example. Similar to bootstrap aggregation, or "bagging," a random forest is populated by trees that are each estimated on a bootstrap sample of the original data. In addition, each tree uses only a randomly-selected subset of the original state variables. The predictions from each of these trees are then averaged to obtain the random forest forecast.

If we denote trees in the random forest as $\boldsymbol{\mathcal{P}_b}$ for $b = 1, ..., B$, and the forecast from each tree for a given value of the vector of state variables, $\mathbf{Z_t}$ as $f_t^{(b)}(\mathbf{Z_t})$, obtained using equation (2), then the GAS forest forecast is obtained simply as

$$f_t(\mathbf{Z_t}) \quad = \quad \frac{1}{B}\sum_{b=1}^{B} f_t^{(b)}(\mathbf{Z_t}) \qquad (3)$$

We next turn to the estimation of the tree structure used in GAS trees and forests.

2.4 Estimating GAS Trees and Forests

The estimation of a GAS tree requires finding the optimal state variables and thresholds from the set of candidate variables, as well as estimating the parameters of the GAS model. Finding the global optimum of this optimization problem is computationally infeasible in

[2] In the kernel-based local M-estimation approach of Oh and Patton (2021) it is not possible to allow only a subset of parameters to vary with the state variable(s); the framework adopted in that paper requires an "all-or-nothing" assumption on parameter variation.

even moderately-sized regression tree applications, and to reduce the computational burden Breiman et al. (1984) proposed a greedy estimation algorithm that finds a near-optimal solution and converges quickly. The algorithm finds a state variable and a threshold to locally minimize the prediction error at each splitting step, continuing until a stopping criteria is satisfied.

Standard regression tree estimation involves estimating a regression separately for each terminal node in the tree, but given the autoregressive nature of GAS models, this is not possible for our application. We propose a modified estimation algorithm similar to Audrino and Bühlmann (2001) that uses a tree structure for the GAS model parameters and retains the autoregressive structure for f_t.

In the case that the number of terminal nodes, J, is one, there is no tree structure and the original GAS model is estimated via maximum likelihood:

$$(\hat{\theta}_T, \hat{\nu}_T) = \underset{\theta,\nu}{\operatorname{argmax}} \frac{1}{T}\sum_{t=1}^{T} \log p(y_t; f_t(\theta), \nu) \tag{4}$$

For $J \geq 2$ we use the following estimation algorithm to estimate the tree structure or, equivalently, to find the optimal partition $\boldsymbol{\mathcal{P}}$. 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 $\boldsymbol{\mathcal{P}}^{(0)}$. Estimate the parameters of the model as in equation (4), and denote these as $(\hat{\theta}^0, \hat{\nu}^0)$.

Step 2: Define a new partition: $\boldsymbol{\mathcal{P}}^{(m+1)}_{j,k} = \boldsymbol{\mathcal{P}}^{(m)}_{-j} \cup \{\mathcal{P}^{(m)}_{j,k,L}, \mathcal{P}^{(m)}_{j,k,R}\}$ where $\boldsymbol{\mathcal{P}}^{(m)}_{-j} = \boldsymbol{\mathcal{P}}^{(m)}/\mathcal{P}_j$ 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

$$\begin{aligned} \mathcal{P}^{(m)}_{j,k,L} &= \{\mathbf{Z_t} : \mathbf{Z_t} \in \mathcal{P}^{(m)}_j \quad \text{and} \quad Z_{t,k} \leq c\} \\ \mathcal{P}^{(m)}_{j,k,R} &= \{\mathbf{Z_t} : \mathbf{Z_t} \in \mathcal{P}^{(m)}_j \quad \text{and} \quad Z_{t,k} > c\} \end{aligned} \tag{5}$$

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

$$(\hat{\theta}_{j,k,L}^{(m+1)}, \hat{\theta}_{j,k,R}^{(m+1)}) \quad = \quad \underset{\theta_L,\theta_R}{\text{argmax}} \, \frac{1}{T}\sum_{t=1}^{T} \log p(y_t; f_t(\hat{\theta}_{-j}^{(m)}, \theta_L, \theta_R), \hat{\nu}^{(m)}) \tag{6}$$

The compute the log-likelihood value at estimated parameter values.

$$\log p(y; \boldsymbol{\mathcal{P}}_{j,k}^{(m+1)}) = \frac{1}{T}\sum_{t=1}^{T} \log p(y_t; f_t(\hat{\theta}_{-j}^{(m)}, \hat{\theta}_{j,k,L}^{(m+1)}, \hat{\theta}_{j,k,R}^{(m+1)}), \hat{\nu}^{(m)}) \tag{7}$$

Step 4: Maximize equation (7) over the partition j, state variable k, and threshold c. Denote the optimized new partition as $\boldsymbol{\mathcal{P}}^{(m+1)}$ and estimate all the model parameters using equation (4), denote these as $\hat{\theta}^{(m+1)}$.[4]

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 of M between one and six. We tune this parameter using a validation sample.

Step 6: For the GAS forest, repeat steps 2-5 for $B = 200$ trees.[5] 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. The forecasts from each of the bootstrap trees are then averaged to obtain the GAS forest forecast.

All computations are done using the Duke Computing Cluster exploiting multiple computing nodes. We parallelize the split optimization steps, and use the Numba package to speed up the code. Estimating a single tree takes around five minutes with forty CPUs. We apply fixed-window estimation all models: we estimate the model parameters using the estimation sample and use those parameters to compute all out-of-sample forecasts.

[3] Optimizing the split is the most demanding step in the entire algorithm. This assumption significantly reduces the computational burden without hurting the results. Similar ideas are also implemented in the literature, for example Athey et al. (2019) uses a gradient approximation in the split selection step.

[4] In steps 3 and 4, parameter estimations are done by nonlinear solvers available in Scipy package of Python language. Following the *warm start* idea in the optimization literature, we use the estimates from the previous iteration, $\hat{\theta}^{(m)}$, as starting values for optimization procedure to accelerate the algorithm.

[5] In preliminary analyses we obtained very similar results for $B = 500$.

3 Forecast performance of GAS trees and GAS forests

We apply our new GAS tree and forest models in four out-of-sample forecasting analyses. We firstly consider forecasting S&P 500 return volatility using the GARCH model of Bollerslev (1986), followed by predicting the entire conditional density of S&P 500 returns using the "t-GAS" model of Creal et al. (2013). In our third application we consider a flexible model for the joint distribution of S&P 500 returns and 10-year U.S. government bond returns, motivated in part by work on the switching sign of this correlation, see Guidolin and Timmermann (2006) and Baele et al. (2010), using the t-GAS copula to link t-GAS models for the marginal distributions, as in Janus et al. (2014). Finally, we consider the "autoregressive conditional duration" model of Engle and Russell (1998), using high frequency transaction data on the exchange traded fund tracking the S&P 500 index, the SPY. These four applications represent a range of predictive environments, and we provide evidence of the merits of GAS trees and forests in each of them.

In addition to the baseline GAS model for each application, we consider two other benchmark models. The first is a low-dimensional GAS tree ("small GAS tree"), in which we only consider the lag of the dependent variable(s) as a state variable. This model is similar to that of Audrino and Bühlmann (2001), and comparing the GAS tree and forest forecasts with this benchmark reveals the benefits of using a larger set of state variables. The third benchmark model is the "distributional random forest" of Schlosser et al. (2019). This model has no time series dynamics, but can provide a flexible distributional fit through the forest structure.

We compare all models in terms of one-step-ahead predictive performance.For the volatility forecasting application, we use the QLIKE loss function with realized volatility as the volatility proxy, see Patton (2011) for details. For the remaining applications we use the negative log-likelihood, which is a consistent scoring rule for density forecasts, see Gneiting and Raftery (2007). We conduct Diebold and Mariano (1995) tests of equal predictive accuracy, using Newey and West (1987) standard errors based on 10 lags.

3.1 Data description

We consider GAS trees and forests in four empirical applications. The first three of these use daily data on S&P 500 index returns and 10-year U.S. government bond returns from January 2000 to December 2021, a total of 5447 observations. Our fourth application uses high frequency trade durations for the S&P 500 index tracker fund, SPY, during the calendar year 2021, and has 5,100 observations. In all applications we split the sample into three sub-periods: an estimation sample (first 30% of observations), a validation sample for optimizing hyperparameters (next 30%), and a test sample for out-of-sample forecast comparisons (remaining 40%).

We consider ten state variables for use in the applications based on daily data, and we add three high frequency state variables in the fourth application. We firstly include the (lagged) return on the S&P 500 index and the 10-year U.S. government bond, to capture any nonlinearities omitted by the GAS models. We next consider three measures of volatility: 5-min subsampled realized volatility (RVOL) on the S&P 500 index, a one-month (backward-looking) rolling average of RVOL, motivated by prominent HAR model of Corsi (2009), and the VIX index, a measure of S&P 500 volatility implied by options prices. We then consider three measures from the fixed income market: the federal funds rate, the difference between 10-year and 3-month bond yields, representing the level and slope of the yield curve, and the "default spread" defined as the difference between BAA and AAA rate corporate bond yields. For our ninth state variable we include the economic policy uncertainty index proposed by Baker et al. (2016), based on newspaper coverage. This index tracks important policy related events like the failure of Lehman Brothers or presidential elections. We take a rolling monthly average of policy uncertainty index to eliminate the noise in the data. Our tenth state variable is time, to capture potential structural breaks, see for example Coulombe et al. (2020) and Goulet Coulombe (2020). In our fourth application, we additionally consider three high-frequency state variables: the first lag of duration, which can capture nonlinearities missed by the benchmark model, the return on SPY over the last trade event period, which can capture leverage-type effects, and the market liquidity of Amihud (2002), which can gauge whether the ACD model parameters differ during periods of high versus low liquidity.

We use the augmented Dickey-Fuller test (Dickey and Fuller, 1979) for each state variable to test for the presence of a unit root. We fail to reject the null of a unit root for the federal funds rate and the difference between 10 year and 3 month yield, and we take the first difference of these two variables. To avoid look-ahead bias, we use a one-period lag of the state variables when forming the tree and forest forecasts.

All of our data comes from the FRED database at the Federal Reserve Bank of St. Louis, with the following exceptions: the realized volatility data comes from the Oxford Realized Library; the high frequency data comes from the New York Stock Exchange's TAQ database; the 10 year bond return series is from Liu and Wu (2021); and the policy uncertainty series is from Baker et al. (2016).[6]

3.2 *Forecasting stock return volatility*

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). Assuming a zero conditional mean, the model is:

$$\begin{aligned} y_t &= \sigma_t \epsilon_t; \quad \epsilon_t \sim \; i.i.d.\ \mathcal{N}(0,1) \\ \sigma_t^2 &= \omega + \beta \sigma_{t-1}^2 + \alpha y_{t-1}^2. \end{aligned} \tag{8}$$

Creal et al. (2013) show that this model can be interpreted as a GAS model for the scale parameter of the Normal distribution. Given this equivalence, the GAS tree and forest models for this case can also be labeled GARCH tree and forest models. The "distributional random forest" (DRF) of Schlosser et al. (2019) in this application sets $\beta = \alpha = 0$ and allows the intercept, ω to vary with the forest structure, while the "small GAS tree" model of Audrino and Bühlmann (2001) uses a decision tree with only y_{t-1} as a state variable. We compare forecasts from these three models with those from the new GAS tree and GAS forest models introduced in Sections 2.2 and 2.3 in Table 1.

We observe that each variant of tree-based GARCH model (small GAS tree, GAS tree and GAS forest) significantly outperforms the benchmark GARCH model with t statistics all less than -2.5. Moreover, the GARCH tree and forest models significantly beat the

[6] The data for the latter two variables is available at https://sites.google.com/view/jingcynthiawu/yield-data and www.policyuncertainty.com.

Table 1: **Out-of-sample performance of GARCH models using QLIKE loss.** This table presents t-statistics from Diebold-Mariano tests of out-of-sample forecast performance (top four rows) and average out-of-sample QLIKE losses (bottom row). A negative t-statistic indicates that the model in the column had lower average loss than the model in the row, while a positive t-statistic indicates the opposite.

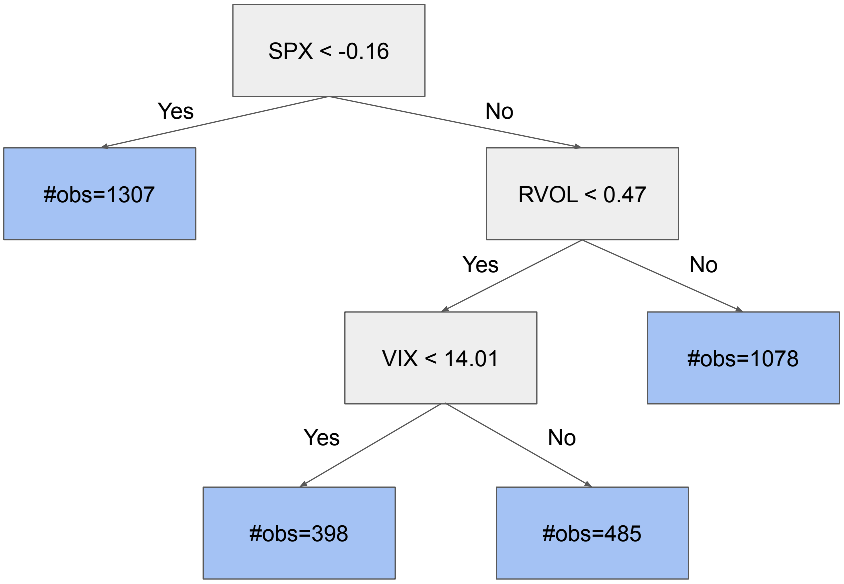
	GARCH	DRF	Small GARCH Tree	GARCH Tree	GARCH Forest
DRF	-1.470				
Small GARCH Tree	-2.547	-0.414			
GARCH Tree	-8.651	-5.577	-8.288		
GARCH Forest	-6.409	-3.429	-2.777	4.973	
Avg loss	0.393	0.375	0.367	0.303	0.343

DRF specification. Thus, in this first application, we find that tree structured models improve the out-of-sample forecast accuracy over simple GARCH, a conventional econometrics model, and DRF, a machine learning tool. Table 1 also shows that the "small GAS tree" outperformed by the GAS tree and forest models, with t-statistics below -2.7, revealing that external variables carry important information about future volatility.

Interestingly, and in contrast with both the econometrics and the machine learning literatures which generally find ensemble methods tend to outperform forecasts from individual models, we find that the GAS tree outperforms the GAS forest, with a t-statistic of nearly five. We interpret this result by noting that random forests have the potential to improve forecast accuracy through variance reduction at the cost of increasing bias, see for example Hastie et al. (2009). In our case, the variance reduction attained by the GAS forest cannot compensate the associated increased bias, leading to less accurate forecasts.

To understand the source of forecast gains from the GAS tree model, Figure 2 presents the estimated tree structure. The optimal tree depth was found to be three, with three different splitting variables. The algorithm first chooses the S&P 500 return with a threshold value -0.16 (its 40^{th} percentile) which approximately splits the sample using positive and negative market returns, consistent with an asymmetric reaction of future volatility to past returns, also known as a "leverage effect" (Black, 1976). The second split in the tree is for

Figure 2: **The estimated GARCH tree model.** This figure depicts the tree structure for the GARCH model. The tree's splits are based on SPX, RVOL and VIX, which refer to the S&P 500 return, realized volatility, and the option-implied volatility index respectively.



positive returns and uses realized volatility with a threshold of 0.47 (10.9% in annualized standard deviation form), corresponding to the 45th percentile of RVOL (conditional on the first split), thus approximately splitting positive return days into "high" and "low" volatility days. The third and final split is for low volatility days and uses VIX with a threshold of 14.01, corresponding to its 45th conditional percentile. Recalling that the "variance risk premium" (Carr and Wu, 2008; Bollerslev et al., 2009) can be approximated as the difference between VIX^2 and RVOL, the four terminal nodes of the tree in Figure 2 can be interpreted, approximately, as those associated with (1) negative returns, (2) positive returns and high realized volatility, (3) positive returns, low realized volatility and low variance risk premium, (4) positive returns, low realized volatility and high variance risk premium.[7]

[7] Employing a semi-structural regime switching model, Baele et al. (2010) finds that variance risk premium is an important economic factor in explaining stock return volatility.

Table 2: **Out-of-sample performance of t-GAS models using negative log-likelihood loss.** This table presents t-statistics from Diebold-Mariano tests of out-of-sample forecast performance (top four rows) and average out-of-sample negative log $\mathcal{L}$ losses (bottom row). A negative t-statistic indicates that the model in the column had lower average loss (i.e., a higher out-of-sample log-likelihood) than the model in the row, while a positive t-statistic indicates the opposite.

	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

3.3 *Forecasting the distribution of future stock returns*

We next consider the problem of forecasting the entire distribution of daily returns on the S&P 500 index. Our baseline model is the t-GAS model introduced by Creal et al. (2013), which captures both excess kurtosis, through the use of the Student's t distribution for the standardized residuals, and time-varying volatility, through the GAS structure for the scale parameter. Assuming a zero conditional mean, the t-GAS model is:

$$\begin{aligned} y_t &= \sigma_t \epsilon_t; \quad \epsilon_t \sim i.i.d.\ t(v) \\ \sigma_t^2 &= \omega + \beta\sigma_{t-1}^2 + \alpha(1+3v^{-1})\left(\frac{1+v^{-1}}{1-2v^{-1}}\left\{1+\frac{v^{-1}}{1-2v^{-1}}\frac{y_{t-1}^2}{\sigma_{t-1}^2}\right\}^{-1} y_{t-1}^2 - \sigma_{t-1}^2\right) \end{aligned} \tag{9}$$

where ν is the degrees of freedom parameter for the t distribution. As in Creal et al. (2013), ν is assumed constant, while σ_t^2 varies over time. The dynamics of σ_t^2 differs from the familiar GARCH structure when $\nu < \infty$, and simplifies to the GARCH model when $\nu \to \infty$. The $\{\cdot\}$ term in equation (11) 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.

Figure 3: **The estimated t-GAS tree model.** This figure depicts the tree structure for the t-GAS model. The tree's splits are based on SPX and RVOL, which refer to the S&P 500 return and realized volatility respectively.

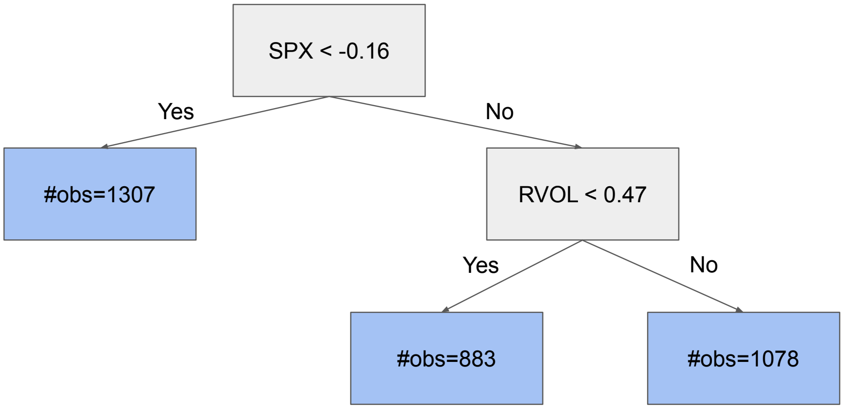


Table 2 presents comparisons of the t-GAS model with the distributional random forest (DRF), the "small GAS tree" (which only uses the lagged return as a state variable), and the GAS tree and forest models, both of which use all ten state variables described in Section 3.1. The first column shows that the t-GAS model is significantly out-performed by all four competing models, with Diebold-Mariano t-statistics less than -3.5 in all cases. We also observe that the GAS tree significantly outperforms the "small GAS tree" and also the GAS forest, both with p-values less than 0.01. The GAS tree also outperforms the DRF forecast, but the difference is not statistically significant at the 5% level.

Figure 3 shows the estimated t-GAS tree. Unlike the structure for the GARCH tree in the previous section, this tree only has depth of two, but those first two levels are identical to those of the GARCH tree.[8] The three terminal nodes of the tree have roughly equal numbers of observations, and can be interpreted as (1) negative returns, (2) positive returns and low realized volatility, (3) positive returns and high realized volatility.

[8] We use a grid of 19 values, corresponding to the 0.05, 0.10, ..., 0.95 quantiles of the state variable, for the threshold for each state variable, making our finding of identical threshold values less surprising.

3.4 Forecasting the joint distribution of stock and bond returns

We now focus on forecasting the joint distribution of stock and bond returns, using the S&P 500 index and 10-year Treasury bond for this purpose. We construct this model by combining t-GAS models for the marginal distributions, as utilized in the previous section, see equation (11), with a Student's t copula, as in Janus et al. (2014). Copulas are convenient tools for capturing the dependence between variables separately from the marginal distributions of each of the variables; see Patton (2013) for a review of these methods for economic time series. The use of a t copula allows for the possibility of tail dependence, or "joint crashes and joint booms." Although the marginal distributions and the copula are all from the Student's t family, this joint distribution is not bivariate Student's t distribution unless all three degrees-of-freedom parameters are identical; instead, this joint distribution allows for varying degrees of fat tails in each marginal distribution and the joint tails. The Student's t copula with GAS dynamics for the correlation parameter is:

$$\begin{aligned}
\mathbf{u}_t &\sim \mathbf{C}_{\text{Student}}(\rho_t, \nu) \qquad (10)\\
\rho_t &= \frac{\exp\{\tilde{\rho}_t\} - 1}{\exp\{\tilde{\rho}_t\} + 1}\\
\tilde{\rho}_t &= \omega + \beta\tilde{\rho}_{t-1}\\
&\quad +\alpha\left(\frac{2}{1-\rho_t^2}\right)\left(\frac{1+\rho_t^2}{g+(2g-1)\rho_t^2}\right)\left(w_t(x_{1,t}x_{2,t}-\rho_t) - \frac{\rho_t}{1+\rho_t^2}(w_t x_{1,t}^2 + w_t x_{2,t}^2 - 2)\right)
\end{aligned}$$

where $x_{i,t} \equiv F_{\text{Student}}^{-1}(u_{i,t}; \nu_i)$ uses the inverse t CDF with degrees-of-freedom parameter ν_i, and g and w_t are scalars defined in Appendix A. As in the univariate t-GAS model, we impose that the copula degrees of freedom parameter, ν is constant over time.

Table 3 shows the out-of-sample performance of competing models. We see that the benchmark GAS model significantly beats the distributional random forest (DRF), unlike in the two univariate applications, but it is beaten by both the "small tree" and the GAS tree models. The best-performing model is the GAS forest, which significantly outperforms the GAS model, with a t-statistic of -3.7. The GAS forest also significantly beats the DRF, but does not significantly beat either of the tree models. Interestingly, in this application the GAS tree reduces to the "small GAS tree" model, in that the only state variables selected for use in the tree structure are lags of the stock and bond returns.

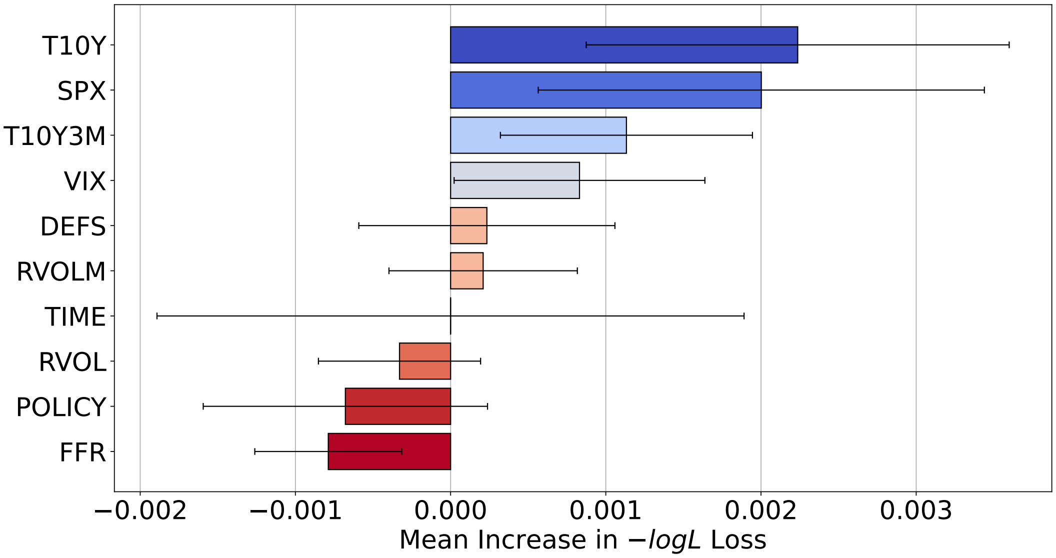
Table 3: **Out-of-sample performance of t Copula GAS models using negative log-likelihood loss.** This table presents t-statistics from Diebold-Mariano tests of out-of-sample forecast performance (top four rows) and average out-of-sample negative $\log \mathcal{L}$ losses (bottom row). A negative t-statistic indicates that the model in the column had lower average loss (i.e., a higher out-of-sample log-likelihood) than the model in the row, while a positive t-statistic indicates the opposite.

	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

In contrast with the univariate applications, the best-performing model is the GAS forest, not the GAS tree, and so we cannot present a tree diagram to better understand the structure of the best model. In its place, we consider two methods for interpreting the optimal model. Firstly, we conduct a *leave-one-out* analysis to measure the importance of each state variable. Specifically, we drop each state variable from the analysis, one at a time, and re-compute the optimal GAS forest forecasts. We then compare the average out-of-sample average loss from the original GAS forest and the GAS forest using one fewer state variable. If the difference is small, then the omitted state variable is unimportant, while if the difference is positive, then the omitted variable is important for forecast performance.[9] We can use Diebold-Mariano tests to determine whether the change in out-of-sample loss is statistically significant. Figure 4 presents the results of this analysis, and shows that the most important state variable for the GAS forest is the lagged bond return, T10Y, followed by the lagged stock market return, SPX. Omitting either of these significantly (at the 5% level) deteriorates the GAS forest forecasts. The slope of the term structure (T10Y3M)

[9] As this is an out-of-sample comparison of models, it is possible that the difference is negative, meaning that the smaller GAS forest is *preferred* to the original GAS forest. With a large enough sample size, including irrelevant state variables leads to no change, positive or negative, in out-of-sample loss, as such state variables will never be selected. In finite samples, however, irrelevant state variables may be mistakenly included.

Figure 4: **Leave-one-out variable importance for the Student's t copula GAS forest.** This figure plots the change in out-of-sample average negative log-likelihood between using the original GAS forest and a GAS forest with a state variable (listed on the y-axis) omitted. Positive values indicate a worsening of forecast performance, and thus that the omitted state variable is an important component of the original model. The horizontal lines represent 95% confidence intervals for the difference in average log-likelihoods.

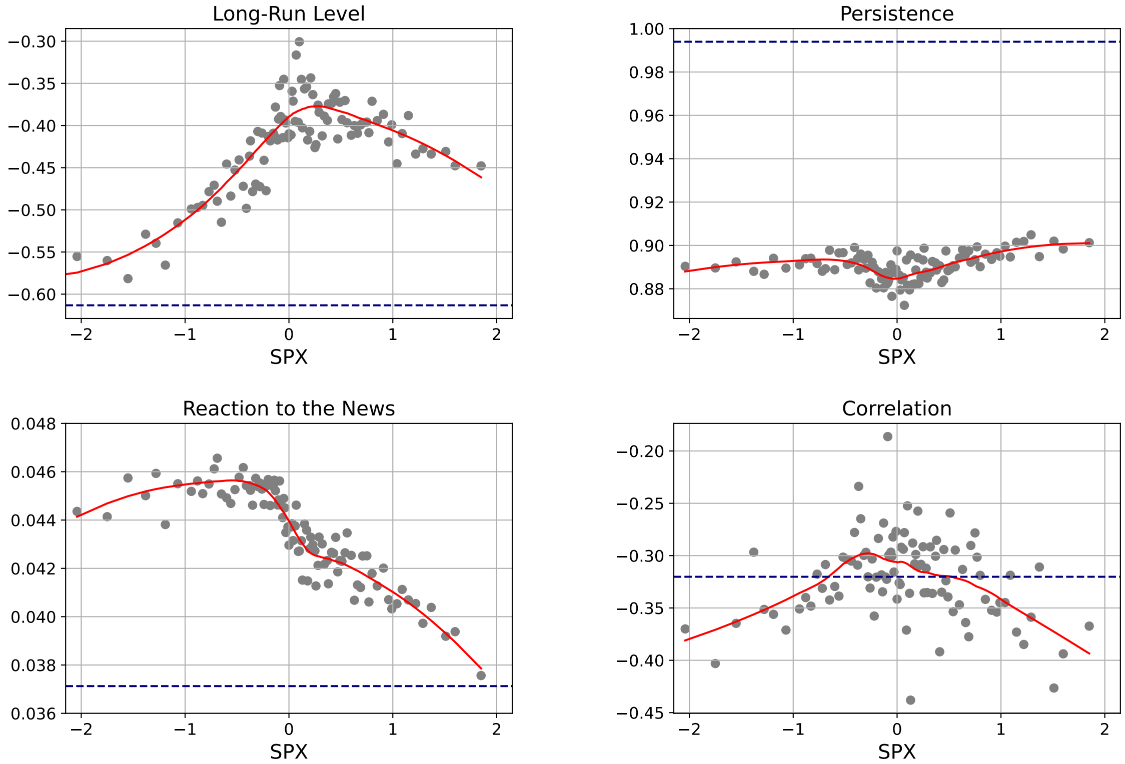


and the volatility index (VIX) are also found to be important for the quality of GAS forest forecasts. Interestingly, we observe a statistically significant *improvement* in forecast performance by omitting the Federal funds rate (FFR) as a state variable, indicating that this variable is unhelpful for out-of-sample forecasting, but is selected for inclusion in the forest often enough to deteriorate the forecast.

We next analyze the impact of the most important state variable on the GAS forest model by plotting the parameters of the GAS model (recall equation 1) as a function of the state variable, see Figure 5.[10] The parameters β and α are interpretable as the persistence and reaction-to-news of the model. The intercept, ω is not directly interpretable, and we instead plot $\omega/(1-\beta)$ which is interpretable as the long-run level of the GAS process. As the GAS forest involves averaging 200 bootstrap samples, each based on a random subset

[10] Similar plots for the other variables found to be significant in Figure 4 are presented in the supplemental appendix.

Figure 5: **Parameter estimates as a function of S&P 500 index returns (SPX) for the Student's t copula GAS forest.** This figure plots the average, across bootstrap samples, values of $\omega/(1-\beta)$ (upper-left), β (upper-right), and α (lower-left) from the GAS model in equation (1), as well as the average predicted correlation, ρ_t (lower-right) from that model. The state variable is discretized into bins based on 1% quantiles. The values for each of these quantities from the benchmark GAS model are plotted in horizontal dashed lines. The solid lines are local quadratic polynomials fitted to the grey dots.



of state variables, we construct this plot by averaging the GAS parameters within each 1% quantile of the state variable.

Figure 5 presents the results for the stock return as a state variable. The upper-left panel shows that the long-run correlation peaks when the stock market return is around zero, at about 0.35. It declines to about −0.45 as the stock return increases to 2%, while it declines markedly to nearly −0.6 when the stock return is −2%. This is interpretable as a "flight-to-quality" effect, with low stock market returns leading to more negative comovements between the stock and bond markets. Figure S.1 in the supplemental appendix plots the corresponding results with the bond return as the state variable, and that figure is also consistent with a flight-to-quality effect.

The upper-right panel of Figure 5 shows that the persistence of the GAS model is roughly unrelated to the stock market return. The lower-left panel shows that the GAS model reacts about 20% more strongly to news when the stock market is down versus up: 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.

The lower-right panel of Figure 5 shows the predicted correlation from the GAS forest as a function of the stock market return, and reveals an inverted U-shaped pattern, though with substantial noise. Without an underlying model to guide interpretation, one might be hesitant to draw too much from this panel. With the benefit of the GAS structure underlying our forest forecast, we know that this shape is primarily coming from the long-run level, $\omega/(1-\beta)$, in the upper-left panel, and that that relationship is strong. This reveals an important benefit of combining machine learning tools with economically motivated, and/or empirically successful, econometric models.

3.5 Forecasting market activity

Finally, we consider the problem of forecasting the time between consecutive trade events, known as a "trade duration."[11] Trade durations are a measure of market activity, and are important for high-frequency risk management and transaction cost minimization. We take as our benchmark the "autoregressive conditional duration" (ACD) model of Engle and Russell (1998). Denoting y_t as the time (in minutes) between consecutive trade events, the ACD model assumes an exponential distribution for y_t with a time-varying conditional mean, μ_t:

$$\begin{aligned} y_t &\sim \mathrm{Exp}(\mu_t) \\ \mu_t &= \omega + \beta\mu_{t-1} + \alpha y_{t-1}. \end{aligned} \tag{11}$$

Creal et al. (2013) show that the ACD model is also a special case of a GAS model, allowing us to consider it in our study of tree- and forest-based extensions of GAS models. See Bauwens and Hautsch (2009) for a review of ACD and related models.

[11] A "trade event" could be a single transaction occurring, or a total of x transactions occuring, or a total of $\$y$ value of transactions occuring, or a total of z shares being transacted, or some other event defined as a function of characteristics of transactions.

Table 4: **Out-of-sample performance of ACD models using negative log-likelihood loss.** This table presents t-statistics from Diebold-Mariano tests of out-of-sample forecast performance (top four rows) and average out-of-sample negative $\log \mathcal{L}$ losses (bottom row). A negative t-statistic indicates that the model in the column had lower average loss (i.e., a higher out-of-sample log-likelihood) than the model in the row, while a positive t-statistic indicates the opposite.

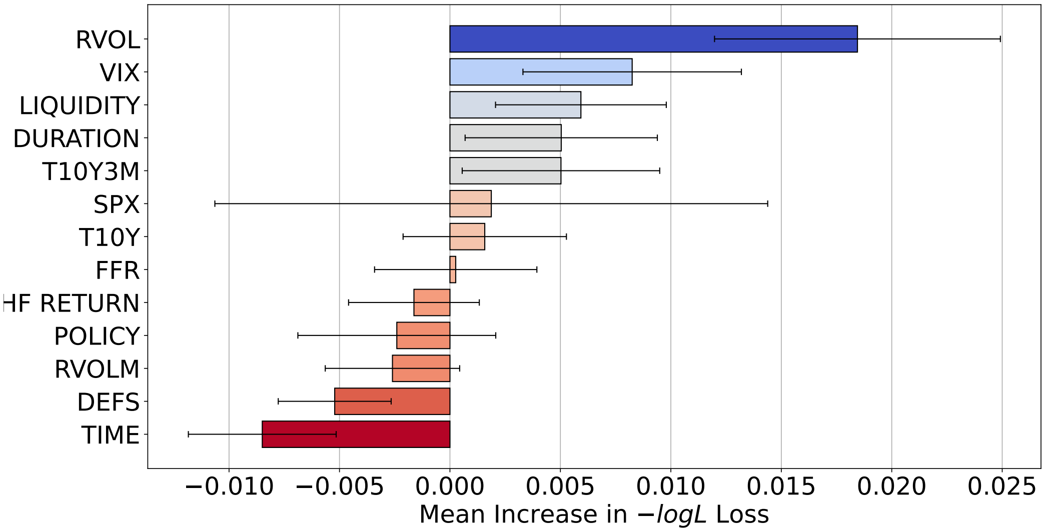
	ACD	DRF	Small ACD Tree	ACD Tree	ACD Forest
DRF	6.170				
Small ACD Tree	-2.448	-5.778			
ACD Tree	-1.004	-3.988	-0.458		
ACD Forest	-2.293	-9.358	-0.830	0.083	
Avg Loss	7.412	7.494	7.398	7.388	7.389

For our empirical analysis in this section, we use high-frequency data on SPY, an exchange traded fund tracking the S&P 500 index, between January 1^{st} 2021 and December 31^{st} 2021. We study the time taken for 10,000 shares of SPY to be transacted, leading to 5,100 durations during this sample period, and corresponding to an average duration of 14.9 minutes.

Table 4 presents the out-of-sample forecast performance of the baseline ACD model as well as the competing models considered in previous sections: the distributional random forest (DRF), the ACD tree using only lagged durations as a state variable (Small ACD tree), the ACD using all 13 state variables, and the ACD forest model. We firstly observe that the DRF model for durations, which is pure machine learning tool, is beaten by all other models including the benchmark with t-statistics all less than -2.4.

The baseline ACD model has higher loss in the out-of-sample compared with to tree- and forest-based extensions. Interestingly, the "small ACD tree" model, which only uses lagged duration as a state variable, significantly beats the baseline ACD model, while the "ACD tree" model, which considers 13 state variables *including* lagged duration, does not significantly beat the baseline model. This reveals the value in imposing some structure (namely, reducing the number of potential state variables) on the tree-based extension in

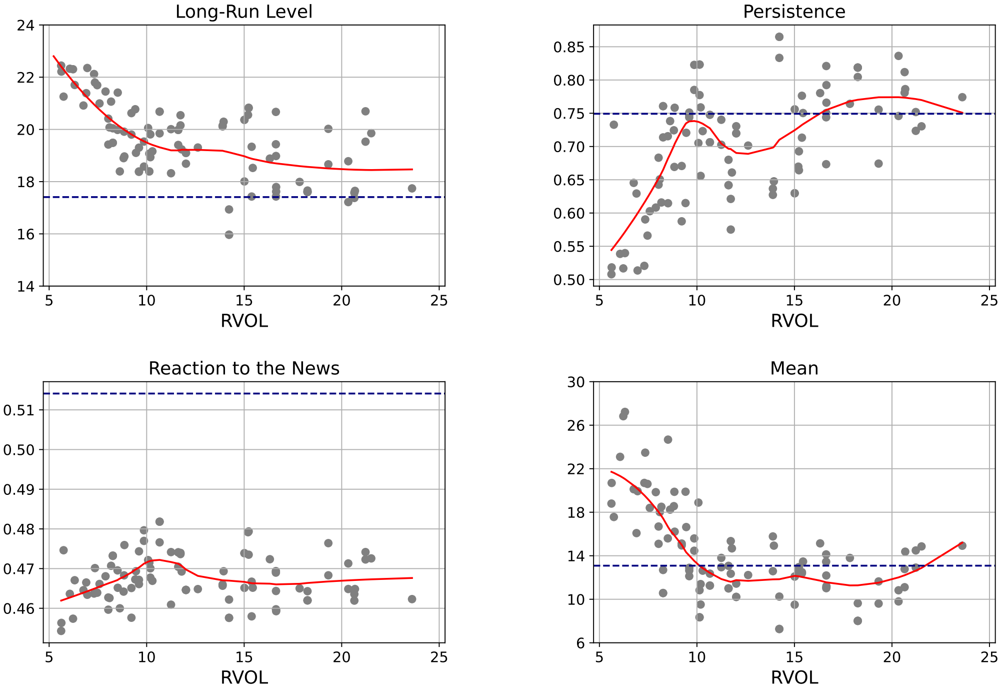
Figure 6: **Leave-one-out variable importance for the ACD forest.** This figure plots the change in out-of-sample average negative log-likelihood between using the original ACD forest and a ACD forest with a state variable (listed on the y-axis) omitted. Positive values indicate a worsening of forecast performance, and thus that the omitted state variable is an important component of the original model. The horizontal lines represent 95% confidence intervals for the difference in average log-likelihoods.



this application. The "ACD forest" model significantly beats the baseline ACD model, revealing the forecast gains available from averaging forecasts from randomly formed trees, consistent with Breiman (2001) in a linear regression setting.

To understand how forecast accuracy is improved in the ACD forest model, we calculate the variable importance measure for each of the state variables, introduced in the previous section, and present the results in Figure 6. The horizontal bars show the increase in average loss function from omitting a state variable, and a positive value indicates that that state variable is important for forecasting. The lines refer to 95% confidence intervals computed from Diebold-Mariano tests. We find that the volatility variables RVOL and VIX are two most important state variables, despite the fact that these are measured only daily, and so are constant within a trade day. The next two most important state variables are both high-frequency variables: Amihud (2002) liquidity, and duration. Interestingly, we find that omitting default spread and time from the set of potential state variables actually improves

Figure 7: **Parameter estimates as a function of realized volatility (RVOL) for the ACD forest model.** This figure plots the average, across bootstrap samples, values of $\omega/(1-\alpha-\beta)$ (upper-left), $\alpha+\beta$ (upper-right), and α (lower-left) from the ACD model in equation (1), as well as the average predicted duraction, μ_t (lower-right) from that model. The state variable is discretized into bins based on 1% quantiles. The values for each of these quantities from the benchmark ACD model are plotted in horizontal dashed lines. The solid lines are local quadratic polynomials fitted to the grey dots.



forecast accuracy, indicating these are harmful when used in a forest-based ACD model.[12]

In Figure 7 we plot the parameters of the forest ACD model as a function of the most important state variable, realized volatility (RVOL). We see that the long-run average duration implied by the model is highest when volatility is low, at around 23 minutes, and it steeply declines as volatility increases to about 10%, at around 19 minutes. Persistence, measured by $\alpha+\beta$ in the ACD model, is lowest when volatility is low, and it increases sharply with volatility to around 10% and is approximately flat beyond that. The parameter governing the reaction of the model to news, α, is essentially flat as a function of

[12] Figure S.4 in the supplemental appendix presents the optimal tree structure for the ACD tree. We find three terminal nodes in this structure, all reflecting the direction of the stock and bond markets. We find one state when the stock market is up (representing 55% of the sample), another when the stock market is down and the bond market is not strongly up (representing 92% of the remaining sample), and a small third state when the stock market is down and the bond market is strongly up (representing just 3% of the total sample).

volatility. The pattern for the forecasts from the ACD forest model, in the lower-right panel, shows that predicted durations are around 12 minutes when volatility is above 10%, while they are around double that when volatility is low. This is consistent with the positive volume-volatility relationship (see, e.g., Karpoff, 1987): volume and durations are negatively correlated (longer durations correspond to lower volumes, and vice versa) and so in periods of lower volatility average trade durations tend to be longer.

4 Conclusion

Since its publication a decade ago, the class of generalized autoregressive score (GAS) models of Creal et al. (2013) and Harvey (2013) has proven to be a popular, parsimonious way to capture time variation in the parameter(s) of a given model. Its parsimonious nature, however, means that some important exogenous information or nonlinearities may be neglected, and how to best incorporate such additional features is difficult to determine *ex ante*. We propose adapting methods from machine learning to search across a wide range of exogeneous variables and capture various forms of nonlinearity: the "GAS tree" combines the parsimonious structure of the GAS model with the flexibility of decision trees (Breiman et al., 1984, 2017), and the "GAS forest," analogous to the random forests of Breiman (2001), averages the forecasts from many GAS trees each produced on a bootstrap sample of the original data. Our GAS tree and GAS forest models can be applied whenever a GAS model is considered, and require from the researcher only a set of exogenous variables that are thought to be possibly useful.

We apply the proposed GAS tree and GAS forest models in four diverse applications: forecasting stock return volatility, the distribution of stock returns, the joint distribution of stock and bond returns, and high-frequency trade durations. We find that the proposed extensions lead to significantly improved forecasts in all four applications. We moreover uncover economic explanations for the *sources* of these forecast gains. Through inspections of the optimal GAS tree structures, and variable importance and parameter sensitivity analyses for GAS forest forecasts, we find that the best-performing GAS tree and forest models are those that incorporate well-known empirical regularities, such as the leverage effect in volatility, the flight-to-quality effect in stock-bond correlations, and the volume-volatility relationship in trade durations.

Faced with evolving data generating processes and the resulting “small” data sets, the success of machine learning methods in economics and finance relies on good, parsimonious benchmark models as reference points, an observation made nicely in Israel et al. (2020). We used GAS models for this purpose; future work may consider augmenting a different class of forecasting models with machine learning methods.

A Derivation of the score function for the t-GAS copula

In this section we present the score function for the t-copula analysis discussed in Section 3. We refer to Creal et al. (2013) for the details of the univariate applications (the GARCH and t-GAS models).

A.1 Notation

We adopt the notation of Creal et al. (2011) for ease of comparability with that article. The Kronecker product is denoted by $A \otimes B$ for any matrices A and B. $A_{\otimes}$ stands for $A \otimes A$. The function $vec(A)$ vectorizes matrix A into a column vector, and $vech(A)$ vectorizes just the lower triangle of A, which eliminates duplicates in the case that A is symmetric. The duplication matrix is implicitly defined as the solution to $\mathcal{D}\,\mathrm{vech}(A) = \mathrm{vec}(A)$. Finally, $\mathbb{E}_{t-1}$ denotes the expectation conditional on the information available up to period $t-1$.

A.2 The probability density function of t copula

We adopt Student's t copula specification in our empirical analysis and its probability density function is given by

$$c(\mathbf{u}_t;\Sigma_t,\nu) = \frac{\Gamma\left(\frac{\nu+2}{2}\right)\Gamma\left(\frac{\nu}{2}\right)}{\sqrt{|\Sigma_t|}\left[\Gamma\left(\frac{\nu+1}{2}\right)\right]^2}\left(1+\frac{\mathbf{x}_t'\Sigma_t^{-1}\mathbf{x}_t}{\nu}\right)^{-\frac{\nu+2}{2}}\prod_{i=1}^{2}\left(1+\frac{x_{i,t}^2}{\nu}\right)^{\frac{\nu+1}{2}} \tag{12}$$

where $\mathbf{x}_t = [x_{1,t}, x_{2,t}] = [T_\nu^{-1}(u_{1,t}), T_\nu^{-1}(u_{2,t})]'$ obtained by applying the inverse of the univariate t distribution with ν degrees of freedom, $\Gamma(\cdot)$ is gamma function and Σ_t is 2-by-2 correlation matrix. We denote the off-diagonal element of Σ_t with ρ_t which is the variable of interest:

$$\Sigma_t = \begin{bmatrix} 1 & \rho_t \\ \rho_t & 1 \end{bmatrix} \tag{13}$$

A.3 The score and information matrix

We use inverse information matrix of the score function as a scaling factor in all applications. Given the complex structure of the Student's t copula, derivation of the information matrix requires tedious calculations, but Creal et al. (2011) provide a closed-form formula of both score and information matrix. Based on their results, we can write

$$\begin{aligned}
\nabla_t &= \tfrac{\partial \log c_t(y_t|\Sigma_t;\nu)}{\partial f_t} \\
&= \tfrac{1}{2}(\mathcal{D}\Psi_t)'\Sigma_{t\otimes}^{-1}\left[w_t \mathbf{x}_{t\otimes} - \operatorname{vec}(\Sigma_t)\right] \\
\mathcal{I}_{t|t-1} &= \mathbb{E}_{t-1}\left[\nabla_t \nabla_t'\right] \\
&= \tfrac{1}{4}(\mathcal{D}\Psi_t)' J_{t\otimes}' \left[gG - \operatorname{vec}(\mathrm{I})\operatorname{vec}(\mathrm{I})'\right] J_{t\otimes}\mathcal{D}\Psi_t
\end{aligned} \tag{14}$$

where $\Psi_t \equiv \frac{\partial \operatorname{vech}(\Sigma_\mathrm{t})}{\partial \rho_t}$, J_t is such that $\Sigma_t^{-1} = J_t' J_t$, $w_t \equiv \frac{\nu+2}{\nu-2+\mathbf{x}_t'\Sigma_t^{-1}\mathbf{x}_t}$, $g \equiv \frac{v+2}{v+4}$, and the explicit form of matrix G is

$$G = \begin{bmatrix} 3 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}. \tag{15}$$

We define the scaled score functions as $s_t = \mathcal{I}_{t|t-1}^{-1}\nabla_t$. We apply an additional transformation to ρ_t which ensures that it lies between -1 and 1. Specifically, we assume that $\rho_t = \frac{1-\exp(-\tilde{\rho}_t)}{1+\exp(-\tilde{\rho}_t)}$. In order to obtain scaled score function for transformed $\tilde{\rho}_t$, we multiply the original scaled score with the derivative of transformation function: $\tilde{s}_t = \frac{\partial \tilde{\rho}_t}{\partial \rho_t} s_t$. When we use the explicit form of each component in equation (14), we obtain the following formula of the scaled score function t copula:

$$s_t = \left(\frac{2}{1-\rho_t^2}\right)\left(\frac{1+\rho_t^2}{g+(2g-1)\rho_t^2}\right)\left(w_t(x_{1,t}x_{2,t}-\rho_t) - \frac{\rho_t}{1+\rho_t^2}(w_t x_{1,t}^2 + w_t x_{2,t}^2 - 2)\right). \tag{16}$$

Note that $(g, w_t) \to (1,1)$ as $\nu \to \infty$, we also obtain the scaled score function for Gaussian copula:

$$s_t = \left(\frac{2}{1-\rho_t^2}\right)\left(x_{1,t}x_{2,t} - \rho_t - \frac{\rho_t}{1+\rho_t^2}(x_{1,t}^2 + x_{2,t}^2 - 2)\right). \tag{17}$$

References

Amihud, Y. (2002). Illiquidity and stock returns: cross-section and time-series effects. *Journal of financial markets*, 5(1):31–56.

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. and Imbens, G. W. (2019). Machine learning methods that economists should know about. *Annual Review of Economics*, 11:685–725.

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 (Statistical Methodology)*, 63(4):727–744.

Baele, L., Bekaert, G., and Inghelbrecht, K. (2010). The determinants of stock and bond return comovements. *The Review of Financial Studies*, 23(6):2374–2428.

Baker, S. R., Bloom, N., and Davis, S. J. (2016). Measuring economic policy uncertainty. *The quarterly journal of economics*, 131(4):1593–1636.

Bauwens, L. and Hautsch, N. (2009). Modelling financial high frequency data using point processes. In Andersen, T. G., Davis, R. A., Kreiss, J.-P., and Mikosch, T., editors, *Handbook of Financial Econometrics*. Springer.

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

Black, F. (1976). Studies of stock price volatility changes. *Proceedings of the Business and Economics Section of the American Statistical Association*, page 177–181.

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

Bollerslev, T., Tauchen, G., and Zhou, H. (2009). Expected stock returns and variance risk premia. *Review of Financial Studies*, 22(11):4463–4492.

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.

Carr, P. and Wu, L. (2008). Variance Risk Premiums. *Review of Financial Studies*, 22(3):1311–1341.

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

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196.

Coulombe, P. G., Leroux, M., Stevanovic, D., and Surprenant, S. (2020). How is machine learning useful for macroeconomic forecasting? *arXiv preprint arXiv:2008.12477*.

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.

Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263.

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

Engle, R. F. and Russell, J. R. (1998). Autoregressive conditional duration: a new model for irregularly spaced transaction data. *Econometrica*, pages 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.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:358–378.

Goulet Coulombe, P. (2020). The macroeconomy as a random forest. *SSRN working paper 3633110.*

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

Guidolin, M. and Timmermann, A. (2006). An econometric model of nonlinear dynamics in the joint distribution of stock and bond returns. *Journal of Applied Econometrics*, 21(1):1–22.

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.*

Israel, R., Kelly, B., and Moskowitz, T. (2020). Can machines "learn" finance? *Journal of Investment Management*, 18(2):23–36.

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.

Karpoff, J. M. (1987). The relation between price changes and trading volume: A survey. *Journal of Financial and Quantitative Analysis*, 22:109–126.

Liu, Y. and Wu, J. C. (2021). Reconstructing the yield curve. *Journal of Financial Economics*, 142(3):1395–1425.

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.

Newey, W. K. and West, K. D. (1987). A simple, positive definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55:703–708.

Nguyen, T.-N., Tran, M.-N., Gunawan, D., and Kohn, R. (2022a). A statistical recurrent stochastic volatility model for stock markets. *Journal of Business & Economic Statistics*, forthcoming.

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

Oh, D. H. and Patton, A. J. (2021). Better the devil you know: Improved forecasts from imperfect models. *Finance and Economics Discussion Series*, working paper 2021-071.

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1):246–256.

Patton, A. J. (2013). Copula methods for forecasting multivariate time series. In Elliott, G. and Timmermann, A., editors, *Handbook of Economic Forecasting, Volume 2*. Elsevier, Oxford.

Patton, A. J. and Sheppard, K. (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics and 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.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2):3–28.