

# Supplementary Online Appendix for “A mixture autoregressive model based on Student’s $t$ -distribution” by Meitz, Preve, and Saikkonen (not meant for publication)

The supplementary online appendix includes information on the numerical optimization methods employed for maximum likelihood estimation, simulation experiments, and further details of the empirical example.

## 1 Estimation

### 1.1 Numerical optimization

Finding maximum likelihood estimates of the unknown parameters of an StMAR( $p, M$ ) model amounts to maximizing  $L_T^{(c)}(\boldsymbol{\theta})$ , a function in  $M(p+4) - 1$  variables, under several constraints. Our experience with both actual and simulated data indicates that this can be challenging, in part due to multiple local maxima, and that advanced numerical optimization methods are needed. We use a hybrid numerical optimization scheme combining randomized search methods and classical gradient based methods to efficiently search for a global maximum that satisfies the constraints. (Using the commonly employed EM algorithm is also a possibility; however, contrary to some previous mixture models, the mixing weights of the StMAR model depend on the autoregressive parameters implying that the optimization problem in the maximization step does not simplify much.)

We first employ a genetic algorithm, a randomized search method designed to search for a global optima, using a variety of initial populations (collections of starting points; for discussions on the genetic algorithm, other popular algorithms, and their applications in econometrics, see Goffe et al., 1994, and Dorsey and Mayer, 1995). For each of the initial populations, the genetic algorithm is run for a small number of generations to reach the region near an optimum point relatively quickly. Corresponding to each initial population, the solution found by the genetic algorithm is then used as a starting point for MATLAB’s optimization method `fmincon`, which is faster and more efficient for local search (for `fmincon` we further use a sequential quadratic programming method; see e.g. Nocedal and Wright, 2006). The final parameter estimate is the best solution found by `fmincon` for all the starting points considered. This hybrid optimization scheme combining multiple initial populations, the genetic algorithm, and `fmincon` allows us to efficiently search the parameter space and reduces the risk of ending up with a local, not global, maximum. We parallelize our code to consider multiple initial populations and starting points in parallel. This helps to speed up the optimization considerably. In view of the complexity of the estimation procedure, numerical gradients and Hessians are used for the optimization.

The StMAR code used in our S&P 500 realized kernel example, further described in our StMAR MATLAB Toolbox Documentation, is available for download through the second authors webpage at [https://www.researchgate.net/profile/Daniel\\_Preve](https://www.researchgate.net/profile/Daniel_Preve). R code by Savi Virolainen is available through the CRAN repository in the form of the ‘uGMAR’ package.

## 1.2 Simulation experiments

We carried out several Monte Carlo studies to evaluate the performance of the numerical optimization scheme described above. The results of two of these studies are reported in Tables 1 and 2. In these experiments, 500 independent simulated sample paths were generated from an StMAR(1,2), and also from an StMAR(4,2), process; the sample sizes and parameter values used are displayed in Tables 1 and 2.

Overall, the performance of the numerical optimization scheme is quite satisfactory. As is commonly known, the degrees of freedom parameter of a Student's  $t$ -distribution is relatively difficult to estimate, especially if its true value is large. This is also the case for our StMAR model, and our simulation results indicate that the  $\nu_m$  parameters can be relatively difficult to estimate even in moderate or large samples. Similar difficulties were reported by Wong et al. (2009) when estimating their (constant mixing weights) version of a Student  $t$ -mixture autoregressive model using the EM algorithm (see their Table 3).

Table 1: Simulation results for a StMAR(1,2) with various sample sizes  $T$  and 500 replications. M, Md and SD denote the sample mean, median, and standard deviation, respectively.

		$T = 500$			$T = 1000$			$T = 2500$			$T = 5000$		
	Value	M	Md	SD	M	Md	SD	M	Md	SD	M	Md	SD
$\varphi_{1,0}$	-1.50	-2.31	-1.75	2.25	-2.02	-1.58	1.55	-1.64	-1.51	0.65	-1.52	-1.51	0.22
$\varphi_{1,1}$	0.85	0.73	0.83	0.20	0.78	0.84	0.16	0.83	0.85	0.08	0.85	0.85	0.03
$\sigma_1^2$	0.35	0.38	0.32	0.45	0.35	0.33	0.09	0.35	0.34	0.05	0.35	0.35	0.04
$\nu_1$	4.00	14.01	5.29	44.43	6.45	4.59	11.47	4.47	4.16	2.22	4.12	4.03	0.47
$\varphi_{2,0}$	-5.50	-4.79	-5.33	2.92	-5.10	-5.43	1.65	-5.41	-5.48	0.84	-5.50	-5.49	0.39
$\varphi_{2,1}$	0.35	0.44	0.37	0.30	0.40	0.36	0.21	0.36	0.35	0.10	0.35	0.35	0.05
$\sigma_2^2$	0.30	0.96	0.30	14.03	0.31	0.30	0.06	0.30	0.30	0.03	0.30	0.30	0.02
$\nu_2$	8.00	20.48	7.08	53.76	15.15	7.44	36.00	9.19	7.76	6.64	8.30	7.80	2.12
$\alpha_1$	0.60	0.61	0.59	0.10	0.59	0.59	0.07	0.59	0.59	0.04	0.60	0.60	0.03

Table 2: Simulation results for a StMAR(4,2) with various sample sizes  $T$  and 500 replications. M, Md and SD denote the sample mean, median, and standard deviation, respectively.

		$T = 1000$			$T = 2500$			$T = 5000$			$T = 10000$		
	Value	M	Md	SD	M	Md	SD	M	Md	SD	M	Md	SD
$\varphi_{1,0}$	-1.00	-1.65	-1.16	1.03	-1.46	-1.08	0.83	-1.27	-1.03	0.67	-1.15	-1.02	0.51
$\varphi_{1,1}$	0.35	0.34	0.34	0.06	0.34	0.34	0.06	0.34	0.35	0.03	0.35	0.35	0.02
$\varphi_{1,2}$	0.20	0.16	0.17	0.07	0.18	0.19	0.05	0.19	0.20	0.04	0.19	0.20	0.03
$\varphi_{1,3}$	0.15	0.12	0.13	0.07	0.13	0.14	0.06	0.14	0.15	0.04	0.14	0.15	0.03
$\varphi_{1,4}$	0.10	0.08	0.08	0.06	0.09	0.09	0.05	0.09	0.09	0.03	0.10	0.10	0.02
$\sigma_1^2$	0.25	1.84	0.26	17.82	0.46	0.25	3.77	0.26	0.25	0.04	0.25	0.25	0.02
$\nu_1$	9.00	9.06	7.02	16.71	8.14	8.32	3.64	8.50	8.87	2.62	8.70	8.88	1.91
$\varphi_{2,0}$	-3.00	-2.95	-2.92	2.86	-2.68	-2.97	0.92	-2.80	-3.00	0.72	-2.88	-2.98	0.53
$\varphi_{2,1}$	0.30	0.29	0.30	0.14	0.30	0.30	0.04	0.30	0.30	0.03	0.30	0.30	0.02
$\varphi_{2,2}$	0.10	0.11	0.12	0.13	0.12	0.11	0.06	0.11	0.10	0.04	0.11	0.10	0.03
$\varphi_{2,3}$	0.05	0.06	0.07	0.13	0.07	0.06	0.06	0.06	0.05	0.04	0.06	0.05	0.03
$\varphi_{2,4}$	0.05	0.04	0.04	0.14	0.05	0.05	0.05	0.05	0.05	0.03	0.05	0.05	0.02
$\sigma_2^2$	0.30	1.32	0.23	12.28	0.35	0.26	0.42	0.32	0.28	0.17	0.30	0.29	0.07
$\nu_2$	3.00	23.26	4.63	69.40	4.96	3.38	3.68	3.98	3.11	2.47	3.48	3.04	1.66
$\alpha_1$	0.55	0.62	0.59	0.10	0.57	0.56	0.05	0.56	0.56	0.04	0.55	0.55	0.03

## 2 Empirical example

### 2.1 Two-sided prediction intervals

Table 2 of the main paper reported the percentage shares of 1, 5, 10, and 22-day cumulative  $RK_t$  out-of-sample observations that belong to the 99%, 95%, and 90% one-sided upper prediction intervals based on the distribution of the simulated sample paths. The corresponding numbers for two-sided prediction intervals (for nominal levels 99%, 95%, 90%, 70%, and 50%) are presented in Table 3. Overall, it is seen that the empirical coverage rates of the StMAR based prediction intervals are closer to the nominal levels than the ones obtained with the reference models. By comparison, the accuracy of the prediction intervals obtained with the HAR quickly degrade as the forecast period increases.

Note that to generate prediction intervals for the reference AR and HAR models, we need to specify an error distribution in these models; we assume that the errors are Gaussian. The order of the AR model is chosen using AIC and BIC; both favour an  $AR(p)$  model with  $p = 11$ .

### 2.2 Volatility point forecast evaluation criteria

Let  $RM$  denote a (cumulative) realized measure (volatility proxy), such as the realized variance or realized kernel, and  $\widehat{RM}$  a forecast of  $RM$ . Although realized measures generally are consistent estimators of the underlying latent volatility, in practice they are noisy proxies. Because of this, care needs to be taken when choosing a loss function to evaluate and compare volatility forecasts. Following the literature on volatility forecast comparison (Patton and Sheppard, 2009; Patton, 2011), we consider the two most widely used loss functions, namely squared loss (MSE)

$$L_{MSE}(RM, \widehat{RM}) = (RM - \widehat{RM})^2$$

Table 3: The percentage shares of cumulative realized kernel observations that belong to the 99%, 95%, 90%, 70% and 50% two-sided prediction intervals based on the distribution of 1,000,000 simulated conditional sample paths.

	<i>Daily</i>					<i>Weekly</i>				
	99%	95%	90%	70%	50%	99%	95%	90%	70%	50%
AR(11)	98.59	94.56	89.92	65.73	45.97	95.12	88.01	79.47	61.18	42.48
HAR	98.59	95.16	89.72	65.93	44.96	94.31	86.79	79.67	60.37	41.87
StMAR(4,1)	98.79	94.35	88.91	64.92	45.56	96.34	91.26	84.55	64.63	45.73
StMAR(4,2)	98.79	94.56	89.52	66.73	48.59	96.34	89.63	83.13	63.82	45.53
StMAR(4,3)	98.79	94.56	89.92	66.94	47.78	96.34	89.63	83.13	63.41	45.12

	<i>Biweekly</i>					<i>Monthly</i>				
	99%	95%	90%	70%	50%	99%	95%	90%	70%	50%
AR(11)	93.22	83.98	77.82	60.99	43.94	93.47	84.00	78.11	59.79	41.05
HAR	93.02	83.98	77.41	58.52	41.07	89.89	80.84	76.42	56.63	38.11
StMAR(4,1)	96.92	89.94	84.39	65.91	48.46	99.16	90.32	86.74	66.11	45.05
StMAR(4,2)	96.71	88.50	81.52	62.22	44.15	97.47	86.32	82.32	65.26	43.16
StMAR(4,3)	96.71	88.50	81.52	61.60	44.15	97.47	86.53	82.11	64.63	42.74

and QLIKE (quasi-likelihood) loss

$$L_{QLIKE}(RM, \widehat{RM}) = \frac{RM}{\widehat{RM}} - \log \frac{RM}{\widehat{RM}} - 1.$$

Moreover, as Patton and Sheppard (2009) recommend the use of QLIKE rather than MSE in volatility forecasting applications, we employ QLIKE loss as our primary loss function, and squared loss as our secondary loss function.

### 2.3 Point forecasts

Results for 1, 5, 10, and 22-day cumulative  $RK_t$  forecasts based on the sample median are presented in Figure 1. The left panel reports QLIKEs and the right panel MSEs. Forecast accuracy of the models is reported relative to the StMAR(4,2) model: The horizontal line (at 100) represents the StMAR(4,2) model, whereas the other lines represent the size of the forecast error measure made relative to this model (for instance, a value of 110 in the left panel is to be interpreted as a QLIKE 10% larger than for the StMAR(4,2) model). The overall performance of the StMAR(4,2) model is quite reasonable; the model does particularly well in terms of our primary loss function, QLIKE. Overall, the StMAR(4,3) model performs almost equally well. Figure 1 also suggests that the more parsimonious StMAR(4,1) model may be preferred to the StMAR(4,2) model over longer (biweekly, monthly) forecast periods. The popular HAR model performs well under MSE, but considerably less so under QLIKE.

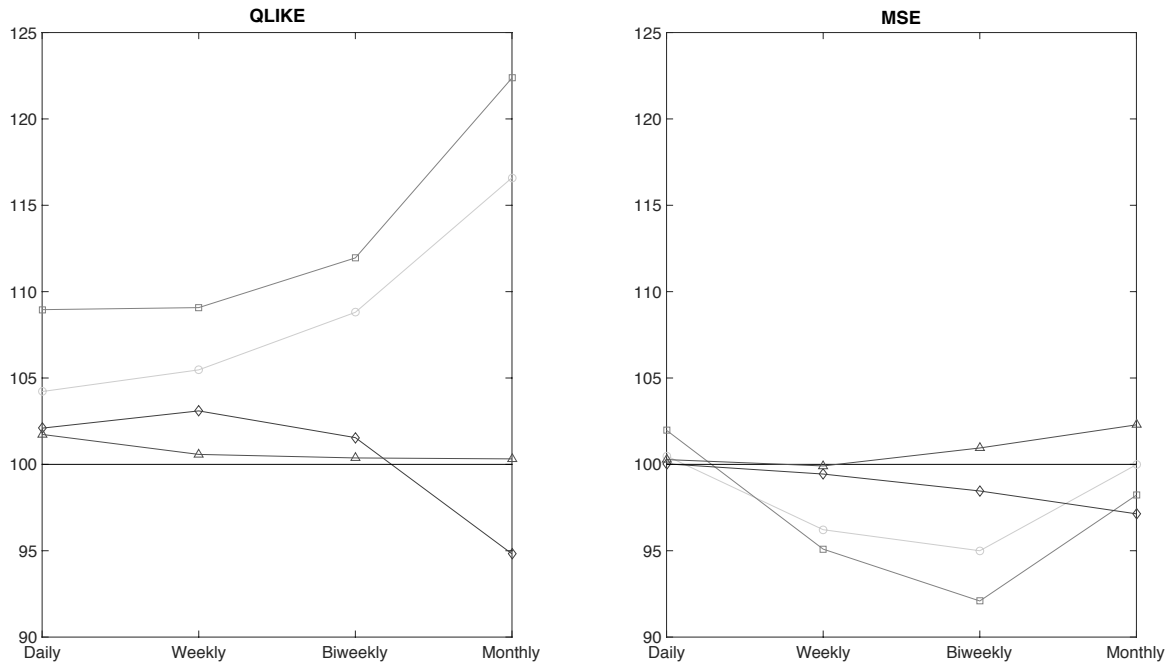


Figure 1: Relative forecast accuracies for the S&P 500 RK data in terms of QLIKE (left) and MSE (right). Results for the AR(11) (circle), HAR (square), StMAR(4,1) (diamond), StMAR(4,2) (solid), and StMAR(4,3) (triangle) models.

## References

- Dorsey, R.E., and W.J. Mayer (1995) Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. *Journal of Business & Economic Statistics* **13**(1): 53–66.
- Goffe, W.L., G.D. Ferrier, and J. Rogers (1994) Global optimization of statistical functions with simulated annealing. *Journal of Econometrics* **60**(1–2): 65–99.
- Kalliovirta, L., M. Meitz, and P. Saikkonen (2015) A Gaussian mixture autoregressive model for univariate time series. *Journal of Time Series Analysis* **36**: 247–266.
- Kalliovirta, L., M. Meitz, and P. Saikkonen (2016) Gaussian mixture vector autoregression. *Journal of Econometrics* **192**: 485–498.
- Nocedal, J., and S.J. Wright (2006) *Numerical Optimization (2nd ed.)*. Springer, New York.
- Patton, A.J. (2011) Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics* **160**(1): 246–256.
- Patton, A.J., and K. Sheppard (2009) Evaluating volatility and correlation forecasts, in T.G. Andersen, R.A. Davis, J.P. Kreiß and T. Mikosch (Eds.), *Handbook of Financial Time Series*. Springer, Berlin Heidelberg.
- Wong, C.S., W.S. Chan, and P.L. Kam (2009) A Student  $t$ -mixture autoregressive model with applications to heavy-tailed financial data. *Biometrika* **96**(3): 751–760.