GMAM 2.0 : Reduced Variance Estimator

iCapital Portfolio Analytics

March 8, 2023

I. Motivation and Purpose

The objective of the new GMAM (Global Multi-Asset Model) version 2.0 is to improve upon the current (GMAM1.0) methodology employed in the factor analysis of client portfolios.

A short summary of the methodology employed in GMAM 1.0 provides context to the GMAM 2.0 enhancements. As with GMAM 2.0, the core of GMAM 1.0 is a multi-factor model of returns. Both perform analysis at the asset-level and aggregate to the portfolio-level. Unlike GMAM 2.0 however, GMAM 1.0 uses a single-step LASSO (Least Absolute Shrinkage and Selection Operator) regression to estimate each asset's exposure. The shrinkage hyper-parameter (λ) is estimated using a heuristic based on the length of asset's history.

The above methodology has the following major shortcomings -

1. The shrinkage hyper-parameter does not seem aligned with the risk analysis objectives of the model. It is only a functional heuristic.
2. The single-step LASSO provides a biased set of estimates of exposures.

The key improvements deployed through GMAM2.0 are:

1. A two-step procedure for estimating the exposures of assets using LASSO for the variable selection step and an Ordinary Least Squares (OLS) regression for the parameter estimation step. This allows for an unbiased estimate of the coefficients conditional on the variable selection.
2. A Cross-Validation (CV) procedure to estimate the LASSO parameter in the first step above. The explicit objective of this CV-based LASSO parameter is to optimize the out-of-sample efficacy of the model.

We will dive deeper in to these specific enhancements in the next section. A brief overview of necessary prerequisites is provided where necessary.

II. Methodology

A. The Two-stage Estimation Process

The two-step approach employed in GMAM2.0 uses a LASSO regression to select from a set of regressors. LASSO is a regression analysis method with regularization capability that helps enhance prediction accuracy and interpretability. GMAM 2.0 reduces over-fitting by using LASSO to penalize the number of regressors in relation to the estimation error. To put in mathematical terms, examine the linear regression's matrix formulation:

$$\begin{aligned}\mathbf{y} &= \boldsymbol{\alpha} + \tilde{X}\tilde{\boldsymbol{\beta}} + \boldsymbol{\varepsilon} \\ &= X\boldsymbol{\beta} + \boldsymbol{\varepsilon} \end{aligned} \tag{1}$$

where $\mathbf{y}$ and $\tilde{\mathbf{X}}$ are given, we look to estimate the intercept $\boldsymbol{\alpha}$, the exposures $\tilde{\boldsymbol{\beta}}$ and $\boldsymbol{\varepsilon}$, the error term. Without loss of generality, the intercept may be absorbed in the exposures by including a vector of ones in the covariate matrix $\tilde{\mathbf{X}}$. E.g., $\mathbf{X} = [\mathbf{1}; \tilde{\mathbf{X}}]$ and $\beta = [\alpha, \tilde{\beta}]$. Lasso arrives at an estimate of vector $\boldsymbol{\beta}$ by both minimizing the estimation error and penalizing models with a larger absolute sum of regressor coefficients. The objective function is as follows:

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=2}^{p} |\beta_j|. \tag{2}$$

where $\lambda \geq 0$ is the penalty term. λ controls the amount of shrinkage in L1 term of Equation 2. When $\lambda = 0$, no parameters are eliminated and the setup becomes a familiar case of ordinary least squares (OLS). As λ increases, the coefficients shrink towards zero. At larger values of lambda, the regression may set a significant number of the coefficients to zero. This gives Lasso variable selection properties. See Figure 1 where the effect of increasing λ on the model fit's bias-variance trade-off is illustrated.

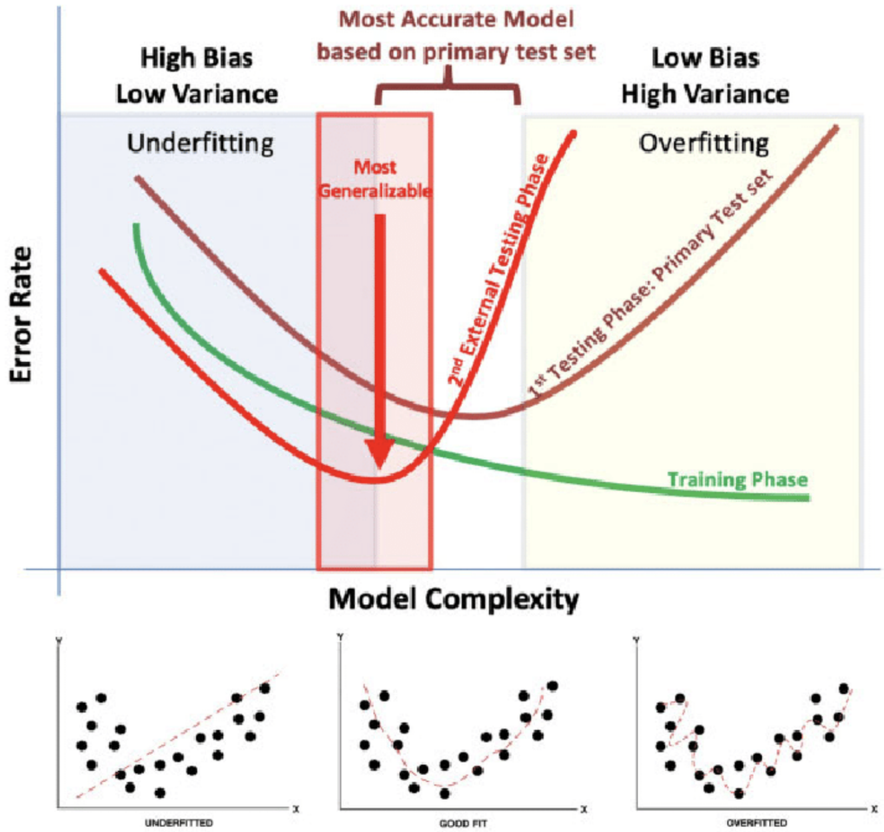

Figure 1 https://www.researchgate.net/figure/Bias-variance-trade-off-in-machine-learning-This-figure-illustrates-the-trade-off-fig2-335604816

In rare cases, the Lasso procedure selects a constant model. In such cases where no regressors are selected we subject the data to a second approach to select regressors. In such cases, the number of regressors are selected along the LASSO path (via LARS) where the Akaike Information Criterion (AIC) corrected for small sample bias (AICC) is minimized. The AICC metric is defined as -

$$AICC = AIC + \frac{2k(k+1)}{T-k-1} \tag{3}$$

where k is the number of selected regressors.

The non-zero regressors from above feed the RHS of a second-stage OLS regression. More formally, let $\mathbf{X}_\lambda \subseteq \mathbf{X}$ represent the subset of regressors that are identified with λ as the shrinkage hyper-parameter. The two-step OLS error, $\boldsymbol{\varepsilon}_{2ols}$, is:

$$\boldsymbol{\varepsilon}_{2ols} = \mathbf{y} - \mathbf{X}_\lambda \boldsymbol{\beta}. \tag{4}$$

The estimates for the coefficients $\boldsymbol{\beta}$ originate from the standard OLS estimation equation:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}_\lambda^\top \mathbf{X}_\lambda\right)^{-1} \mathbf{X}_\lambda^\top \mathbf{y}. \tag{5}$$

B. The Optimal Lambda Estimation

The objective is to find the Lasso λ for a given asset A such that it minimizes the 2-step OLS regression error, using cross validation. Cross Validation (CV) is a method of re-sampling data to conduct training and testing multiple times on a fixed set of data. This allows for getting a better estimate of model performance. Cross validation is also a common practice in hyper-parameter tuning. There is a variety of specifications for CV, each serving a different need while following a uniform set of principles. Expanding and sliding window variations of cross validation address the issue of look-ahead-bias when the temporal dependence in data is meaningful as is the case in our application.

B.1. Expanding Window Cross Validation

With the same objective of re-sampling data, modify the standard K–Folds Cross Validation setup such that it respects the temporal nature of data. In other words, limit the folds to only those in which all test data occur after the last training data point. See Figure 2

Begin with the following three parameters:

1. Minimum training window size (*min_train*)
2. Test window size (*test_size*)
3. Increment to training window (*inc_size*)

In the first fold, train the model on the first *min_train* data points and test on the next non-overlapping *test_size* data points. Next define the new fold before repeating the train and test steps:

- Training set: increase the previous training window of the set by *inc_size*.
- Test set: choose *test_size* data points starting after the last training data point.

B.2. Sliding Window Cross Validation

Temporal data decay over time. In sliding window cross validation, only most recent data is used for training. See Figure 2.

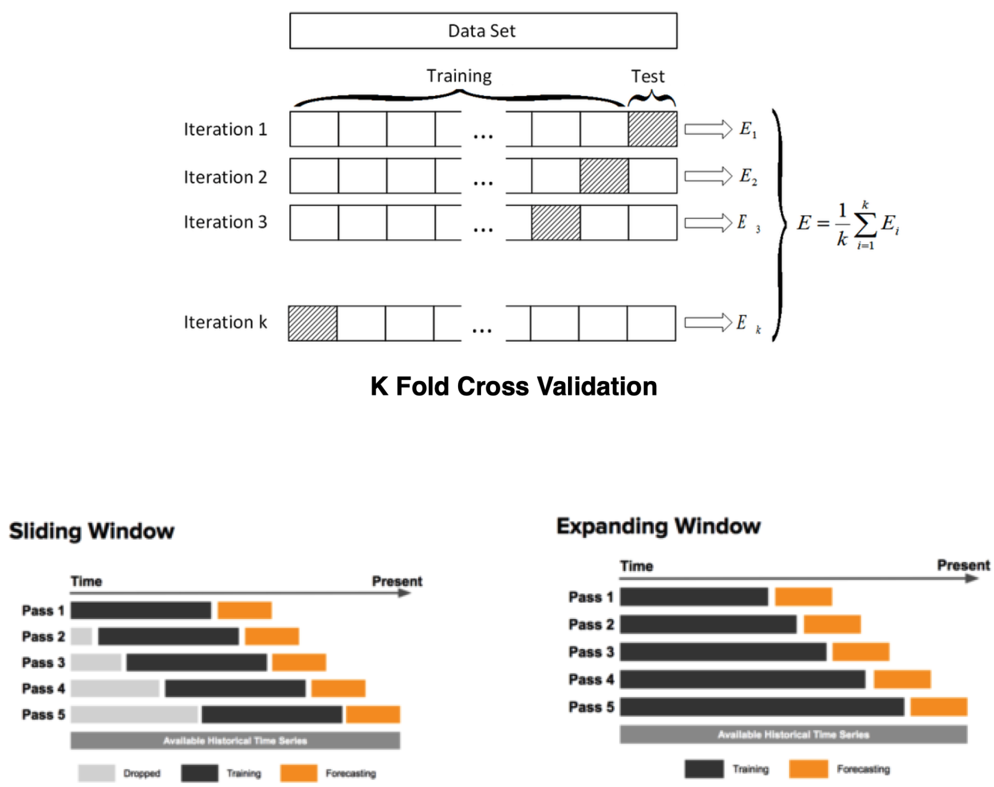

Figure 2 Cross Validation Variants

Set up the first fold as with the Expanding Window method. The next fold is defined as:

- Training set: expand the training set by *inc_size* and trim out *inc_size* of the oldest training data points.
- Test set: choose *test_size* data points starting after the last training data point.

C. The Algorithm

With the above understanding of our two-step analysis, and an expanding or sliding window CV, the following procedure is implemented in our application -

1. Pre-process factor returns, i.e. the covariate data.
   - RHS, which is assumed to include the effect of an intercept is standardized via `sklearn.preprocessing.StandardScaler`.
     $x = X\Sigma_X^{-1}$
     where Σ_X is a diagonal matrix with the standard deviations of the columns of X
   - LHS is left unchanged.
2. For each asset A in a given portfolio, setup an expanding window cross validation for lifetime analysis or an expanding, then sliding window cross validation for the rolling window analysis, F_A.
   - The initial training window size is $\frac{T_A}{2}$ if performing lifetime analysis, $\frac{W}{2}$ if using window analysis with window size W. In case of window analysis, expand the training window until of size W, then use rolling windows of size W.
   - The validation window size is the greater of 1M[1] or a validation window that allows for ($K =$) 20 approximately equal-sized folds
   - The training window size is adjusted as previously described.
3. For every fold $f_i \in F_A$:
   (a) Let Λ_i be the set of candidate λ's that Lasso via Lars returns on fold f_i using `sklearn.linear_model.lars_path`.
   (b) For every $\lambda_{ij} \in \Lambda_i$ (s.t. $0 \leq \lambda_{i1} \leq \lambda_{i2} \leq ... \leq \lambda_{iM}$):
      - Let $X_{\lambda_{ij}}$ be the subset of regressors identified by λ_{ij}.
      - Calculate $sse_{ij} = ||\varepsilon_{2ols}||^2 = (\mathbf{y}_{test} - \mathbf{X}_{\lambda_{ij}}\boldsymbol{\beta}_{train})^T(\mathbf{y}_{test} - \mathbf{X}_{\lambda_{ij}}\boldsymbol{\beta}_{train})$ where $\boldsymbol{\beta}_{train} = \left(\mathbf{X}_{\lambda_{ij}}^T\mathbf{X}_{\lambda_{ij}}\right)^{-1}\mathbf{X}_{\lambda_{ij}}^T\mathbf{y}_{train}$

[1]Note that because our data is in monthly frequency, 1M represents one data point.

(c) It is important to note at this stage that the SSE of the i^{th} fold is a step function between the λ_{ij}'s. This is because the OLS error as defined above does not change until a covariate is included/excluded at theseλ_{ij}'s.

$$SSE_i(\lambda) = \begin{cases} \infty \text{ if } \lambda < \lambda_{i1} \\ sse_{ij} \text{ if } \lambda_{i,j} \leq \lambda < \lambda_{i,j+1} \text{ where } j \in [1, 2, ..., M-1] \\ sse_{iM} \text{ if } \lambda >= \lambda_{i,M} \end{cases}$$

4. Define $MSE(\lambda) = \frac{\sum_{i=1}^{K} SSE_i(\lambda) N^i_{\text{training data}}}{N_{\text{total testing data}} N_{\text{total training data}}}$ and return $\min\limits_{\lambda \in \bigcup \Lambda_i} MSE(\lambda) = \lambda_{opt}$. Let $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{P_{all}}$ be the rank ordered λ's in $\bigcup \Lambda_i$. Given that the sum of step functions is a step function, it suffices to calculate $MSE(\lambda)$ at the above λ's to find the minima[2].

5. For each asset in the portfolio, calculate the exposures using the above calculated λ_{opt} in the two-step estimator; linearly aggregate to the portfolio-level exposures using given asset weights.

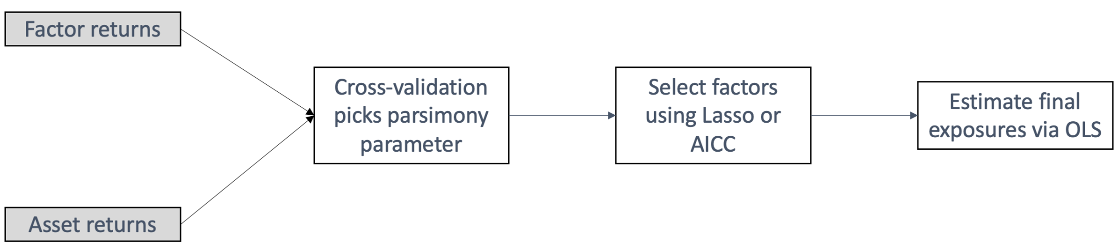

Figure 3 Flowchart

6. Define an asset's factor exposure estimation to be reliable if the exposures are jointly statistically distinct from zero. The significance level used for the F-statistic used therein is defined as -

$$1 - 0.9^{\frac{1}{\min(K, T_A)}} \tag{6}$$

At the portfolio level, in addition to assessing the exposures being distinct from zero, the total weight of unreliable assets should be greater than 50% for the analysis to be

[2]we evaluate at the midpoints between the lambdas in order to avoid numerical instabilities

considered as unreliable.

In the event that an asset in a portfolio is “unreliable”, use a constant model for its exposures.

7. The following metric assesses the out-of-sample performance of GMAM 2.0. The metric is related to the R^2 metric used to assess a regression fit and is therefore called out-of-sample R^2)

$$R^2_{OOS} = 1 - \frac{\sum_{t=1}^{T}(r_t - \hat{r}_t)^2}{\sum_{t=1}^{T}(r_t - \bar{r}_t)^2} \tag{7}$$

$$s.t$$

$$\bar{r}_t = \frac{1}{t-1} \sum_{s \in 1:(t-1)} r_s$$

where r_t is the out-of-sample data and $\hat{r}_t$ is the model predicted values using only the data up through time $t-1$.

It is not feasible to calculate the above metric on-the-fly for the computational cost of estimating $\lambda_{t,opt}$ at each t. Hence we use the lifetime optimal lambda at each t and define the R-squared thus estimated as cross-validated R-squared(R^2_{CV}). We use R^2_{CV} as a proxy for R^2_{OOS}; see Figure 4.

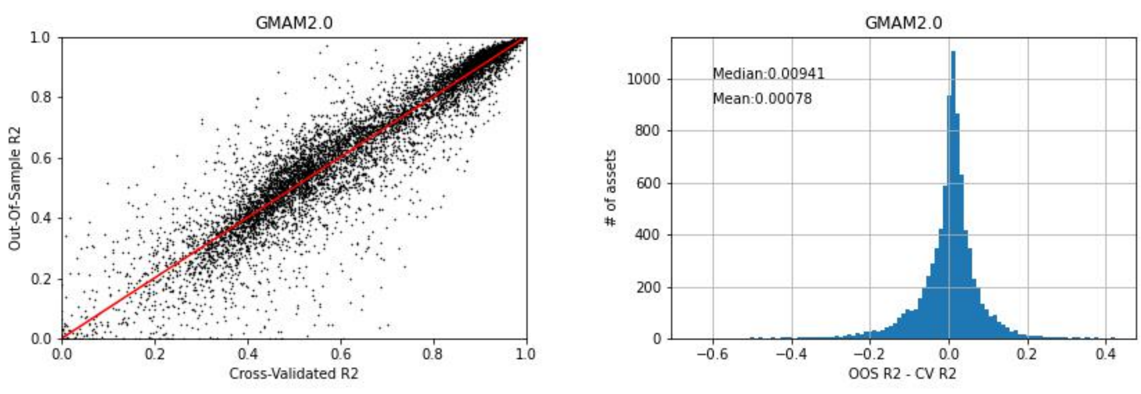

Figure 4 Cross Validated R^2 is highly correlated, and thus a good proxy for, Out-of-sample R^2

III. Glossary

Table I provides a terminology look-up for quick reference.

Term	Description
CV	Cross Validation
T_A	Number of records available for asset A
F_A	Set of cross validation folds for asset A
λ	Penalty parameter in Lasso regression setup
λ_{opt}	Optimal Lambda in the 2-step regression for asset A
Λ_i	Subset of select λ's in fold f_i
SSE_{ij}	sum-of-squared-errors from 2 step estimator associated with $\lambda_j \in \Lambda_i$
LHS	Left Hand Side
RHS	Right Hand Side

Table I Terminology LookUp Table

IV. Empirical results - Comparison to GMAM1.0

Unless otherwise stated, the results shown in this section are for the 8728 assets that were in active client portfolios or featured by iCapital as of 2022-09-12.

A. Estimation vs. Explanation

GMAM2.0 is designed to improve predictive capability over GMAM1.0, at the cost of in-sample fit. In Figure 5 the trade-off between explanatory vs. predictive power is demonstrated. This trade-off is desirable as it brings the in-sample fit better in-line with the out-of-sample fit; see Figure 6

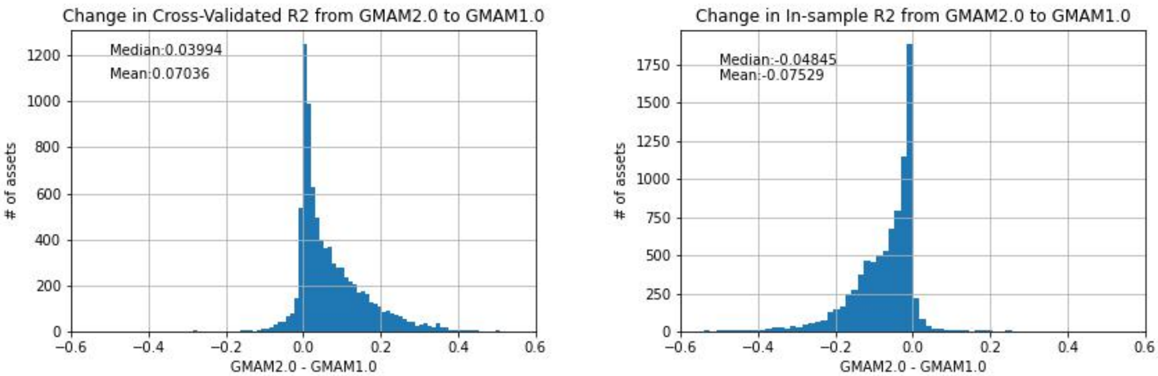

Figure 5 The Cross-Validated R^2 (left panel) improves almost as much as the In-sample R^2 (right panel) drops in the median

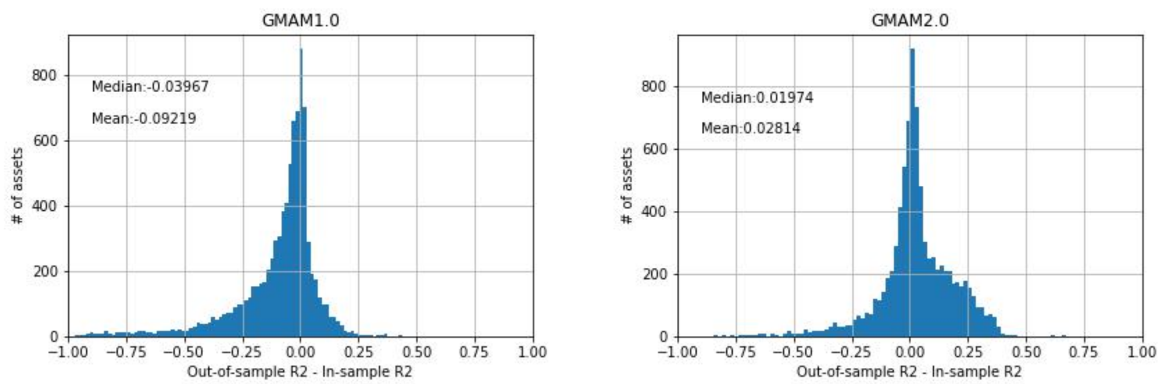

Figure 6 Out-of-sample R^2 is substantially lower than the In-sample R^2 fit for GMAM1.0 (left panel) vis-à-vis GMAM2.0 (right panel)

B. Parsimony

As GMAM2.0 increases the threshold for a factor to be included in the analysis of an asset, in general, the number of factors selected for an asset is lower in GMAM2.0 vs. GMAM1.0; see Figure 7

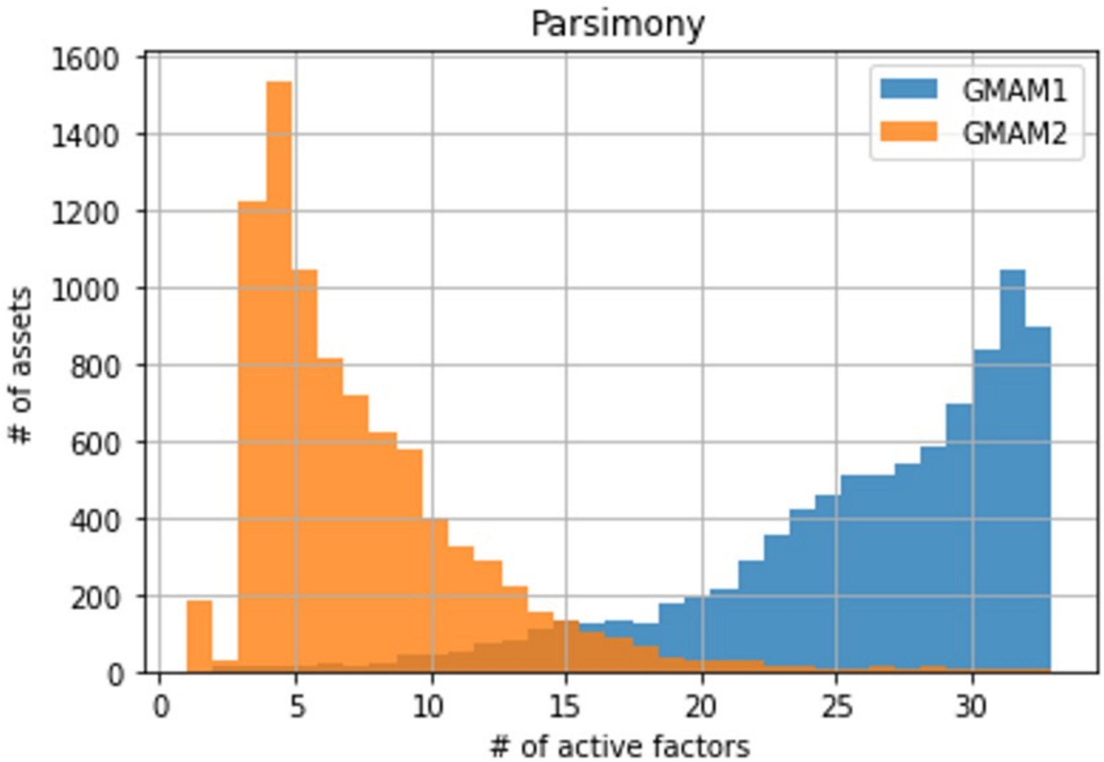

Figure 7 Parsimony

C. Exposure Stability

Shown in Figure 8 the month-over-month change in the L2 norm of exposures of assets for GMAM1.0 and GMAM2.0. As GMAM2.0 is better able to capture the underlying economic dependence of the assets' returns to the risk factors in our models, we clearly observe that the exposures for GMAM2.0 are stabler compared to GMAM2.0.

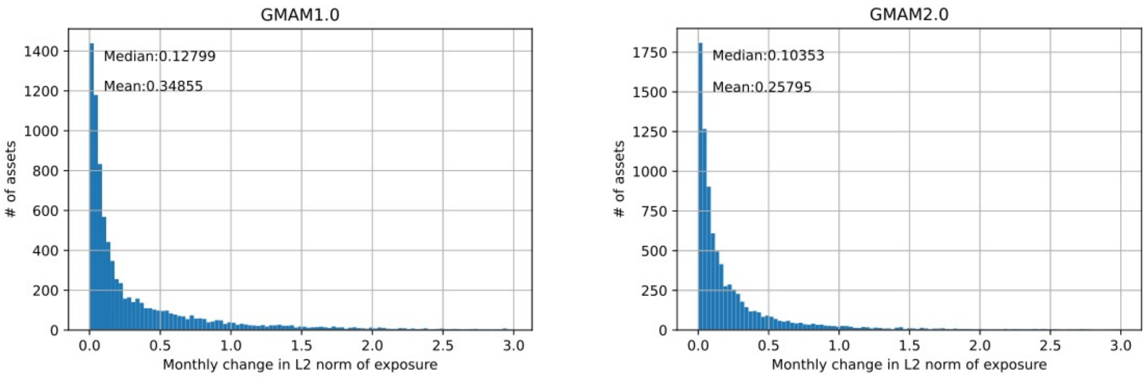

Figure 8 The mean month-over-month change in the L2 norm of exposures is higher for GMAM1.0 (left panel) compared to GMAM2.0 (right panel)