2021

Machine Learning Econometrics

Philippe Goulet Coulombe
University of Pennsylvania

Machine Learning Econometrics

Abstract

Much of econometrics is based on a tight probabilistic approach to empirical modeling that dates back to Haavelmo (1944). This thesis explores a modern algorithmic view, and by doing so, finds solutions to classic problems while developing new avenues.

In the first chapter, Kalman-filter based computations of random walk coefficients are replaced by a closed-form solution only second to least squares in the pantheon of simplicity.

In the second chapter, random walk "drifting" coefficients are themselves dismissed. Rather, evolving coefficients are modeled and forecasted with a powerful machine learning algorithm. Conveniently, this generalization of time-varying parameters provides statistical efficiency and interpretability, which off-the-shelf machine learning algorithms cannot easily offer.

The third chapter is about the to the fundamental problem of detecting at which point a learner stops learning and starts imitating. It answers "why can't Random Forest overfit?" The phenomenon is shown to be a surprising byproduct of randomized "greedy" algorithms – often deployed in the face of computational adversity. Then, the insights are utilized to develop new high-performing non-overfitting algorithms.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Economics

First Advisor

Francis X. Diebold

Second Advisor

Frank Schorfheide

Subject Categories

Economics

MACHINE LEARNING ECONOMETRICS

Philippe Goulet Coulombe

A DISSERTATION

in

Economics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2021

Co-Supervisor of Dissertation

Francis X. Diebold, Professor of Economics

Co-Supervisor of Dissertation

Frank Schorfheide, Professor of Economics

Graduate Group Chairperson

Jesus Fernandez Villaverde, Professor of Economics

Dissertation Committee

Francis X. Diebold, Professor of Economics

Frank Schorfheide, Professor of Economics

Karun Adusumilli, Assistant Professor of Economics

MACHINE LEARNING ECONOMETRICS

Philippe Goulet Coulombe

"If I knew where the good songs came from, I'd go there more often." - L. Cohen

ACKNOWLEDGMENT

"What a long strange trip it's been."

I am indebted to my two supervisors, Frank Diebold and Schorfheide, for their vast knowledge, invaluable guidance, and continuous support throughout my time at Penn. I thank Karun Adusumilli, whose input has been a key ingredient for many of my projects. I am grateful to Dalibor Stevanovic for his friendship and guidance — and especially for bringing me back to machine learning in the early days of this PhD.

I thank the many colleagues (now friends) I have met at Penn. Among them, Edvard Bakhitov, Évelyne Brie, Paul Décaire, Max Esser, Maximilian Göbel, Ricardo Marto, and Boyuan Zhang. I am also thankful to all the members of the Climate Econometrics group. In particular, some special thanks are due to Dave Wigglesworth, Jiwhan Moon, Isaac Tham, Preston Ching, Jipeng Liu, Akshay Malhotra, and Liza Brover who have helped me in one way or another on several projects. I also thank collaborators outside of Penn who received many phone calls at dereasonnable hours. Among them, Hugo Couture, Stéphane Suprenant, and Maxime Leroux.

At a personal level, I want to concisely thank people that made my time in Philly more than a mere academic excursion: Sandra, Camille, the French crew, and the Key Sushi crew. Special thanks to Rob, my partner in rock'n'roll, and to Jeff and the Bob and Barbara's community. I am grateful to Sophie, who, so long ago now, opened my eyes to the world and without whose support, I would not have made it to (or through) Penn. Finalement, je tiens à remercier ma famille pour leur support constant durant l'entierté de ma scolarité – de l'aide aux devoirs au primaire jusqu'à la recherche universitaire.

A last word. I dedicate this work to my four grandparents, Armande, Jeannette, Laurent, Patrice, whose life story never ceased to amaze me. It reminds me that, while working on *ideas* is a necessity, it is also a luxury.

ABSTRACT

MACHINE LEARNING ECONOMETRICS

Philippe Goulet Coulombe

Francis X. Diebold and Frank Schorfheide

Much of econometrics is based on a tight probabilistic approach to empirical modeling that dates back to Haavelmo (1944). Yet, the landscape of quantitative economics research has been changing rapidly in the last decade. Large data sets are increasingly available, and so is computational power. Both permits and require innovation in how economists treat data. In statistics and computer science, methods falling under the umbrella of "Machine Learning" (ML) have become common use in academia and industry, thanks to their ability to solve modern empirical problems – especially that of prediction. This thesis is part of a research agenda that leverages, adapts, and develops ML tools for economic data analysis. More than simply offering a basket of new methods, this strand of research also embeds a change in philosophy, largely borrowed from ML, where models design should be as data-driven as possible and their evaluation, more empirical than theoretical. Along these lines, this thesis explores a modern algorithmic view to macroeconometric modeling, and by doing so, finds solutions to classic problems while developing new avenues.

In the first chapter, *Time-Varying Parameters as Ridge Regressions*, Kalman-filter based computations of random walk coefficients are replaced by a closed-form solution akin to OLS. In the second chapter, *The Macroeconomy as a Random Forest*, evolving coefficients are modeled and forecasted with a powerful machine learning algorithm instead of the widely used random-walk process. Conveniently, this generalization of time-varying parameters provides statistical efficiency and interpretability, which off-the-shelf ML algorithms cannot easily offer with macro data. Finally, the third chapter *To Bag is to Prune* answers the question: why can't Random Forest (RF) overfit? I show it is a surprising byproduct of randomized "greedy" algorithms – often deployed in the face of computational adversity. Then, I capitalize on the new insight by developing new high-performing non-overfitting algorithms.

Those three chapters are highly interconnected. The first chapter realizes that a classical time series model can be rewritten as a decades-old regression tool, which simplifies greatly both computations and tuning. But we are still fitting an old model. The second chapter notices that this old model made easier in the first chapter should itself be dismissed. Then, I propose a new algorithm that truly leverage the abundance of predictors

available to applied macroeconomists. The third chapter reflects on the empirical success of Random Forest – the driving force behind the second chapter – and formulate an explanation for it. Thus, we learn that ML methods (i) often beat traditional econometric methods when carefully adjusted for macroeconomic problems, (ii) can provide important insights about macroeconomic mechanisms, and (iii) owe their success to properties traditional econometric analysis would overlook.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1 : TIME-VARYING PARAMETERS AS RIDGE REGRESSIONS

1.1. Introduction

Economies change. Intuitively, this should percolate to the parameters of models characterizing them. To econometrically achieve that, a popular approach is Time-Varying Parameters (TVPs), where a linear equation's coefficients follow stochastic processes — usually random walks. Classic papers in the literature consider TVP Vector Auto Regressions (TVP-VARs) to study changing monetary policy (Primiceri, 2005) and evolving inflation dynamics (Cogley and Sargent, 2001, 2005).[1] Recently, such ideas were introduced to Jordà (2005)'s local projections (LPs) to obtain directly time-varying impulse response functions (Ruisi, 2019; Lusompa, 2020).[2]

In both the VAR and LP cases, important practical obstacles reduce the scope and applicability of TVPs. One is prohibitive computations limiting the model's size. Another is the difficulty of tuning the crucial amount of time variation. To address those and other pressing problems, I show that TVP models are ridge regressions (RR). The connection is useful: 50 years (Hoerl and Kennard, 1970) of RR widespread use, research and wisdom is readily transferable to TVPs. Among other things, this provides fast computations via a closed-form *dual* solution only using matrix operations. The amount of time variation is automatically tuned by cross-validation (CV). Adjustments for evolving residuals' volatility and heterogeneous parameter drifting speeds (random walk variances) are implemented via a 2-step ridge regression (henceforth 2SRR) the flagship model of this paper. In sharp contrast with the usual Bayesian machinery, it is incredibly easy to implement and to operate.[3] For instance, it will never face initialization and convergence issues since it avoids altogether MCMC simulations and filtering. Moreover, the setup is directly extendable to deploy additional shrinkage schemes (sparse TVP, reduced rank TVPs) recently proposed in the literature (Stevanovic, 2016; Bitto and Frühwirth-Schnatter, 2018). Finally, credible

[1] There is also a wide body of work using TVPs to study structural change in "great" macroeconomic (univariate) equations (Stock and Watson, 1996; Boivin, 2005; Blanchard et al., 2015).

[2] Well-known applications where time variation in LPs is obtained by interacting a linear specification with a "state of the economy" variable are Auerbach and Gorodnichenko (2012a) and Ramey and Zubairy (2018a).

[3] In its simplest form, it consists or creating many new regressors out of the original data and using it as a feature matrix in any RR software, which requires 3 lines of code (cross-validation, estimation, prediction).

regions are available since RR is alternatively a plain Bayesian regression. For the remainder of this introduction, I review the issues facing current TVP models, survey their related literature, and explain how the ridge approach can remedy those.

Computations. Using standard implementations allowing for stochastic volatility (SV) in TVP-VARs, researchers are limited to few lags (usually 2 for quarterly data) and a small number of variables (not more than 4 or 5) (Kilian and Lütkepohl, 2017). This constraint leaves the reader ever-wondering whether time variation is not merely the byproduct of omitted variables. Consequently, a growing number of contributions attempt to deal with the computational problem. In the state-space paradigm, Koop and Korobilis (2013) and Huber et al. (2020) proposed approximations to speed up MCMC simulations. Giraitis et al. (2014) and Kapetanios et al. (2019) drop the state-space paradigm altogether in favor of a nonparametric kernel-based estimator. Chen and Hong (2012) consider a similar approach to develop a test for smooth structural change while Petrova (2019) develops a Bayesian version particularly apt with large multivariate systems. While this type of framework allows the estimation of the desired big models, it is unclear how we can incorporate useful features such as heterogeneous variances for parameters (as in Primiceri (2005)). Further, there seems to be an artificial division between nonparametric and law-of-motion approaches. Later, by showing that random walk TVPs give rise to a ridge regression (which will also be a smoothing splines problem), it will become clear that random walks TVPs are no less nonparametric than TVPs obtained from the "nonparametric approach". Yet, RR implements nearly the same model as the benchmark Bayesian TVP-VAR and preserves the interpretation of TVPs as latent stochastic states. Keeping alive the parallel to a law of motion has some advantages — like an obvious prediction for tomorrow's coefficients.

Tuning and Forecasting. On the forecasting front, D'Agostino et al. (2013), Baumeister and Kilian (2014), and Pettenuzzo and Timmermann (2017) have all investigated, with varying angles, the usefulness of time variation to increase prediction accuracy. A critical choice underlying forecasting successes and failures is the amount of time variation. Notoriously, tuning parameter(s) determining it can largely influence prediction results and estimated coefficients, accounting for much of the cynicism regarding the transparency and reliability of TVP models. Amir-Ahmadi et al. (2018) propose to treat those pivotal hyperparameters as another layer of parameters to be estimated within the Bayesian procedure — and find this indeed helps. By showing the TVP–RR equivalence, this chapter defines even more clearly what is the fundamental tuning problem for this class of models. TVP models are simply standard (very) high-dimensional regressions which need to be regularized somehow. By construction, the unique ridge tuning parameter, in this context,

(Golub et al., 1979) mechanically corresponds to a ratio of two variances, that of parameter innovations and that of residuals. Hence, tuning λ via standard (and fast) cross-validation techniques deliver the holy quantity of how much time variation there is in the coefficients. Given how the quantity is paramount for both predictive accuracy and economic analysis, it is particularly comforting that it suddenly can be tuned the same way ridge λ's have been tuned for decades.

FANCIER SHRINKAGE. TVP models are densely parametrized which makes overfitting an enduring sword of Damocles. The RR approach makes this explicit: TVP models are linear regressions where parameters always outnumber observations — and by a lot. Precisely, the ratio of parameters to observations is always K, the number of original regressors. Clearly, things do get any better with large models. Assuming a random walk as a law of motion and enforcing it with a varying degree of rigidity (using λ in my approach) kills overfitting, provided the plain constant-coefficient models itself does not overfit.[4] However, an unpleasant side effect of λ "abuse" is that time variation itself is annihilated. Since this problem pertains to the class of models rather than the estimation method, I borrow insights from the recent literature to extend my framework into two directions. Firstly, I consider *Sparse* TVPs. This means that not all parameters are created equal: some will vary and some will not. This brings hope for larger models. If adding regressors actually make some other coefficients time-invariant, we are gaining degrees of freedom. In that spirit, Bitto and Frühwirth-Schnatter (2018), Belmonte et al. (2014), Korobilis (2014), and Hauzenberger et al. (2020) have proposed such extensions to MCMC-based procedures. In the RR setup, this amounts to the development of the Group Lasso Ridge Regression (GLRR) which is shown to be obtainable by simply iterating 2SRR. Secondly, another natural way to discipline TVPs is to impose a factor structure. This means that instead of trying to filter, say, 20 independent states, we can span these with a parsimonious set of latent factors. This extension is considered in Stevanovic (2016), de Wind and Gambetti (2014) and Chan et al. (2018). Such reduced rank restrictions are brought to this chapter's arsenal by developing a Generalized Reduced Rank Ridge Regression (GRRRR).

RESULTS. I first evaluate the method with simulations. For models of smaller size, where traditional Bayesian procedures can also be used, 2SRR does as well and sometimes better at recovering the true parameter path than the (Bayesian) TVP-VAR. This is true whether SV is involved or not. This is practical given that running *and* tuning 2SRR for such small models (300 observations, 6 TVPs per equation) takes less than 5 seconds to compute. Additionally, I evaluate the performance of 3 variants of the RR approach in a substantive

[4]Guaranteeing that a large constant-coefficients model behaves well often requires shrinkage of its own (Bańbura et al., 2010; Kadiyala and Karlsson, 1997; Koop, 2013).

forecasting experiment. 2SRR and its iterated extension provide sizable gains for interest rates and inflation, two variables traditionally associated with the need for time variation. I complete with an application to estimating large time-varying LPs (more than 4,500 TVPs) in a Canadian context using Champagne and Sekkel (2018)'s narrative monetary policy (MP) shocks. It is found that MP shocks long-run impact on inflation increased substantially starting from the 1990s (onset of inflation targeting), whereas the effects on real activity indicators (GDP, unemployment) became milder.

OUTLINE. Section 1.2 presents the ridge approach, its extensions, and related practical issues. Sections 1.3 and 1.4 report simulations and forecasting results, respectively. Section 1.5 applies 2SRR to (large) time-varying LPs. Tables, additional graphs and technical details are in the Appendix.

NOTATION. $\beta_{t,k}$ refers to the coefficient on regressor X_k at time t. To make things lighter, $\beta_t \in \mathbb{R}^K$ or $\beta_0 \in \mathbb{R}^K$ always refers to all coefficients at time t or time zero, respectively. Analogously, β_k represents the whole time path for the coefficient on X_k. $\boldsymbol{\beta} \in \mathbb{R}^{KT}$ is stacking all β_k's one after the other, for $k = 1, ..., K$. All this also applies to u and θ.

1.2. Time-Varying Parameters are Ridge Regressions

1.2.1. A Useful Observation

Consider a generic linear model with random walk time-varying parameters

$$y_t = X_t\beta_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2_{\epsilon_t}) \tag{1.1a}$$

$$\beta_t = \beta_{t-1} + u_t, \quad u_t \sim N(0, \Omega_u) \tag{1.1b}$$

where $\beta_t \in \mathbb{R}^K$, $X_t' \in \mathbb{R}^K$, $u_t \in \mathbb{R}^p$ and both y_t and ϵ_t are scalars. This chapter first considers a general single equation time series model and then discuss its generalization to the multivariate case in section 1.2.4. For clarity, a single equation in a VAR with M variable and P lags has $K = PM + 1$ parameters for each equation. For simplicity of exposition, I first impose $\Omega_u = \sigma^2_u I_K$ and $\sigma^2_{\epsilon_t} = \sigma^2_\epsilon \quad \forall t$. This means all parameters are assumed to vary equally a priori and constant variance of residuals. These assumptions will be relaxed in section (1.2.4). The textbook way of estimating (1.1) is to impose some value for $\frac{\sigma_\epsilon}{\sigma_u}$ and use the Kalman filter for linear Gaussian model (Hamilton, 1994). The advantages of the newly proposed methods will be more apparent when considering complications typically encountered in macroeconometric modeling (e.g. evolving volatility, heterogeneous variances for coefficients paths, unknown $\frac{\sigma_\epsilon}{\sigma_u}$ and a large X_t).

A useful observation is that (1.1) can be written as the penalized regression problem

$$\min_{\beta_1 \dots \beta_T} \frac{1}{T} \sum_{t=1}^{T} \frac{(y_t - X_t \beta_t)^2}{\sigma_\epsilon^2} + \frac{1}{KT} \sum_{t=1}^{T} \frac{\|\beta_t - \beta_{t-1}\|^2}{\sigma_u^2}. \tag{1.2}$$

This is merely an implication of the well-known fact that l_2 regularization is equivalent to opting for a standard normal prior on the penalized quantity (see, for instance sections 7.5-7.6 in Murphy 2012). Hence, model (1.3) implicitly poses $\beta_t - \beta_{t-1} \sim N(0, \sigma_u^2)$, which is exactly what model (1.1) also does. Defining $\lambda \equiv \frac{\sigma_\epsilon^2}{\sigma_u^2} \frac{1}{K}$, the problem has the more familiar look of

$$\min_{\beta_1 \dots \beta_T} \sum_{t=1}^{T} (y_t - X_t \beta_t)^2 + \lambda \sum_{t=1}^{T} \|\beta_t - \beta_{t-1}\|^2. \tag{1.3}$$

The sole hyperparameter of the model is λ and it can be tuned by cross-validation (CV).[5] This model has a closed-form solution as an application of generalized ridge regression (Hastie et al., 2015). In particular, it can be seen as the l_2 norm version of the "Fused" Lasso of Tibshirani et al. (2005) and embeds the economic assumption that coefficients evolve slowly. However, as currently stated, solving directly (1.3) may prove unfeasible even for models of medium size.

1.2.2. Getting a Ridge Regression by Reparametrization

The goal of this subsection is to rewrite the problem (1.3) as a ridge regression. Doing so will prove extremely useful at the conceptual level, but also to alleviate the computational burden dramatically. Related reparametrizations have been seldomly discussed in various literatures. For instance, it is evoked in Tibshirani et al. (2015) as a way to estimate "fused" Lasso via plain Lasso. Within to the time series realm, Koop (2003) discuss that a local-level model can be rewritten as a plain Bayesian regression. More recently, Korobilis (2019) uses it as a building block of his "message-passing" algorithm, and Goulet Coulombe et al. (2020a) use derivations inspired by those below to implement regularized lag polynomials in Machine Learning models. From now on, it is less tedious to use matrix notation. The fused ridge problem reads as

$$\min_{\boldsymbol{\beta}} (\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\beta})' (\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}' \boldsymbol{D}' \boldsymbol{D} \boldsymbol{\beta}$$

[5] This definition of λ does not imply it decreases in K since σ_u^2 will typically decrease with K to avoid overfitting.

where $\boldsymbol{D}$ is the first difference operator. $\boldsymbol{W} = [diag(X_1) \quad \dots \quad diag(X_K)]$ is a $T \times KT$ matrix. To make matters more visual, the simple case of $K = 2$ and $T = 4$ gives rise to

$$\boldsymbol{W} = \begin{bmatrix} X_{11} & 0 & 0 & 0 & X_{21} & 0 & 0 & 0 \\ 0 & X_{12} & 0 & 0 & 0 & X_{22} & 0 & 0 \\ 0 & 0 & X_{13} & 0 & 0 & 0 & X_{23} & 0 \\ 0 & 0 & 0 & X_{14} & 0 & 0 & 0 & X_{24} \end{bmatrix}.$$

The first step is to reparametrize the problem by using the relationship $\beta_k = C\theta_k$ that we have for all k regressors. C is a lower triangular matrix of ones (for the random walk case) and I define $\theta_k = [u_k \quad \beta_{0,k}]$. For the simple case of one parameter and $T = 4$:

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

For the general case of K parameters, we have

$$\boldsymbol{\beta} = \boldsymbol{C\theta}, \quad \boldsymbol{C} \equiv I_K \otimes C$$

and $\boldsymbol{\theta}$ is just stacking all the θ_k into one long vector of length KT. Note that the summation matrix C could accommodate easily for a wide range of law of motions just by changing summation weights. Actually, any process that can be rewritten (a priori) in terms of uncorrelated u's could be used. For instance, AR models of arbitrary order and RW with drifts would be straightforward to implement.[6] Furthermore, one could use C^2 in the random walk setup and obtain smooth second derivatives, e.i. a local-level model. While it clear that many more exotic configurations are only a C choice away, there is a clear advantage to random walks-based processes: the corresponding C has no parameter to estimate. If we wanted to consider an AR(1) process with a coefficient $\phi \in (0, 1]$, either a 2-step estimation procedure or cross-validating ϕ would be necessary. Thus, the ridge approach is possible, whether β_t's are random walks or not.

Using the reparametrization $\boldsymbol{\beta} = \boldsymbol{C\theta}$, the fused ridge problem becomes

$$\min_{\boldsymbol{\theta}} (\boldsymbol{y} - \boldsymbol{WC\theta})' (\boldsymbol{y} - \boldsymbol{WC\theta}) + \lambda \boldsymbol{\theta' C' D' DC\theta}$$

and it is now clear what should be done. Let $\boldsymbol{Z} \equiv \boldsymbol{WC}$ and use the fact that $\boldsymbol{D} = \boldsymbol{C}^{-1}$ to

[6] In the latter case, it can be shown that one simply needs to add regressors $t * X_t$ to those implied by the RW without drift, that is the Z_t's to be detailed later.

obtain the desired ridge regression problem

$$\min_{\boldsymbol{\theta}} (\boldsymbol{y} - \boldsymbol{Z\theta})' (\boldsymbol{y} - \boldsymbol{Z\theta}) + \lambda \boldsymbol{\theta'\theta}. \tag{1.4}$$

Again, for concreteness, the matrix $\boldsymbol{Z} = \boldsymbol{WC}$ looks like

$$\boldsymbol{Z} = \begin{bmatrix} X_{11} & 0 & 0 & 0 & X_{21} & 0 & 0 & 0 \\ X_{12} & X_{12} & 0 & 0 & X_{22} & X_{22} & 0 & 0 \\ X_{13} & X_{13} & X_{13} & 0 & X_{23} & X_{23} & X_{23} & 0 \\ X_{14} & X_{14} & X_{14} & X_{14} & X_{24} & X_{24} & X_{24} & X_{24} \end{bmatrix}$$

in the $K = 2$ and $T = 4$ case.[7] The solution to the original problem is thus

$$\hat{\boldsymbol{\beta}} = \boldsymbol{C}\hat{\boldsymbol{\theta}} = \boldsymbol{C}(\boldsymbol{Z'Z} + \lambda I_{KT})^{-1}\boldsymbol{Z'y}. \tag{1.5}$$

This is really just a standard (very) high-dimensional Ridge regression.[8] These derivations are helpful to understand TVPs, which is arguably one of the most popular nonlinearity in modern macroeconometrics. (1.5) is equivalent to that of a first-order smoothing splines estimator.[9] More generally, the equivalence between Bayesian stochastic process estimation and splines has been known since Kimeldorf and Wahba (1970). Following along, considering a local-level model for β_t would yield second order smoothing splines. Clearly, random walk TVPs and their derivatives can hardly be described as "more parametric" than kernel-based approaches: splines methods are prominent within the nonparametric canon. Furthermore, the basis expansion and associated penalty $\boldsymbol{D'D}$ in the original fused problem can be approximated by a very specific reproducing kernel (Dagum and Bianconcini, 2009b). This brings the one-step estimator in the direction of the kernel proposition of Giraitis et al. (2018).

In other words, assuming a law of motion implies assuming implicitly a certain kernel. The ridge approach makes clear that there is nothing special about random walks beyond that it is just another way of doing a nonparametric regression. This new view of the problem – in sync with the reality of implementation – allows dispensing with some theoretical

[7] The structure of $\boldsymbol{Z}$'s echoes to Castle et al. (2015) "indicator-saturation" approach to detect location shifts in the intercept via model selection tools. TVPs via RR first generalize the approach by interacting all individual regressors with a stray of shifting indicators. Then, rather than selecting one or few of them (sparsity), they are all kept in the model by constraining to incremental location shifts only via heavy ridge shrinkage (smooth time variation).

[8] In this section, I assumed for simplicity that we wish to penalize equally each member of θ which is not the case in practice. It is easy to see why starting values β_0 should have different (smaller) penalty weights and this will be relaxed as a special case of the general solution presented in section 1.2.4.

[9] Also, it has the flavor of Hoover et al. (1998) for time series rather than panel data.

worries, like the one that a random walk parameter is not bounded, which are of little empirical relevance.

At this point, the computational elephant is still in the room since the solution implies inverting a $KT \times KT$ matrix. Avoiding this inversion is crucial; otherwise the procedure will be limited to models of similar size to Primiceri (2005). Fortunately, there is no need to invert that matrix.

1.2.3. Solving the Dual Problem

The goal of this subsection is to introduce a computationally tractable way of obtaining the ridge estimator $\hat{\boldsymbol{\beta}}$ in (1.5). It is well known from the splines literature (Wahba, 1990) and later generalized by Schölkopf et al. (2001) that for a $\hat{\boldsymbol{\theta}}$ that solves problem (1.4), there exist a $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^T$ such that $\hat{\boldsymbol{\theta}} = \boldsymbol{Z}'\hat{\boldsymbol{\alpha}}$. Using this knowledge about the solution, we can replace $\boldsymbol{\theta}$ in (1.4) to obtain

$$\min_{\boldsymbol{\alpha}} \left(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{Z}'\boldsymbol{\alpha}\right)' \left(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{Z}'\boldsymbol{\alpha}\right) + \lambda \boldsymbol{\alpha}'\boldsymbol{Z}\boldsymbol{Z}'\boldsymbol{\alpha}.$$

The solution to the original problem becomes

$$\hat{\boldsymbol{\beta}} = \boldsymbol{C}\boldsymbol{Z}'\hat{\boldsymbol{\alpha}} = \boldsymbol{C}\boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}' + \lambda \boldsymbol{I}_T)^{-1}\boldsymbol{y}. \tag{1.6}$$

When the number of observations is smaller than the number of regressors, the *dual* problem allows to obtain numerically identical estimates by inverting a smaller matrix of size T. Since sample sizes for macroeconomic applications quite rarely exceed 700 observations (US monthly data from the 1960s), the need to invert that matrix is not prohibitive. While computational burden does still increase with K, it increases much slowly since the complexity of matrix multiplication is now $O(KT^3)$ and $O(T^3)$ for matrix inversion. Solving the primal problem, one would be facing $O(K^2T^3)$ and $O(K^3T^3)$ complexities respectively. Concretely, solving the dual problem brings high-dimensional TVP models to be more feasible than ever. Estimating a small TVP-VAR with T=300 with 6 lags and 5 variables takes roughly 10 seconds on a standard computer. This includes hyperparameters optimization by cross-validation, which is usually excluded in the standard MCMC methodology. However, the latter provides full Bayesian inference. A VAR(20) with the same configuration takes less than 2 minutes. Section 1.3 reports detailed results on this.

1.2.4. Heterogenous Parameter and Residual Variances

For pedagogical purposes, previous sections considered the simpler case of $\Omega_u = \sigma_u^2 I_K$ and no evolving volatility of residuals. I now generalize the solution (1.6) to allow for heterogeneous $\sigma_{u_k}^2$ (a diagonal $\Omega_u \neq \sigma_u^2 I_K$) and $\sigma_{\epsilon,t}^2$. The end product is 2SRR, this chapter's flagship model.

New matrices must be introduced. First, we have the standard matrix of time-varying residuals variance $\Omega_\epsilon = diag([\sigma^2_{\epsilon_1} \quad \sigma^2_{\epsilon_2} \quad ... \quad \sigma^2_{\epsilon_T}])$. I assume in this section that both Ω_ϵ and Ω_u are given and will provide a data-driven way to obtain them later. Departing from the homogeneous parameter variances assumption implies that the sole hyperparameter λ must now be replaced by an enormous $KT \times KT$ diagonal matrix $\boldsymbol{\Omega_u} = \Omega_u \otimes I_T$ which is fortunately only used for mathematical derivations. For convenience, I split $\boldsymbol{Z}$ in two parts so they can be penalized differently. Hence, the original $\boldsymbol{Z} \equiv [X \quad \boldsymbol{Z_{-0}}]$. The new primal problem is

$$\min_{\boldsymbol{u},\beta_0} (\boldsymbol{y} - \boldsymbol{X}\beta_0 - \boldsymbol{Z_{-0}u})' \, \Omega_\epsilon^{-1} \, (\boldsymbol{y} - \boldsymbol{X}\beta_0 - \boldsymbol{Z_{-0}u}) + \boldsymbol{u'\Omega_u^{-1}u} + \lambda_0 \beta_0' \beta_0. \tag{1.7}$$

For convenience, let the $\boldsymbol{\Omega_\theta}$ matrix that stacks on the diagonal all the parameters prior variances, which allow rewriting the problem in a more compact form

$$\min_{\boldsymbol{\theta}} (\boldsymbol{y} - \boldsymbol{Z\theta})' \, \Omega_\epsilon^{-1} \, (\boldsymbol{y} - \boldsymbol{Z\theta}) + \boldsymbol{\theta'\Omega_\theta^{-1}\theta}.$$

Using a GLS re-weighting scheme on observations **and** regressors, we get a "new" standard primal ridge problem

$$\min_{\tilde{\boldsymbol{\theta}}} (\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{Z}}\tilde{\boldsymbol{\theta}})' \, (\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{Z}}\tilde{\boldsymbol{\theta}}) + \tilde{\boldsymbol{\theta}}'\tilde{\boldsymbol{\theta}}.$$

where $\tilde{\boldsymbol{\theta}} = \boldsymbol{\Omega_\theta^{-\frac{1}{2}}u}$, $\tilde{\boldsymbol{Z}} = \Omega_\epsilon^{-\frac{1}{2}} \boldsymbol{Z\Omega_\theta^{\frac{1}{2}}}$ and $\tilde{\boldsymbol{y}} = \Omega_\epsilon^{-\frac{1}{2}} \boldsymbol{y}$. Solving this problem by the "dual path" and rewriting it in terms of original matrices gives the general formula

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\Omega_\theta Z'}(\boldsymbol{Z\Omega_\theta Z'} + \Omega_\epsilon)^{-1}\boldsymbol{y}. \tag{1.8}$$

Equation (1.8) contains all the relevant information to back out the parameters paths, provided some estimates of matrices $\boldsymbol{\Omega_\theta}$ and Ω_ϵ.[10]

Implementation

The solution (1.8) takes $\boldsymbol{\Omega_\theta}$ and Ω_ϵ as given. In this section, I provide a simple adaptive algorithm to get the heterogeneous variances model estimates empirically. Multi-step approaches to obtain the obtain analogs of $\boldsymbol{\Omega_\theta}$ and Ω_ϵ have been proposed in Ito et al. (2017) and Giraitis et al. (2014). It is conceptually convenient to extent the original penalized regression (1.3) for heterogeneous variances of parameters and residuals. By bringing back

[10]This two-step procedure is partly reminiscent of Ito et al. (2014) and Ito et al. (2017)'s non-Bayesian Generalized Least Squares (GLS) estimator, where a two-step strategy is also proposed for reasons similar to the above. Their approach could have a ridge regression interpretation with certain tuning parameters fixed. However, the absence of tuning leads to overfitting and the GLS view cannot handle bigger models because the implied matrices sizes are even worse than that of the *primal* ridge problem discussed earlier.

to σ_ϵ and σ_u their respective subscripts and moving σ_{ϵ_t} to the penalty side of the program, ones obtains

$$\min_{\beta_1 \dots \beta_T} \sum_{t=1}^{T} (y_t - X_t \beta_t)^2 + \sum_{k=1}^{K} \sum_{t=1}^{T} \lambda_t \lambda_k \|\beta_{k,t} - \beta_{k,t-1}\|^2. \tag{1.9}$$

It is clear from this perspective that neglecting time variation in the variance of residuals can be understood as forcing homogeneity of tuning parameters. Indeed, evolving residuals volatility, when seen as λ_t, implies a time-varying level of smoothness. Neglecting it does not imply biased estimates, but inefficient ones. In other words, fixing $\lambda_t = \lambda \ \ \forall t$ prevents from using extra regularization to reduce the estimation variance of β_t during high volatility episodes. Conversely, this means low volatility periods may suffer from over-regularized estimates. Fortunately, dealing with heterogeneous regularization is exactly the motivation behind the panoply of "adaptive" algorithms tuning hyperparameters in a data-driven way – usually as a special case of a broader EM algorithm (Murphy, 2012). Algorithm 1 follows along this perspective and proposes a two-step ridge regression (2SRR) which uses a first stage plain RR to gather the necessary hyperparameters in one swift blow.

Algorithm 1 2SRR

1: Use the homogeneous variances approximation. That is, get $\hat{\boldsymbol{\theta}}_{\mathbf{1}}$ with (1.6).[11] λ is obtained by CV.
2: Obtain $\hat{\sigma}^2_{\epsilon,t}$ by fitting a volatility model to the residuals from step 1.[12] Normalize $\hat{\sigma}^2_{\epsilon,t}$'s mean to 1.
3: Obtain $\hat{\sigma}^2_{u,k} = \frac{1}{T}\sum_{\tau=1}^{T} \hat{u}^2_{k,\tau}$ for $k = 1, ..., K$. Normalize the new vector to have its previous mean ($1/\lambda$).
4: Stack these into matrices $\boldsymbol{\Omega_u}$ and Ω_ϵ. Use solution (1.8) to rerun CV and get $\hat{\boldsymbol{\theta}}_{\mathbf{2}}$, the final estimator.[13]

While reweighting observations is nothing new from an econometric perspective, reweighting variables is less common since its effect is void unless there is a ridge penalty. 2SRR (and eventually GLRR in section 1.2.5) makes use of adaptive (or data-driven) shrinkage. Adaptive prior tuning has a long tradition in Bayesian hierarchical modeling (Murphy, 2012) but the term itself came to be associated with the Adaptive Lasso of Zou (2006). To modulate the penalty's strength in Lasso, the latter suggest weights based on preliminary OLS (or Ridge) estimates. Those, taken as given, may be contaminated with a considerable amount of noise, especially when the regression problem is high-dimensional (like the one considered here). Thankfully, adaptive weights in 2SRR have a natural group structure, which drastically improves their accuracy by the simple power of averaging.

Choosing λ

Derivations from previous sections rely on a given λ. This section explains how to obtain the amount of time variation by CV (as alluded to in Algorithm 1), and how that new strategy compares to more traditional approaches to the problem.

Within the older literature where TVPs were obtained via classical methods, estimating the parameters variances (σ_u^2 in my notation) made MLE's life particularly difficult (Stock and Watson, 1998c; Boivin, 2005). In the RR paradigm, with σ_u^2 expressed through λ, it is apparent as to why those issues arose in the first place: nobody would directly maximize an in-sample likelihood to obtain ridge's λ. In the Bayesian TVP-VAR literature, it is common to implicitly fix the influential parameter to a value loosely inspired by Primiceri (2005). Some consider a few and argue ex-post about their relative plausibility (D'Agostino et al., 2013). Within this paradigm or that of RR, it is known that a high λ (or its equivalent) guarantees well-behaved paths but also shrinks β_k to a horizontal line. As a result, one is often left wondering whether the recurrent finding of not so much time variation is not merely a reflection of the prior (and its absence of tuning) doing all the talking.

More recently, Amir-Ahmadi et al. (2018) propose to estimate hyperparameters within the whole Bayesian procedure and find that doing so can strongly improve forecasting results. This suggests that opting for a data-driven choice of λ is the preferable strategy. Nonetheless, CV is not carried without its own theoretical backing. Golub et al. (1979)'s Theorem 2 shows that the λ minimizing the expected generalization error as calculated by generalized CV is equal to the "true" ratio of the parameters prior variance and that of residuals. In the case of TVPs, this is a multiple of $\frac{\sigma_\epsilon}{\sigma_u}$, the ratio guiding the amount of time variation in the coefficients. Of course, the specific elements of this ratio are only of interest if one truly believes random walks are being estimated (in contrast to simply being a tool for non-parametric estimation as discussed in section 1.2.3). Also, the ridge approach makes clear that only the ratio influences out-of-sample performance rather than the denominator or numerator separately, and should be the focus of tuning so to optimally balance bias and variance. Thus, by seeing the TVP problem as the high-dimensional regression it really is, one avoids the "pile-up" problem inherent to *in-sample* maximum likelihood estimation (Grant and Chan, 2017) and the usual necessity of manually selecting a highly influential parameter in the Bayesian paradigm.

I use k-fold CV for convenience, but anything could be used – conditional on some amount of thinking about how to make it computationally tractable. This is also what standard RR implementations use, like `glmnet` in `R`. A concern is that k-fold CV might be overoptimistic with time series data. Fortunately, Bergmeir et al. (2018) show that without residual

autocorrelation, k-fold CV is consistent. Assuming models under consideration include enough lags of y_t, this condition will be satisfied for one-step ahead forecasts. Moreover, Goulet Coulombe et al. (2019) report that macroeconomic forecasting performance can often be improved by using k-fold CV rather than a CV procedure that mimics the recursive pseudo-out-of-sample experiment.

One last question to address is that of the hypothesized behavior of λ as a function of T and K. In a standard ridge context (i.e., with constant parameters) where the number of regressors is fixed as the sample increases, $\lambda \to 0$ as T grows and we get back the OLS solution. This is not gonna happen in the TVP setup since the effective number of regressors is KT, and it clearly grows as fast as T. When it comes to K, λ will tend to increase with it simply because a larger model requires more regularization. This is a feature of the model, not the ridge estimation strategy. But the latter helps make it crystal clear. This means there is little hope to find a lot of time variation in a large model – provided it is tuned to predict well.

Credible Regions

In various applications, quantifying uncertainty of $\boldsymbol{\beta}$ is useful. This is possible for 2SRR by leveraging the link between ridge and a plain Bayesian regression (Murphy, 2012). In the homoscedastic case, we need to obtain

$$V_{\boldsymbol{\beta}} = \boldsymbol{C}(\boldsymbol{Z}'\boldsymbol{Z} + \boldsymbol{\Omega_\theta}^{-1})^{-1}\boldsymbol{C}'\hat{\sigma}_\epsilon^2.$$

This is precisely the large matrix we were avoiding to invert earlier. However, this is much less of an issue here because we only have to do it once at the very end of the procedure.[14] Since heterogeneous volatility is incorporated in a GLS fashion and taken as given in the 2nd stage, the bands for the heteroscedastic case can be obtained by using the formula above with the properly re-weighted data matrix $\boldsymbol{Z}$.

In the simple case where $\Omega_u = \sigma_u^2 I_K$ and $\Omega_\epsilon = \sigma_\epsilon^2 \ I_T$, there is a clear Bayesian interpretation allowing the use of the posterior variance formula for linear Bayesian regression. However, it treats the cross-validated λ as known. This also means these credible regions are conditional on σ_ϵ^2. In a similar line of thought, I treat the hyperparameters inherent to 2SRR as given when computing the bands. That is, I regard steps 1–3 of the algorithm as a practical approximation to a full-blown cross-validation operation on both diagonals of Ω_u and Ω_ϵ matrices.

[14] To compute the posterior mean, only one inversion is needed. However, to cross-validating λ requires a number of inversions that is the multiple of the number of folds (usually 5) and the size of potential λ's grid.

From Univariate to Multivariate

Many applications of TVPs are multivariate and derivations so far have focused on the univariate case. This section details the modifications necessary for a multivariate 2SRR.

Since both Ω_u and Ω_ϵ are equation-specific, we must use (1.8) for each $\boldsymbol{y}$. However, all estimation procedures proposed in this chapter have the homogeneous case of section 1.2.3 as a first step. This is usually the longer step since it is where cross-validation is done. Hence, it is particularly desirable not to have computations of the first step scaling up in M, the number of variables in the multivariate system. Thankfully, in the plain ridge case, we can obtain all parameters of the system in one swift blow, by stacking all $\boldsymbol{y}$'s into $\boldsymbol{Y}$ (a $T \times M$ matrix) and computing

$$\hat{\boldsymbol{\Theta}} = \boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{Z}' + \lambda \boldsymbol{I}_T)^{-1}\boldsymbol{Y} = \boldsymbol{P}_Z^\lambda \boldsymbol{Y}. \tag{1.10}$$

This is precisely the approach that will be used as a first step for any multivariate extension. Of course, this works because the multivariate model has the same regressor matrix for each equation (like VARs and LPs). In this common case, $\boldsymbol{P}_Z^\lambda$ is the same for all equations and cross-validation still implies inverting $(\boldsymbol{Z}\boldsymbol{Z}' + \lambda \boldsymbol{I}_T)$ as many times as we have candidates for λ. That is, even if we wish to have a different λ for each equation in the first step, computing time does not increase in M, except for matrix multiplication operations which are much less demanding.[15]

When entering multivariate territory, modeling the residuals covariances – a necessary input to structural VARs (but not forecasting) – arises as an additional task. The number of TVPs entering the $\boldsymbol{\Omega_\epsilon}$ matrix can quickly explode. There are $\frac{M(1+M)}{2}$ of them. Here, I quickly describe a workaround for the computational issues this could engender. Let $\tilde{\eta}_t = vec^*(\boldsymbol{\epsilon_t}\boldsymbol{\epsilon_t}')$ where $*$ means only the non-redundant covariances are kept. Let $\tilde{\boldsymbol{\eta}}$ be the $T \times \frac{M(1+M)}{2}$ matrix that binds them all together. We can get the whole set of paths $\hat{\boldsymbol{\eta}}$ with

$$\hat{\boldsymbol{\eta}} = (\boldsymbol{I}_T + \varphi \boldsymbol{D}'\boldsymbol{D})^{-1}\tilde{\boldsymbol{\eta}} \tag{1.11}$$

where φ is a smoothness hyperparameter (just like λ) and $\boldsymbol{D}$ is the matrix difference operator described above. In this new case, CV can be conducted very fast even if φ differs by elements of $\hat{\boldsymbol{\eta}}$. Of course, if φ are heterogeneous, we may have to invert $\frac{M(1+M)}{2}$ times a $T \times T$ matrix. While this may originally appear like some form of empirical Waterloo, it is not. In practice, one would reasonably consider a grid for φ's that has between 10 and

[15] Precisely, cross-validation implies calculating # of folds $\times$ # of λ's the $\boldsymbol{P}_Z^\lambda$. Then, these matrices can be used for the tuning of every m equation, which is precisely why the computational burden only very mildly increases in M.

20 elements. By forming $\tilde{\boldsymbol{\eta}}$'s subgroups that share the same φ, one has to invert at most 20 matrices.

1.2.5. Extensions

Iterating Ridge to Obtain Sparse TVPs

Looking at the 2SRR's algorithm, one may rightfully aks: "why not iterate it further?" This section provides a way to iterate it so that not only it fine tunes Ω_u but also set some of its elements to zero. That is, some parameters will vary and some will not. Applications in the literature often suggest that the standard TVP model may be wildly inefficient. A quick look at some reported TVP plots (D'Agostino et al., 2013) suggests there are potential efficiency gains waiting for harvest by *Sparse* TVPs. Those have already been proposed in the standard Bayesian MCMC paradigm most notably by Bitto and Frühwirth-Schnatter (2018) and Belmonte et al. (2014). However, such an extension would be more productive if it were implemented in a framework which easily allow for the computation of the very models that could benefit most from it — the bigger ones.

The new primal problem is

$$\min_{\boldsymbol{u}} (\boldsymbol{y} - \boldsymbol{X}\beta_0 - \boldsymbol{Z_{-0}}\boldsymbol{u})' \Omega_\epsilon^{-1} (\boldsymbol{y} - \boldsymbol{X}\beta_0 - \boldsymbol{Z_{-0}}\boldsymbol{u}) + \boldsymbol{u'}(\Omega_u^{-1} \otimes I_T)\boldsymbol{u} + \xi tr(\Omega_u^{\frac{1}{2}}), \tag{1.12}$$

which is just adding the penalty $\xi tr(\Omega_u^{\frac{1}{2}})$ to (1.7).[16] In the RR paradigm, it is quite straightforward to implement: it corresponds to a specific form of Group Lasso — a Group Lasso ridge regression (GLRR). Mechanically, making a parameter constant amounts to dropping the group of regressors $\boldsymbol{Z}_k$ corresponding to the basis expansion of X_k making it time-varying. That is, for this "group", λ_k is set to infinity. The proposed implementation, formalized by Algorithm 2, amounts to iterating ridge regressions and updating penalty weights in a particular way. This estimation approach is inspired by Grandvalet (1998)'s proposition of using Adaptive Ridge to compute the Lasso solution. The insight has since been recuperated by Frommlet and Nuel (2016) and Liu and Li (2014) to implement l_0 regularization in a way that makes computations tractable. In particular, Liu and Li (2014) show that such algorithm has three desirable properties. It converges to a unique minimum, it is consistent and has the oracle property. GLRR goes back to implementing the l_1 norm by Adaptive Ridge as in Grandvalet (1998) but extend it to do Group Lasso and add a Ridge penalty within selected groups. Many derivations details are omitted from the main text and can be found in Appendix 1.7.1.

As we will see in simulations, iterative weights can help in many situations, but not all.

[16] The properties of such a program are already known in the Splines/non-parametrics literature because it corresponds to a special case of the Component Selection and Shrinkage Operator (COSSO) of Lin et al. (2006)

Algorithm 2 GLRR

1: Initiate the procedure with $\hat{\boldsymbol{\theta}}_{\mathbf{1}}$ or $\hat{\boldsymbol{\theta}}_{\mathbf{2}}$ from Algorithm 1. Keep the sequence of $\sigma^{2,(1)}_{u_k}$'s and the chosen $\lambda_{(1)}$. Set $\tilde{\lambda} = \lambda_{(1)}$. Choose a value for α. In applications, it is set to 0.5.
2: Iterate the following until convergence of λ_{u_k}'s. For iteration i:
   1. Use solution (1.8) to get $\hat{\boldsymbol{\theta}}^{(i)}_{\mathbf{3}}$.
   2. Obtain $\hat{\sigma}^{2,(i)}_{u,k}$ the usual way and normalize them to have mean of 1. Generate next step's weights using
   $$\lambda^{(i+1)}_{u_k} \leftarrow \tilde{\lambda}\left[\alpha\frac{1}{\sigma^{2,(1)}_{u_k}} + (1-\alpha)\frac{1}{\sigma^{(i)}_{u_k}}\right]$$
   on the diagonal of $\Omega_u^{-1,(i)}$. The formula is derived in Appendix 1.7.1.
   3. Obtain $\hat{\sigma}^2_{\epsilon,t}$ by fitting a volatility model to the residuals from step 1. Normalize $\hat{\sigma}^2_{\epsilon,t}$'s mean to 1 and input it to $\Omega^{(i)}_{\epsilon}$.
3: Use solution (1.8) with the converged Ω_ϵ and Ω_u to get $\hat{\boldsymbol{\theta}}_{\mathbf{3}}$, the final estimator.[17]

A relevant empirical example of where it can help in discovering that only the constant is time-varying, an important and frequently studied special case (Götz and Hauzenberger, 2018). One where it can fail is by shutting down many coefficients that were varying only slightly, but jointly. An algorithm tailored for the latter situation is the subject of the next subsection.

Reduced Rank Restrictions

As the TVP-VAR or -LPs increase in size, more shrinkage is needed to keep prediction variance in check. Unsettlingly, chronic abuse of the smoothness prior delivers the smoothest TVP ever: a time-invariant parameter. Looking at this problem through the lenses of RR makes this crystal clear. The penalty function is, essentially, a "time-variation" budget constraint. Thus, when estimating bigger models, we may want to reach for more sophisticated points on the budget line. This subsection explores an extension implementing reduced-rank restrictions – another recent proposition in the TVP literature.

A frequent empirical observation, dating back to Cogley and Sargent (2005), is that β_t's can be spanned very well by a handful of latent factors. de Wind and Gambetti (2014), Stevanovic (2016) and Chan and Eisenstat (2018) exploit this that directly by implementing directly a factor structure within the model. It is clear that dimensionality can be greatly reduced if we only track a few latent states and impose that evolving parameters are linear combinations of those, *Dense* TVPs. Additionally, it can be combined with the idea of section 1.2.5 that not all parameters vary to get sparse and dense TVPs via a *Generalized Reduced Rank Ridge Regression* (GRRRR). "Generalized" comes from the fact that what will be proposed here is somewhat more general than what Mukherjee and Zhu (2011) have coined as Reduced Rank Ridge Regression (RRRR) – or even the classic Anderson (1951)

Reduced Rank regression. Precisely, the model under consideration here is *univariate*. The reduced rank restrictions will be applied to a matrix $\boldsymbol{U} = vec^{-1}(\boldsymbol{u})$ where $\boldsymbol{u}$ are the coefficients from an univariate ridge regression.[18]

The measurement equation from a TVP model can be written more generally as

$$\boldsymbol{y} = \boldsymbol{X}\beta_0 + \boldsymbol{Z}vec(\boldsymbol{U}) + \epsilon \tag{1.13}$$

$$\boldsymbol{U} = A\boldsymbol{S} \tag{1.14}$$

where β_0 are still the starting values for the coefficients and A is a $K \times K$ matrix and S is a $K \times T$ matrix. For identification's sake, the rows of S are imposed to a have a variance of one. The A matrix scales and/or transforms the few (potentially uncorrelated) components of S. The homogeneous variance model of section (1.2.1) correspond to $A = \frac{1}{\sqrt{\lambda}} I_K$ and S is just a matrix of the normalized u's. The heterogeneous variances model, as implemented by 2SRR, corresponds to $A = \Omega_u^{\frac{1}{2}}$ where Ω_u is a diagonal matrix with (possibly) distinct entries. Sparse TVPs discussed earlier consists in setting some diagonal elements of A to zero.

Overfitting complementarily can be dealt with by reducing the rank of the generic A and S. Thus, we can have $A = \Lambda$ being $K \times r$ and $S = F$ being $r \times T$, which, with some additional orthogonality restrictions, corresponds conceptually and notationally to traditional factor model. The new primal problem is

$$\min_{\Lambda, \boldsymbol{F}, \beta_0} \left(\boldsymbol{y} - \boldsymbol{X}\beta_0 - \boldsymbol{Z}vec(\Lambda\boldsymbol{F})\right)' \Omega_\epsilon^{-1} \left(\boldsymbol{y} - \boldsymbol{X}\beta_0 - \boldsymbol{Z}vec(\Lambda\boldsymbol{F})\right) + \boldsymbol{f}'\boldsymbol{f} + \xi\|l\|_1 \tag{1.15}$$

where $\boldsymbol{f} = vec(\boldsymbol{F})$ and $l = vec(\Lambda)$.[19] This is neither Lasso or Ridge. However, there is still a way to implement an iterative procedure sharing a resemblance to the updates needed to estimate a regularized factor model as in Bai and Ng (2017). First, note these two linear algebra facts:

$$vec(\Lambda\boldsymbol{F}) = (I_T \otimes \Lambda)\boldsymbol{f} \tag{1.16}$$

$$vec(\Lambda\boldsymbol{F}) = (\boldsymbol{F}' \otimes I_K)l. \tag{1.17}$$

These two identities are of great help: they allow for the problem to be split in two sim-

[18] This can be done because $\boldsymbol{u}$ has an obvious block structure. It has two dimensions, K and T, that we can use to create a matrix. Note that the principle could be applied (perhaps in a less compelling way) to a constant parameter VAR with many lags where the dimensions of the matrix would be M and P.

[19] To make the exposition less heavy, I assume throughout this section that Ω_ϵ is given and that β_0 are not penalized in any way. Everything below goes through if we drop these simplifications and adjust algorithms accordingly.

ple linear penalized regressions. The solution to (1.15) can be obtained by the following maximization-maximization procedure.

1. Given Λ, we can solve

$$\min_{f,\beta_0} \left(y - X\beta_0 - Z^{\Lambda} f\right)' \Omega_{\epsilon}^{-1} \left(y - X\beta_0 - Z^{\Lambda} f\right) + \lambda f' f \tag{1.18}$$

   where $Z^{\Lambda} = Z(I_T \otimes \Lambda)$. This is just RR.

2. Given F, we can get the solution to

$$\min_{l,\beta_0} \left(y - X\beta_0 - Z^{F} l\right)' \Omega_{\epsilon}^{-1} \left(y - X\beta_0 - Z^{F} l\right) + \xi \|l\|_1 \tag{1.19}$$

   where $Z^{F} = Z(F' \otimes I_K)$. This is just Lasso.[20]

A first observation is that this problem is biconvex. A second one is that at each step, the objective function is further minimized and the objective is bounded from below. Hence, alternating these steps generate a monotonic sequence that converges to a (local) minima.[21] In terms of implementation, one must be carefully imposing the identification restriction of the factor model at all times. Algorithm 3 summarizes this and other practical aspects.

Algorithm 3 GRRRR

1: Get $\hat{\boldsymbol{\theta}}_{\mathbf{2}}$ from Algorithm 1 or plain RR. Estimate $F^{(1)}$ and $\Lambda^{(1)}$ by fitting a factor model to the u's. Choose r the number of factor using a citerion of choice.[22]
2: Iterate the following until convergence. For iteration $i > 1$:
   1. Run (1.18) to get $F^{(i)}$ given $\Lambda^{(i-1)}$. Orthogonalize factors.
   2. Run (1.19) to get $\Lambda^{(i)}$ given $F^{(i)}$. Orthogonalize loadings.
   3. Obtain $\hat{\sigma}^2_{\epsilon,t}$ by fitting a volatility model to (current) residuals. Normalize $\hat{\sigma}^2_{\epsilon,t}$'s mean to 1 and input it to $\Omega_{\epsilon}^{(i)}$.

It is noteworthy that doing Lasso on the loadings Λ operates a fusion of sparse and dense TVPs. If a parameter β_k does not "load" on any of the factors (because the vector Λ_k is shrunk perfectly to 0), we effectively get a constant β_k. In the resulting model, a given parameter can vary or not, and when it does, it shares a common structure with fellow parameters also selected as time-varying.

In Appendix 1.7.2, I present the multivariate extension to GRRRR and discuss its connection to Kelly et al. (2017)'s Instrumented PCA estimator for asset pricing models. Further,

[20] This could also be a RR if we wished to implement dense parameters only. In practice, elastic net with $\alpha = 0.5$ is the wiser choice (vs Lasso) given the strong correlation between the generated predictors.

[21] The other legitimate question is whether this algorithm converges to the solution of 1.15. It turns out to be a modification of Tibshirani et al. (2015) (Chapter 4) alternative algorithm for Lin et al. (2006)'s COSSO. The additional steps are orthogonalization of factors and loadings as in Bai and Ng (2017).

in Appendix 1.7.3, I write the GRRRR updates using summation notation for the simpler $r = 1$ case, which presents an obvious pedagogical advantage over *vec* and Kronecker products operations.

1.3. Simulations

The simulation study investigates how accurately the different estimators proposed in this chapter can recover the true parameters path. Moreover, computational times will be reported and discussed for various specifications.

I consider three numbers of observations $T \in \{150, 300, 600\}$. Most of the attention will be dedicated to $T = 300$ since it is roughly the number of US quarterly observations we will have 15 years from now. The size of the original regressor matrix X is $K \in \{6, 20, 100\}$ and the first regressor in each is the first lag of y. Figure 4 display the 5 types of parameters path f_i that will serve as basic material: cosine, quadratic trend, discrete break, a pure random walk and a linear trend with a break. f_1, f_2 and f_4 "fit" relatively well with the prior that coefficients evolve smoothly whereas f_3 and f_5 can pose more difficulties. In those latter situations, TVP models are expected to underperform.[23] The design for simulations S_1, S_2, S_3 and S_4 can be summarized in a less cryptic fashion as

- $\boldsymbol{S_1}$: $\beta_{k,t}$ follow the red line or is time invariant
- $\boldsymbol{S_2}$: $\beta_{k,t}$ follow the yellow line, the negative of it or is time invariant
- $\boldsymbol{S_3}$: $\beta_{k,t}$ is either the green line or the red one in equal proportions, otherwise time-invariant.
- $\boldsymbol{S_4}$: $\beta_{k,t}$ is a random mixture (loadings are drawn from a normal distribution) from the red, purple and blue lines. Some coefficients are also time-invariant.

The considered proportions of TVPs within the K parameters are ${}^{K^*}/_{K} \in \{0.2, 0.5, 1\}$. Formally, we have

$$
\begin{aligned}
\beta_{k,t}^{S_1} &= (-1)^k I(k < {}^{K^*}/_{K}) f_{1,t} + I(k > {}^{K^*}/_{K}) \beta_{k,0} \\
\beta_{k,t}^{S_2} &= (-1)^k I(k < {}^{K^*}/_{K}) f_{2,t} + I(k > {}^{K^*}/_{K}) \beta_{k,0} \\
\beta_{k,t}^{S_3} &= (-1)^k I(k < {}^{K^*}/_{2K}) f_{3,t} + (-1)^k I({}^{K^*}/_{2K} < k < {}^{K^*}/_{K}) f_{1,t} + I(k > {}^{K^*}/_{K}) \beta_{k,0}. \\
\beta_{k,t}^{S_4} &= I(k < {}^{K^*}/_{K}) \sum_{j \in \{1,4,5\}} l_{j,k} f_{j,t} \quad , \quad l_{j,t} \sim N(0,1).
\end{aligned}
$$

The scale of coefficients is manually adjusted to prevent explosive behavior and/or overwhelmingly high $R^{2\prime}$s. The most important transformation in that regard is a min-max normalization on the coefficient of y_{t-1} to prevent unit/explosive roots or simply persistence levels that would drive the true R^2 above its targeted range. Regarding the latter, I

[23] This partly motivates the creation of Generalized TVPs via Random Forest in Goulet Coulombe (2020b).

consider four different types of noise process. Three of them are homoscedastic and have a {Low, Medium, High} noise level. Those are calibrated so that R^2's are around 0.8, 0.5 and 0.3 for low, medium and high respectively. The last two noise processes are SV, which is the predominant departure from the normality of ϵ_t in applied macroeconomics. For better comparison with time-invariant volatility cases, those are "manually" forced (by a min-max normalization) to oscillate between a predetermined minimum and maximum. The first SV process is constrained within the Low and Medium noise level bounds. For the second, it is Low and High, making the volatility spread much higher than in the first SV process case.

Four estimators are considered: the standard TVP-BVAR with SV[24], the two-step Ridge Regression (2SRR), the Group Lasso Ridge Regression (GLRR) and the Generalized Reduced Rank Ridge Regression (GRRRR).[25] TVP-BVAR results are only obtained for $K = 6$ for obvious computational reasons. Performance is assessed using the mean absolute error (MAE) with respect to the true path. I then take the mean across 100 simulations for each setup. To make this multidimensional notation more compact, let us define the permutation $\mathcal{J} = \{K, K^*/K, \sigma_\epsilon, S_i\}$. I consider simulations $s = 1, ..., 50$ for all $\mathcal{J}$'s. Formally, for model m and setup $\mathcal{J}$, the reported performance metric is $\frac{1}{50}\sum_{s=1}^{50} MAE_{\mathcal{J}}^{s,m}$ where

$$MAE_{\mathcal{J}}^{s,m} = \frac{1}{K}\frac{1}{T}\sum_{k=1}^{K}\sum_{t=1}^{T}|\beta_{k,t}^{\mathcal{J},s} - \hat{\beta}_{k,t}^{\mathcal{J},s,m}|. \tag{1.20}$$

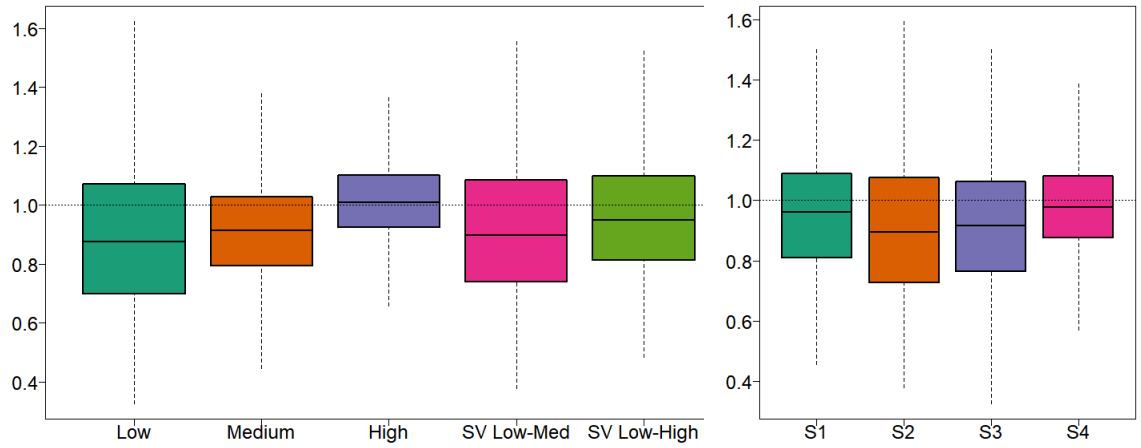
1.3.1. Results

The results for $T = 300$ are in Tables 2 to 5. With these simulations, I am mostly interested in verifying two things. First, I want to verify that 2SRR's performance is comparable to that of the BVAR for models' size that can be handled by the latter. Second, I want to demonstrate that additional shrinkage can help under DGPs that more of less fit the prior of reduced-rank and/or sparsity. To make the investigation of these two points visually easier by looking at the tables, the lowest MAE out of BVAR/2SRR for each setup is in blue while that of the best one out of all algorithms is in bold.

Overall, results for 2SRR and the BVAR are very similar and their relative performance depends on the specific setup. These two models are interesting to compare because they share the same prior for TVPs (no additional shrinkage) but address evolving residuals volatility differently. Namely, the BVAR models SV directly within the MCMC procedure

[24]For the TVP-BVAR, I use the `R` package by Fabian Krueger that implements Primiceri (2005)'s procedure (with the Del Negro and Primiceri (2015) correction), available here.

[25]The maximal number of factors for GRRRR is set to 5 and the chosen number of factors is updated adaptively in the EM procedure according to a share of variance criteria.



(a) By noise process

(b) By DGP

Figure 1: This figures summarizes tables 2 to 5 results comparing 2SRR and the BVAR when $K = 6$ and $T = 300$. The plotted quantity is the distribution of $MAE_J^{s,2SRR}/MAE_J^{s,BVAR}$ for different subsets of interest.

whereas 2SRR is a two-step GLS-like approach using a GARCH(1,1) model of the first step's residuals. Figure 1 summarizes results of the 2SRR/BVAR comparison by reporting boxplots showcasing the distribution of relative MAEs (2SRR/BVAR) for different subsets. Overall, 2SRR does marginally better in almost all cases. A lower noise level seems to help its cause. It is plausible that cross-validating λ as implemented by 2SRR plays a role (BVAR uses default values).[26]

In Table 2, where the DGP is the rather friendly cosine-based TVPs, it is observed that the BVAR will usually outperform 2SRR by a thin margin when the level of noise is high. The reverse is observed for low noise environment and results are mixed for the medium one. Results for the SV cases will the subject of its own discussion later. For $K = 6$, GLRR will marginally improve on 2SRR for most setups, especially those where 2SRR is already better than BVAR. In higher dimensions ($K = 20$ or $K = 100$), GLRR constantly improves on 2SRR (albeit minimally) whereas GRRRR can provide important gains (see the $K^*/K = 1$ block for instance) but is more vulnerable in the high noise environment.

In Table 3, where the DGP is the antagonistic structural break, 2SRR is clearly performing better than the BVAR, providing a smaller average MAE in 12 out of 15 cases for $K = 6$. Still for the small dimensional case, it is observed that GLRR can further reduce the MAE — albeit by a very small amount — in many instances. The same is true for GRRRR when all parameters vary ($K^*/K = 1$). For GRRRR, this observation additionally extends to $K =$

[26] Replacing the absolute distance by the squared distance in (1.20) produces similar looking boxplots as in Figure 1, with wider dispersion but a near-identical ranking of methods by simulations and noise processes.

Table 1: Average Computational Time in Seconds

	$T = 150$			$T = 300$			$T = 600$		
	$K = 6$	$K = 20$	$K = 100$	$K = 6$	$K = 20$	$K = 100$	$K = 6$	$K = 20$	$K = 100$
BVAR	219.3	–	–	513.7	–	–	1105.1	–	–
2SRR	0.3	0.9	2.7	1.0	3.1	12.4	4.8	14.1	69.5
GLRR	0.3	1.2	3.7	1.3	4.1	17.0	6.9	20.9	99.2
GRRRR	2.4	5.1	22.1	4.5	6.9	42.9	14.4	20.5	98.9

Notes: The average is taken over DGPs's, $^{K^*}/_K = 1$, noise processes, and all 100 runs. For all $\sim$RR models, this includes tuning cross-validating λ.

20, an environment where it is expected to thrive. Nonetheless, for setups where only a fraction of parameters vary, 2SRR and GLRR are the best alternatives for all K's.

In Table 4, where the DGP is a mix of trending coefficients and cosine ones, Table 4 reports very similar results to that of Simulation 1. 2SRR is better than BVAR except in the high noise setups, where the latter has a minor advantage. GLRR often marginally improves upon 2SRR whereas GRRRR's edge is more visible in low-noise and high-dimensional environments — factors being more precisely estimated with a large cross-section.

For the simulation in Table 5, a sophisticated mixture of TVP-friendly and -unfriendly parameters paths, BVAR does a better job than 2SRR for 8 out of 15 cases. The gains are, as before, quantitatively small. When 2SRR does better, gains are also negligible, suggesting that BVAR and 2SRR provide very similar results in this environment. When it comes to higher-dimensional setups ($K = 20$ or $K = 100$), GLRR emerges as the clear better option with (now familiar) marginal improvements with respect to 2SRR. This recurrent observation is potentially due to the iterative process producing a more precise $\hat{\Omega}_u$ when $\sigma^2_{u_k}$'s are heterogeneous whether sparsity is involved or not.

The results for $T = 150$ and $T = 600$ are in Tables 6 to 13. When T is reduced from 300 to 150, the performance of 2SRR relative to that of BVAR remains largely unchanged: both report very similar results. When bumping T to 600, overall performance of all estimators improves, but not by a gigantic leap. This is, of course, due to the fact that increasing T also brings up the number of effective regressors. BVAR has a small edge on Simulation 1 in Table 10 whereas 2SRR wins marginally for the more complicated Simulation 4 (Table 13). What is most noticeable from those simulations with a larger T is how much more frequently GLRR and especially GRRRR are preferred, especially in the medium- and high-dimensional cases. For instance, for the Cosine DGP (S_1) with $K \in \{20, 100\}$, GRRRR almost always deliver the lowest MAE, and sometimes by good margins (e.g., $\{S_1, {}^{K^*}/_K = 1, \sigma_\epsilon = \text{Low}\}$ for both K's). Similar behavior is observed for S_3 in almost all cases of $K = 20$. This noteworthy amelioration of GRRRR is intuitively attributable to factor loadings being more precisely estimated with a growing T. Thus, unlike 2SRR whose performance

is largely invariant to T by model design, algorithms incorporating more sophisticated shrinkage schemes may benefit from larger samples.

A pattern emerges across the four simulations: when SV is built in the DGP ($\sigma_{\epsilon,t}$ in tables), 2SRR either performs better or deliver roughly equivalent results to that of the BVAR. Indeed, with $T = 300$, for 17 out of 24 setups with SV-infused DGPs, 2SRR supplants BVAR. The wedge is sometimes small ($\{S_1, K^*/K = 0.2, \text{SV Low-Med}\}$,$\{S_4, K^*/K = 1, \text{both SV}\}$), sometimes large ($\{S_1, K^*/K = 1, \text{SV Low-High}\}$,$\{S_2, K^*/K = 0.5, \text{both SV}\}$). However, it fair to say that small gaps between 2SRR and BVAR performances are the norm rather than the exception. Nonetheless, these results suggest that 2SRR is not merely a suboptimal approximation to the BVAR in the wake of computational adversity. It is a viable statistical alternative with the additional benefit of being easy to compute and to tune.

Speaking of computations, Table 1 reports how computational time varies in K and T, and how 2SRR compares to a standard BVAR implementation. When T is 300 and K is 6, 2SRR takes one second while BVAR takes 513 seconds. When T increases to 600, BVAR takes over 1 000 seconds whereas 2SRR takes less than 5. $T = 300$ with $K = 300$ can be seen as a typical high-dimensional case. It takes 13 seconds to compute (and tune) 2SRRR. When T is reduced to 150, high-dimensional 2SRR runs in less than 3 seconds on average. Only when both T and K gets very large (by traditional macro data sets standards) do things become harder with 2SRR taking 69.5 seconds on average. By construction, GLRR takes marginally longer than 2SRR. Finally, by relying on an EM algorithm, GRRRR inevitably takes longer, yet remains highly manageable for very large models with many observations – taking a bit over 100 seconds.

1.4. Forecasting

In this section, I present results for a pseudo-out-of-sample forecasting experiment at the quarterly frequency using the dataset FRED-QD (McCracken and Ng, 2020). The latter is publicly available at the Federal Reserve of St-Louis's web site and contains 248 US macroeconomic and financial aggregates observed from 1960Q1. The forecasting targets are real GDP, Unemployment Rate (UR), CPI Inflation (INF), 1-Year Treasury Constant Maturity Rate (IR) and the difference between 10-year Treasury Constant Maturity rate and Federal funds rate (SPREAD). These series are representative macroeconomic indicators of the US economy which is based on Goulet Coulombe et al. (2019) exercise for many ML models, itself based on **?** and a whole literature of extensive horse races in the spirit of Stock and Watson (1998a). The series transformations to induce stationarity for predictors are indicated in McCracken and Ng (2020). For forecasting targets, GDP, CPI and UR are considered $I(1)$ and are first-differenced. For the first two, the natural logarithm is applied before differencing. IR and SPREAD are kept in "levels". Forecasting horizons are 1, 2, and

4 quarters. For variables in first differences (GDP, UR and CPI), average growth rates are targeted for horizons 2 and 4.

The pseudo-out-of-sample period starts in 2003Q1 and ends 2014Q4. I use expanding window estimation from 1961Q3. Models are estimated *and* tuned at each step. I use direct forecasts, meaning that $\hat{y}_{t+h}$ is obtained by fitting the model directly to y_{t+h} rather than iterating one-step ahead forecasts. Following standard practice in the literature, I evaluate the quality of point forecasts using the root Mean Square Prediction Error (MSPE). For the out-of-sample (OOS) forecasted values at time t of variable v made h steps ahead, I compute

$$RMSPE_{v,h,m} = \sqrt{\frac{1}{\#\text{OOS}} \sum_{t \in \text{OOS}} (y_t^v - \hat{y}_{t-h}^{v,h,m})^2}.$$

The standard Diebold and Mariano (2002) (DM) test procedure is used to compare the predictive accuracy of each model against the reference AR(2) model. $RMSPE$ is the most natural loss function given that all models are trained to minimize the squared loss in-sample.

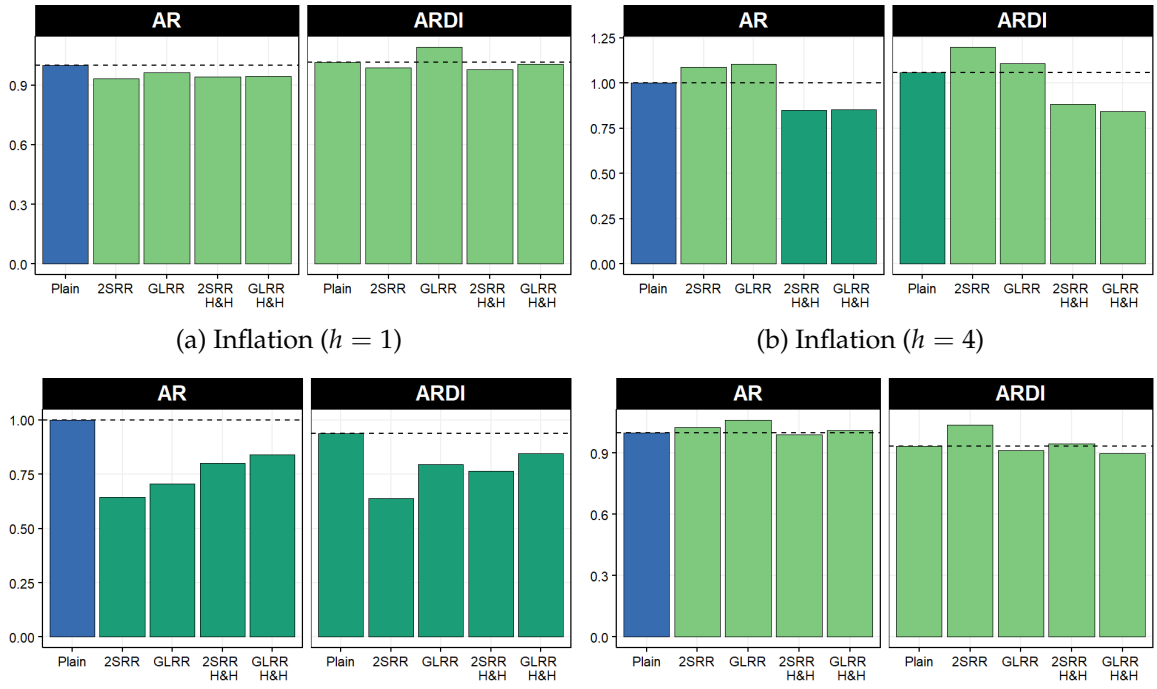
Three types of TVPs will be implemented: 2SRR (section 1.2.4), GLRR (section 1.2.5), GR-RRR (section 1.2.5). I consider augmenting four standard models with different methodologies proposed in this chapter. The first will be an **AR** with 2 lags. The second is the well-known Stock and Watson (2002) **ARDI** (Autoregressive Diffusion Index) with 2 factors and 2 lags for both the dependent variable and the factors. The third is a **VAR(5)** with 2 lags and the system is composed of the 5 forecasted series. Finally, I consider as a fourth model a **VAR(20)** with 2 lags in the spirit of Bańbura et al. (2010)'s medium VAR. Thus, there is a total of $4 \times 4 = 16$ models considered in the exercise. The BVAR used in section 1.3 is left out for computational reasons — models must be re-estimated every quarter for each target. Moreover, the focus of this section is rather single equation *direct* (as opposed to iterated) forecasting.

The first three constant coefficients models are estimated by OLS, which is standard practice. Since the constant parameters VAR(20) has 41 coefficients and around 200 observations, it is estimated with a ridge regression. Potential outliers are dealt with as in Goulet Coulombe et al. (2019) for Machine Learning models. If the forecasted values are outside of $[\bar{\boldsymbol{y}} + 2 * min(\boldsymbol{y} - \bar{\boldsymbol{y}}), \bar{\boldsymbol{y}} + 2 * max(\boldsymbol{y} - \bar{\boldsymbol{y}})]$, the forecast is discarded in favor of the constant parameters forecast.

1.4.1. Results

I report two sets of results. Table 14 corresponds exactly to what has been described beforehand. Table 15 gathers results where TVPs have been additionally shrunk to their constant

parameters counterparts by means of model averaging with equal weights. The virtues of this *Half & Half* strategy are two-fold. First, k-fold CV can be over-optimistic for horizons h>1 because of imminent serial correlation. Second, k-fold CV ranks potential λ's using the whole sample, whereas in the case of "forecasting", prediction always occurs at the boundary of the implicit kernel. In that region, the variance is mechanically higher and ensuing predictions could benefit from extra shrinkage. Shrinking to OLS in this crude and transparent fashion is a natural way to attempt getting even better forecasts.



(a) Inflation ($h = 1$) (b) Inflation ($h = 4$)

(c) Interest Rate ($h = 1$) (d) Interest Rate ($h = 4$)

Figure 2: A subset of $RMSPE_{v,h,m}/RMSPE_{v,h,\text{Plain AR(2)}}$'s (from Tables 14 and 15) for forecasting targets usually associated with the need for time variation. Blue is the benchmark AR with constant coefficients. Darker green means that the competing forecast rejects the null of a Diebold-Mariano test at least at the 10% level (with respect to the benchmark).

Overall, results are in line with evidence previously reported in the TVP literature: very limited improvements are observed for real activity variables (GDP, UR) whereas substantial gains are reported for INF and IR. For the latter, allowing for time variation in either AR or a compact factor model (ARDI) generate very competitive forecasts. For instance, ARDI-2SRR is the best model for IR with a reduction of 36% in RMPSE over the AR(2) benchmark which is strongly statistically significant. Still for IR, at horizon 2 quarters, iterating 2SRR to obtain GLRR generate sizable improvements for both AR and ARDIs. VAR(20) is largely inferior to alternatives in any of its forms. Two exceptions are IR forecasts at a one-year horizon where combining VAR(20) with GRRRR yields the best forecast

by a wide margin with improvement of 19% in RMSPE. VAR(20)-GRRRR also provide a very competitive forecast for IR at an horizon of one quarter. Finally, at horizon 1 quarter, any form of time variation (2SRR, GLRR, GRRRR) at least increases SPREAD's forecasting accuracy for for all models but the VAR(20). Precisely, it is a 16% reduction in RMSPE for AR, about 5% for ARDI and up to 14% in the VAR(5) case. For the latter, its combination with 2SRR provides the best forecast with a statistically significant improvement of 18% with respect to the AR(2) benchmark.

A notable absence from the relatively cheerful discussion above is inflation, which is the first (or second) variable one would think should benefit from time variation. It is clear that, in Table 14, any AR at horizon 1 profits rather timidly from it. A similar finding for Half & Half is reported in Table 15. What differs, however, are longer horizons results for INF. Indeed, mixing in additional shrinkage to OLS strongly helps results for those targets: every form of time variation now improves performance by a good margin. For instance, any time-varying ARs improves upon the constant benchmark by around 15%. It is now widely documented that inflation is better predicted by past values of itself and not much else – besides maybe for recessionary episodes (Kotchoni et al., 2019). Results of Table 15 comfortably stand within this paradigm except for the noticeable efforts from GLRR versions of both ARDI and VAR(20). While those are the best models, they are closely matched in performance by their AR counterparts. Nevertheless, it is noteworthy that this surge in performance mostly occurs for their sparse TVP versions, suggesting time variation is likely crucial for more sophisticated inflation forecasts not to be off the charts. Finally, additional shrinkage marginally improves GDP forecasting at the two longer horizons, with the Half & Half ARDI-GLRR providing the best forecasts.

To a large extent, forecasting results suggest that the three main algorithms presented in section 1.2 can procure important gains for forecasting targets that are frequently associated with the need for time variation. This subset is put on the spotlight by Figure 38. The gains for IR at $h = 1$ and INF at the one-year horizon are particularly visible. For those kinds of targets, it is observed that any form of time-variation will usually ameliorate the constant parameters benchmark, especially in the Half & Half case. This is convenient given how easy 2SRR, GLRR and GRRRR forecasts are to generate, in stark contrast to the typical Bayesian machinery.

1.5. Time-Varying Effects of Monetary Policy in Canada

VARs do not have a monopoly on the proliferation of parameters. Jordà (2005) local projections' – by running a separate regression for each horizon – are also densely parametrized. For that reason, constructing a large LP-based time-varying IRF via a MCMC procedure would either be burdensome or unfeasible. In this section, I demonstrate how 2SRR is up

to the task by estimating LPs chronicling the evolving effects of Canadian monetary policy over recent decades.

The use of 2SRR for estimation of LPs constitutes a very useful methodological development given how popular local projections have become over recent years. Particularly, it is appealing for researchers to identify shocks in a narrative fashion (for instance, à la Romer and Romer (2004)) and then use those in local projections to obtain their dynamic effects on the economy.[27] A next step is to wonder about the stability of the estimated relationship. In that line of thought, popular works include Auerbach and Gorodnichenko (2012a) and Ramey and Zubairy (2018a) for the study of state-dependent fiscal multipliers. To focus on long-run structural change rather than switching behavior, random walk TVPs are a natural choice and 2SRR, a convenient estimation approach.

In this application, I study the changing effects of monetary policy (MP) in Canada using the recently developed MP shocks series of Champagne and Sekkel (2018). The small open economy went through important structural change over the last 30–40 years. Most importantly, from a monetary policy standpoint, it became increasingly open (especially following NAFTA) and an inflation targeting regime (IT) was implemented in 1991 – a specific and publicly known date. Both are credible sources of structural change in the transmission of monetary policy. Champagne and Sekkel (2018) estimate a parsimonious VARs (4 variables) over two non-overlapping subsamples to check visually whether a break occurred in 1992 following the onset of IT. The reported evidence for a break is rather weak with GDP's response increasing slightly while that of inflation decreasing marginally. While the sample-splitting approach has many merits such as transparency and simplicity, there is arguably a lot it can miss. I go further by modeling the full evolution of their LP-based IRFs.

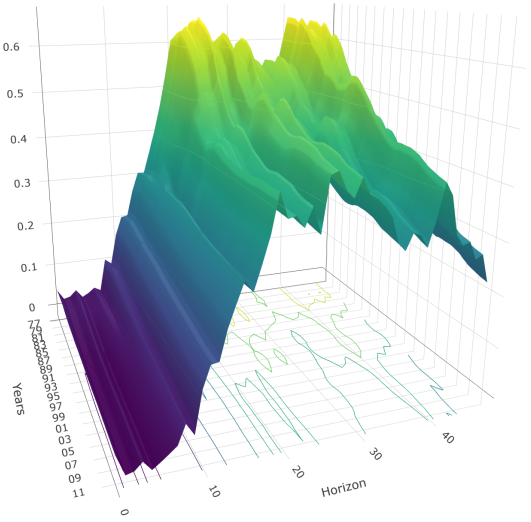
I use the same monthly Canada data set as in Champagne and Sekkel (2018) and the analysis spans from 1976 to 2015. The target variables are unemployment, CPI Inflation and GDP.[28] Their original specification includes 48 lags of the narrative monetary policy (MP) shock series which is constructed in the spirit of Romer and Romer (2004) and carefully adapted to the Canadian context.[29] Furthermore, their regression comprises 4 lags for the controls which are first differences of the log GDP, log inflation and log commodity prices.

[27] Another variant of that is the so-called local projections instrumental variable (LP-IV) where creating the shock itself is replaced by coming up with an IV (like in Ramey and Zubairy (2018a)).

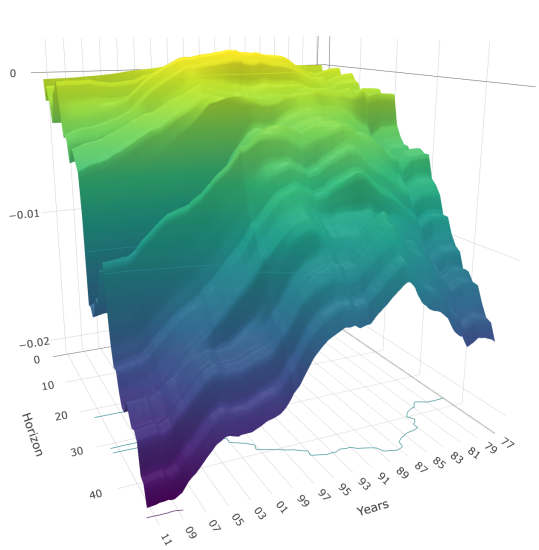
[28] For reference, the three time series being modeled and the shock series can be visualized in Figure 5. Notably, we can see that the conquest of Canadian inflation was done in two steps: reducing the mean from roughly 8% to 5% in the 1980s and from 5% to 2% in the early 1990s.

[29] For details regarding the construction of the crucial series — especially on how to account for the 1991 shift to IT, see Champagne and Sekkel (2018). Note that a positive shock means (unexpected) MP tightening.

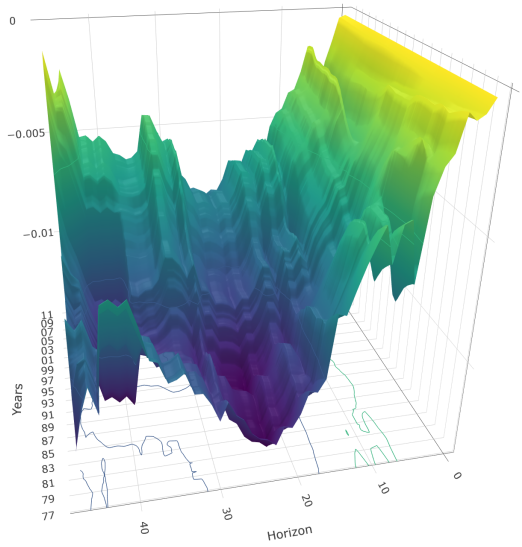
To certify that time-variation will not be found as a result of omitted variables, I increase the lag order from 4 to 6 months and augment the model with the USD/CAD exchange rate, exports, imports and CPI excluding Mortgage Interest Cost (MIC). In terms of TVP accounting, $\boldsymbol{X}$ contains 97 regressors (including the constant) and $\boldsymbol{Y}$ is 48. Thus, a single TV-LP is assembled from a staggering total of $97 \times 48 = 4\,656$ TVPs.



(a) Unemployment



(b) CPI Inflation



(c) GDP

Figure 3: Cumulative Time-Varying Effect of Monetary Policy Shocks. Rotations of 3D plots are hand-picked to highlight most salient features of each time-varying IRF. Interactive plots where the reader can manually explore different rotations are available here.

Figure 3 proposes clear answers to the evolving effect of monetary policy on the economy.

Generally, short- and medium-run effects ($h < 24$ months) have been much more stable than longer-run ones. This is especially true of unemployment which exhibits a strikingly homogeneous response (through time) for the first year and half after the shock. GDP's response follows a similar predicament, but was marginally steeper before the 2000s. When it comes to inflation, its usual long response lag has mildly shortened up in the 2000s. At a horizon of 24 months, the effect of a positive one standard deviation shock was -0.3% from 1976 to the late 1990s, then slowly increased (in absolute terms) to nearly double at -0.6%. Overall, results for horizons up to 18 months suggest that the ability of the central bank to (relatively) rapidly impact inflation has increased, while that of GDP has decreased and that of unemployment remained stable.

Given the long lags of monetary policy, most of the relevant action from an economic standpoint is also where most time variation is found: from 1.5 to 4 years after the shock. Regarding GDP and unemployment, the cumulative long-run effects of MP shocks have substantially shrunk over the sample period. For unemployment, the decrease from a 0.6 to 0.4 unemployment percentage points effect mostly occurred throughout the 1980s, and stabilized at 0.4 thereafter. For GDP, both its peak effect (at around $h = 24$ months) and the long-run one shrunk from the 1990s onward. Both quantities are about twice smaller around 2011 than they were in 1991. The long-run cumulative effect on inflation follows a distinctively different route: it has considerably expanded starting from the late 1980s. The overall effect on CPI, four years after impact, doubled from -1% in 1987 to -2% in 2011.

An important question is what happens to β_t around 1992, after the onset of IT. Figure 6 (in the appendix) reports $\beta_t^{\text{2SRR}} - \beta^{\text{OLS}}$ for the dynamic effect of MP shocks on the three variables. It is found that the response of inflation (in absolute terms) is much larger at the end of the sample than what constant coefficients would suggest. This is especially true at the 24 months horizon. It is quite clear that, for all horizons, the effect of MP shocks on inflation starts increasing in the years following the implementation of IT. The vanishing effect on GDP starting from the 1990s could be consistent with an increased openness of the economy limiting the central bank's grip on economic activity. The story is, however, different for unemployment. The downward trend in MP shocks' impact seems to have started at least since the 1970s and slowed down in recent years. More generally, it is interesting to note that those results are consistent with a flattening Phillips' curve – as reported in Blanchard et al. (2015) for Canada and many other countries. The shrinking responses of GDP and unemployment leave room for expectations to act as the main channel through which MP shocks (eventually) impact the price level.

A crucial advantage of 2SRR is that it takes something complicated and makes it easy. Thus, one could (rather ambitiously) hope for TVPs to enter the traditional empirical

macro robustness checks arsenal, and stand intrepidly next to ad hoc sample-splitting tests. Using the latter strategy paired with a myriad of standard identification schemes (Ramey, 2016), Barakchian and Crowe (2013) provide counterintuitive results (a price puzzle and MP tightening increases GDP) for post-1988 US data. Cloyne and Hürtgen (2016) and Champagne and Sekkel (2018) report less economics-contorting findings for the UK and Canada: GDP response increases marginally after IT and that of inflation shrinks. 2SRR-LPs signs and magnitudes are consistent with those reported in Champagne and Sekkel (2018). Moreover, Figure 6 does not suggest the occurrence of a structural break in 1992, which is in line with most of the international evidence on IT implementation. Nonetheless, at least for GDP and inflation, 2SRR-LPs' results point to a drastic change in coefficients' trending behavior, a subtle phenomenon which effectively stays under the radar of simpler approaches. Particularly, when looking at various $\beta_{t,h}^{\text{2SRR}} - \beta_h^{\text{OLS}}$ for inflation in Figure 6, it is self-evident why mere splitting of the sample would not find any significant change. Additionally, the staircase-like trajectory of Canadian inflation in the 1980s (Figure 5c) is visually supportive of a TVP approach for the intercept, and cast strong doubts about any approach assuming only two regimes.

Unlike most of their predecessors, results presented in this section rely on an approach that is jointly flexible **(i)** in the specification of dynamics by using LPs, **(ii)** in the information set by allowing for many controls and **(iii)** in the time variation by fully modeling β_t's path. The 2SRR-based LPs display that while the cumulative effect of MP shocks became more muted for real activity variables, it has increased for inflation. This suggests that stabilizing Canadian inflation is now much less costly (in terms of unemployment/GDP variability) than it used to be.

1.6. Conclusion

I provide a new framework to estimate TVP models with potentially evolving volatility of shocks. It is conceptually enlightening and computationally very fast. Moreover, seeing such models as ridge regressions suggest a simple way to tune the amount of time variation, a neuralgic quantity. The approach is easily extendable to have additional shrinkage schemes like sparse TVPs or reduced-rank restrictions. The proposed variants of the methodology are very competitive against the standard Bayesian TVP-VAR in simulations. Furthermore, they improve forecasts against standard forecasting benchmarks for variables usually associated with the need for time variation (US inflation and interest rates). Finally, I apply the tool to estimate time-varying IRFs via local projections. The large specification necessary to characterize adequately the evolution of monetary policy in Canada rendered this application likely unfeasible without the newly developed tools. I report that monetary policy shocks long-run impact on the price level increased substantially starting from the early 1990s (onset of inflation targeting), whereas the effects on real activity be-

came milder. This finding is consistent with the hypothesized flattening of the Phillips' curve in advanced economies.

1.7. Appendix

1.7.1. Details of GLRR

To begin with, the penalty part of (1.12) in summation notation is

$$\sum_{k=1}^{K} \frac{1}{\sigma_{u_k}^2} \sum_{t=1}^{T} u_{k,t}^2 + \xi \sum_{k=1}^{K} |\sigma_{u_k}|.$$

The E-step of the procedure provides a formula for σ_{u_j} in terms of u's. Plugging it in gives

$$\sum_{k=1}^{K} \frac{1}{\sigma_{u_k}^2} \sum_{t=1}^{T} u_{k,t}^2 + \xi \sum_{k=1}^{K} (\sum_{t=1}^{T} u_{k,t}^2)^{\frac{1}{2}}.$$

which is just a Group Lasso penalty with an additional Ridge penalty for each individual coefficients. Hence, classifying parameters into TVP or non-TVP categories is equivalent to group selection of regressors where each k of the K groups is defined as $\{Z_{t,k,\tau}\}_{\tau=1}^{\tau=T}$. If we want a parameter to be constant, we trivially have to drop block-wise its respective basis expansion regressors and only keep $\beta_{0,k}$ in the model.

This penalty can be obtained by iterating what we already have. Grandvalet (1998) shows that the Lasso solution can be obtained by iterative Adaptive Ridge. Frommlet and Nuel (2016) and Liu and Li (2014) extend his results to obtain l_0 regularization without the computational burden associated with this type of regularization. Frommlet and Nuel (2016) also argue in favor of a slightly modified version of Grandvalet (1998)'s algorithm which I first review before turning to the final GLRR problem.

To implement Lasso by Adaptive Ridge, we have at iteration i,

$$\boldsymbol{b}_i = \arg\min_{\boldsymbol{b}} \sum_{t=1}^{T} (y_t - X_t \boldsymbol{b})^2 + \lambda \sum_{=1}^{J} w_j b_j^2$$
$$w_{i+1,j} = \frac{1}{(b_{i,j} + \delta)^2}$$

where $\delta > 0$ is small value for numerical stability and we set $w_{j,0} = 1 \quad \forall j$. To get some intuition on why this works, it is worth looking at the penalty part of the problem in the final algorithm step:

$$\lambda \sum_{j=1}^{J} \frac{b_j^2}{|\hat{b}_j| + \delta} \approx \lambda \sum_{j=1}^{J} |b_j|.$$

Liu and Li (2014) show that this qualifies as a proper EM algorithm (each step improves the likelihood). Thus, we can expect it to inherit traditional convergence properties.

Building Iterative Weights for GLRR

The above methodology can be adapted for a case which is substantially more complicated. The complications are twofold. First, we are doing Group Lasso rather than plain Lasso. Second, the individual ridge penalty must be maintained on top of the Group Lasso penalty. I devise a simple algorithm that will split the original Ridge penalty into two parts, one that we will keep as is and one that will be iterated. The first is the 2SRR part and the second implements Group-Lasso.

Let us first focus on the Group penalty and display why iterating the Ridge solution with updating weights converges to be equivalent to Group Lasso. In the last step of the algorithm, we have

$$\xi \sum_{k=1}^{K} \frac{1}{\hat{\sigma}_{u_k}} \sum_{t=1}^{T} u_{k,t}^2 \approx \xi \sum_{k=1}^{K} (\sum_{t=1}^{T} u_{k,t}^2)^{\frac{1}{2}}$$

where $\hat{\sigma}_{u_k} = (\sum_{t=1}^{T} u_{k,t}^2)^{\frac{1}{2}}$. The two penalties must be combined in a single penalizing weight that enters the closed-form solution. I split the original penalty into two parts, one that will remain as such and one that will be iterated to generate group selection. A useful observation is the following. For a given iteration i,

$$\lambda \sum_{k=1}^{K} \frac{1}{\sigma_{u_k}^2} \sum_{t=1}^{T} u_{k,t}^2 + \xi \sum_{k=1}^{K} \frac{1}{\sigma_{u_k}^{(i)}} \sum_{t=1}^{T} u_{k,t}^2$$

can be re-arranged as

$$\sum_{k=1}^{K} \left[\frac{\lambda}{\sigma_{u_k}^2} + \frac{\xi}{\sigma_{u_k}^{(i)}} \right] \sum_{t=1}^{T} u_{k,t}^2.$$

To make this more illuminating, define $\alpha = \frac{\lambda}{\lambda + \xi}$ and $\tilde{\lambda} = (\lambda + \xi)$. We now have

$$\tilde{\lambda} \sum_{k=1}^{K} \left[\alpha \frac{1}{\sigma_{u_k}^2} + (1 - \alpha) \frac{1}{\sigma_{u_k}^{(i)}} \right] \sum_{t=1}^{T} u_{k,t}^2.$$

where $\alpha \in (0, 1)$ is a tuning parameter controlling how the original ridge penalty is split between smoothness and group-wise sparsity. It is now easy to plug this into the closed-form formula: stack $\lambda_{u_k}^{(i)} = \tilde{\lambda} \left[\alpha \frac{1}{\sigma_{u_k}^2} + (1 - \alpha) \frac{1}{\sigma_{u_k}^{(i-1)}} \right]$ on the diagonal of $\Omega_{u_i}^{-1}$ at iteration i in 1.2.4. The reader is now referred to the main text (section 1.2.5) for the benchmark algorithm that uses these derivations to implement GLRR.

Credible regions

In the homoscedastic case, we need to obtain

$$V_{\beta} = \boldsymbol{C}_*(\boldsymbol{Z}_*'\boldsymbol{Z}_* + \boldsymbol{\Omega}_{\theta_*}^{-1})^{-1}\boldsymbol{C}_*'\hat{\sigma}_{\epsilon}^2.$$

where the C_*, $\boldsymbol{Z}_*$ and Ω_{θ_*} are the part of the corresponding matrices left after leaving out the basis expansion parts that correspond to the selected constant parameters. The $*$ versions should be much smaller than the original one especially in a high-dimensional model. Since heteroscedasticity is incorporated in a GLS fashion, credible regions can be obtained by using the formula above with the properly re-weighted data matrix $\boldsymbol{Z}^*$. These bands take the model selection event as given.

1.7.2. Multivariate Extension to GRRRR

Dense TVPs as proposed (among others) by Stevanovic (2016) implement a factor structure for parameters of a whole VAR system rather than a single equation. If time-variation is indeed similar for all equations, we can decrease estimation variance significantly by pooling all parameters of the system in a single factor model. First, the factors are better estimated as the number of series increase. Second, the estimated factors are less prone to overfit because they now target M series rather than a single one.[30] The likely case where r is smaller than M (and P not incredibly big) yields a models that will have more observations than parameters, in contrast to everything so far considered in this chapter. I briefly describe how to modify Algorithm 3 to obtain Multivariate GRRRR (MV-GRRRR) estimates.

Starting values for the algorithm below can be obtained from the multivariate RR of section (1.2.4). This is done by first re-arranging elements of $\hat{\boldsymbol{\Theta}}$ into $\mathcal{U} = [\boldsymbol{U}_1 \quad \dots \quad \boldsymbol{U}_M]$ and then running PCA on $\mathcal{U}$. Then, the MV-GRRRR solution can be obtained by alternating the following steps.

1. Given Λ, we can solve

$$\min_{f,b_0} \left(vec(\boldsymbol{Y}) - (I_M \otimes \boldsymbol{X})b_0 - \boldsymbol{Z_M}^{\Lambda}\boldsymbol{f}\right)' \Omega_{\epsilon_M}^{-1} \left(vec(\boldsymbol{Y}) - (I_M \otimes \boldsymbol{X})b_0 - \boldsymbol{Z_M}^{\Lambda}\boldsymbol{f}\right) + \boldsymbol{f'f} \tag{1.21}$$

where $\boldsymbol{Z_M}^{\Lambda}$ stacks row-wise all the $\boldsymbol{Z}(I_T \otimes \Lambda_m)$ from $m = 1$ to $m = M$. That is, we

[30] This is the kind of regularization being used for linear models in Carriero et al. (2011). However, for MV-GRRRR, the reduced-rank matrix is organized differently and the underlying factors have a different interpretation.

have the $TM \times Tr$ matrix

$$\boldsymbol{Z_M}^{\Lambda} = \begin{bmatrix} \boldsymbol{Z}(I_T \otimes \Lambda_1) \\ \boldsymbol{Z}(I_T \otimes \Lambda_2) \\ \vdots \\ \boldsymbol{Z}(I_T \otimes \Lambda_M) \end{bmatrix}$$

as the regressor matrix. Λ_m is a sub-matrix of Λ that contains the loadings for parameters of equation m. Also, $b_0 = vec(B_0)$ where B_0 is the matrix that corresponds to the multivariate equivalent of β_0. Unlike a standard multivariate model like a VAR, here, we cannot estimate each equation separately because the $\boldsymbol{f}$ is common across equations.

2. The loadings updating step is

$$\min_{l,b_0} \left(vec(\boldsymbol{Y}) - (I_M \otimes \boldsymbol{X})b_0 - \boldsymbol{Z_M^F} l\right)' \Omega_{\epsilon_M}^{-1} \left(vec(\boldsymbol{Y}) - (I_M \otimes \boldsymbol{X})b_0 - \boldsymbol{Z_M^F} l\right) + \xi \|l\|_1 \tag{1.22}$$

where $\boldsymbol{Z_M^F} = (I_M \otimes \boldsymbol{Z}(\boldsymbol{F'} \otimes I_K))$. This is just a Lasso regression. The Kronecker structure allows for these Lasso regressions to be ran separately.

As in Bai and Ng (2017) for the estimation of regularized factor models, there is orthogonalization step needed between each of these steps to guarantee identification.

Note that if $MT > rT + MK$, which is somewhat likely, we have more observations than parameters in step 1. This means standard Ridge regularization is *not* necessary for the inversion of covariance matrix of regressors.[31] Nonetheless, the ridge smoothness prior will still prove useful but can be applied in a much less aggressive way.

An interesting connection occurs in the MV-GRRRR case: the time-varying parameter model with a factor structure in parameters can also be seen as a dynamic factor model with deterministically time-varying loadings. By the latter, I mean that loadings change through time because they are interacted with a known set of (random) variables X_t. This is a more general version of Kelly et al. (2017) Instrumented PCA used to estimate a typical asset-pricing factor model. Formally, this means that the factor TVP model

$$Y_t = X_t \Lambda F_t + \epsilon_t, \qquad F_t = F_{t-1} + u_t$$

[31] This also means that it is now computationally more efficient to solve the primal Ridge problem.

can be rewritten as

$$Y_t = \Lambda_t F_t + \epsilon_t, \qquad F_t = F_{t-1} + u_t, \qquad \Lambda_t = X_t \Lambda \tag{1.23}$$

which is the so-called Instrumented PCA estimator if we drop the law of motion for F_t. An important additional distinction is that Kelly et al. (2017) consider cases where the number of instruments is smaller than the size of the cross-section. Here, with the instruments being X_t, there is by construction more instruments than the size of the cross-section. Nevertheless, the analogy to the factor model is conceptually useful and can point to further improvements of TVP models inspired by advances in empirical asset pricing research.

1.7.3. Simple GRRRR Example with $r = 1$

While Kronecker product operations may seem obscure, they are the generalization of something that much more intuitive: the special case of one factor model ($r = 1$). I present here the simpler model when parameters vary according to a single latent source of time-variation. For convenience, I drop evolving volatility and use summation notation. The problem reduces to

$$\min_{l,f,\beta_0} \sum_{t=1}^{T} \left(y_t - X_t \beta_0 - \sum_{k=1}^{K} l_k f_t X_k \right)^2 + \sum_{t=1}^{T} f_t^2 + \xi \sum_{k=1}^{K} |l_k| \tag{1.24}$$

which can trivially rewritten as

$$\min_{l,f,\beta_0} \sum_{t=1}^{T} \left(y_t - X_t \beta_0 - f_t \sum_{k=1}^{K} l_k X_{k,t} \right)^2 + \sum_{t=1}^{T} f_t^2 + \xi \sum_{k=1}^{K} |l_k|. \tag{1.25}$$

and this model can be estimated by splitting it two problems. The two steps are

1. Given the l vector, we run the TVP regression

$$\min_{f,\beta_0} \sum_{t=1}^{T} \left(y_t - X_t \beta_0 - \bar{X}_t f_t \right)^2 + \sum_{t=1}^{T} f_t^2.$$

   where $\bar{X}_t \equiv \sum_{k=1}^{K} l_k X_{k,t}$. Hence, the new regressors are just a linear combination of original regressors.

2. Given f, the second step is the Lasso regression (or OLS/Ridge if we prefer)

$$\min_{l,\beta_0} \sum_{t=1}^{T} \left(y_t - X_t \beta_0 - \sum_{k=1}^{K} l_k X^f{}_{k,t} \right)^2 + \xi \sum_{k=1}^{K} |l_k|.$$

where the K new regressors are ${X^f}_{k,t} \equiv f_t X_{k,t}$.

1.7.4. *Tables*

Table 2: Results for Simulation 1 (Cosine) and $T = 300$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.128	0.110	**0.097**	0.136	-	0.115	**0.095**	0.114	-	0.163	**0.160**	0.200
σ_ϵ = Medium	**0.159**	0.165	0.163	0.193	-	0.165	**0.161**	0.169	-	0.197	**0.192**	0.314
σ_ϵ = High	**0.228**	0.245	0.244	0.271	-	0.262	**0.262**	0.269	-	0.320	**0.316**	0.580
$\sigma_{\epsilon,t}$ = SV Low-Med	0.129	0.121	**0.110**	0.145	-	0.131	**0.120**	0.132	-	0.168	**0.166**	0.242
$\sigma_{\epsilon,t}$ = SV Low-High	**0.143**	0.151	0.152	0.174	-	0.159	**0.158**	0.175	-	0.189	**0.189**	0.293
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.169	0.130	**0.120**	0.144	-	0.150	0.135	**0.129**	-	0.256	**0.256**	0.263
σ_ϵ = Medium	0.224	0.207	**0.206**	0.247	-	0.227	0.224	**0.221**	-	0.283	**0.278**	0.395
σ_ϵ = High	**0.274**	0.291	0.292	0.330	-	0.314	**0.311**	0.316	-	0.371	**0.365**	0.700
$\sigma_{\epsilon,t}$ = SV Low-Med	0.186	0.147	**0.138**	0.184	-	0.171	0.162	**0.158**	-	**0.259**	0.260	0.303
$\sigma_{\epsilon,t}$ = SV Low-High	0.211	**0.189**	0.189	0.229	-	0.216	**0.214**	0.225	-	**0.273**	0.274	0.361
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	**0.134**	0.149	0.152	0.157	-	0.185	0.191	**0.141**	-	0.367	0.370	**0.284**
σ_ϵ = Medium	0.302	**0.242**	0.247	0.282	-	0.278	0.289	**0.252**	-	0.389	**0.388**	0.467
σ_ϵ = High	**0.337**	0.355	0.360	0.376	-	**0.388**	0.389	0.391	-	0.454	**0.451**	0.766
$\sigma_{\epsilon,t}$ = SV Low-Med	0.171	**0.168**	0.172	0.180	-	0.208	0.215	**0.156**	-	0.380	0.381	**0.351**
$\sigma_{\epsilon,t}$ = SV Low-High	0.262	**0.222**	0.232	0.268	-	0.262	0.274	**0.261**	-	0.383	0.382	**0.368**

Notes: This table reports the average MAE of estimated β_t's for various models. The number in bold is the lowest MAE of all models for a given setup. The number in blue is the lowest MAE between BVAR and 2SRR for a given setup.

Table 3: Results for Simulation 2 (Break) and $T = 300$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.154	0.113	**0.098**	0.146	-	0.149	**0.141**	0.191	-	**0.331**	0.337	0.487
σ_ϵ = Medium	0.216	0.176	**0.165**	0.296	-	**0.249**	0.256	0.294	-	0.587	**0.578**	1.165
σ_ϵ = High	0.295	**0.292**	0.296	0.412	-	**0.473**	0.480	0.498	-	1.267	**1.236**	2.484
$\sigma_{\epsilon,t}$ = SV Low-Med	0.171	0.126	**0.114**	0.180	-	0.175	**0.169**	0.218	-	**0.413**	0.414	0.708
$\sigma_{\epsilon,t}$ = SV Low-High	0.188	0.159	**0.152**	0.249	-	**0.232**	0.248	0.287	-	**0.522**	0.546	0.984
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.154	0.141	**0.130**	0.137	-	0.184	**0.180**	0.232	-	**0.396**	0.432	0.656
σ_ϵ = Medium	0.316	0.207	**0.204**	0.296	-	**0.295**	0.318	0.362	-	**0.633**	0.640	1.064
σ_ϵ = High	0.370	**0.335**	0.348	0.465	-	**0.513**	0.527	0.542	-	1.291	**1.277**	2.674
$\sigma_{\epsilon,t}$ = SV Low-Med	0.197	0.156	**0.148**	0.156	-	**0.215**	0.220	0.275	-	**0.469**	0.487	0.769
$\sigma_{\epsilon,t}$ = SV Low-High	0.242	0.193	**0.190**	0.298	-	**0.284**	0.303	0.389	-	**0.587**	0.620	1.035
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	**0.143**	0.174	0.183	0.149	-	0.232	0.430	**0.188**	-	**0.513**	0.620	0.633
σ_ϵ = Medium	0.308	0.254	0.343	**0.252**	-	0.349	0.493	**0.308**	-	**0.768**	0.786	1.149
σ_ϵ = High	0.506	**0.414**	0.499	0.447	-	0.612	**0.607**	0.690	-	1.437	**1.394**	2.697
$\sigma_{\epsilon,t}$ = SV Low-Med	0.168	0.195	0.205	**0.165**	-	0.258	0.448	**0.220**	-	**0.571**	0.663	0.758
$\sigma_{\epsilon,t}$ = SV Low-High	**0.205**	0.231	0.294	0.231	-	0.340	0.481	**0.334**	-	**0.695**	0.726	1.054

Notes: see Table 2.

Table 4: Results for Simulation 3 (Trend and Cosine) and $T = 300$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	**0.079**	0.088	0.086	0.098	-	0.135	0.126	**0.114**	-	**0.258**	0.259	0.420
σ_ϵ = Medium	0.179	0.142	**0.135**	0.191	-	**0.202**	0.204	0.203	-	0.348	**0.339**	0.684
σ_ϵ = High	**0.218**	0.222	0.223	0.282	-	0.290	**0.290**	0.355	-	0.655	**0.638**	1.313
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.093**	0.103	0.094	0.119	-	0.156	0.153	**0.138**	-	0.281	**0.278**	0.498
$\sigma_{\epsilon,t}$ = SV Low-High	0.137	0.129	**0.121**	0.165	-	**0.196**	0.199	0.222	-	0.340	**0.339**	0.602
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.102	**0.071**	0.082	0.106	-	**0.116**	0.116	0.126	-	**0.232**	0.235	0.413
σ_ϵ = Medium	0.131	**0.119**	0.122	0.176	-	**0.176**	0.180	0.193	-	0.324	**0.320**	0.722
σ_ϵ = High	**0.172**	0.185	0.186	0.234	-	0.274	**0.273**	0.329	-	0.646	**0.634**	1.382
$\sigma_{\epsilon,t}$ = SV Low-Med	0.115	**0.085**	0.089	0.116	-	**0.133**	0.138	0.146	-	0.264	**0.262**	0.433
$\sigma_{\epsilon,t}$ = SV Low-High	0.120	**0.105**	0.106	0.143	-	**0.166**	0.171	0.195	-	**0.316**	0.319	0.611
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	0.067	**0.047**	0.050	0.054	-	0.075	0.091	**0.064**	-	**0.177**	0.186	0.319
σ_ϵ = Medium	0.086	**0.080**	0.087	0.097	-	0.128	0.139	**0.127**	-	**0.302**	0.304	0.607
σ_ϵ = High	**0.131**	0.139	0.139	0.179	-	**0.238**	0.246	0.244	-	0.638	**0.628**	1.470
$\sigma_{\epsilon,t}$ = SV Low-Med	0.067	0.056	0.059	**0.052**	-	0.088	0.102	**0.081**	-	**0.211**	0.221	0.402
$\sigma_{\epsilon,t}$ = SV Low-High	0.071	**0.066**	0.075	0.085	-	**0.115**	0.131	0.137	-	**0.281**	0.289	0.501

Notes: see Table 2.

Table 5: Results for Simulation 4 (Mixture) and $T = 300$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.054	0.054	**0.051**	0.066	-	0.068	**0.066**	0.073	-	0.150	**0.147**	0.269
σ_ϵ = Medium	**0.079**	0.082	0.080	0.100	-	0.115	**0.113**	0.122	-	0.283	**0.278**	0.561
σ_ϵ = High	**0.126**	0.138	0.136	0.160	-	0.232	**0.228**	0.246	-	0.628	**0.613**	1.326
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.058**	0.062	0.060	0.074	-	0.078	**0.077**	0.080	-	0.185	**0.183**	0.338
$\sigma_{\epsilon,t}$ = SV Low-High	**0.065**	0.073	0.077	0.097	-	**0.104**	0.106	0.133	-	**0.258**	0.263	0.487
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.076	0.066	**0.062**	0.082	-	0.086	**0.085**	0.090	-	0.169	**0.167**	0.289
σ_ϵ = Medium	**0.095**	0.097	0.096	0.124	-	0.130	**0.127**	0.135	-	0.294	**0.290**	0.571
σ_ϵ = High	**0.138**	0.151	0.149	0.183	-	0.238	**0.234**	0.254	-	0.633	**0.623**	1.304
$\sigma_{\epsilon,t}$ = SV Low-Med	0.078	0.075	**0.072**	0.090	-	0.097	**0.096**	0.099	-	0.204	**0.200**	0.323
$\sigma_{\epsilon,t}$ = SV Low-High	**0.085**	0.089	0.091	0.111	-	**0.120**	0.123	0.151	-	**0.268**	0.271	0.475
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	0.098	**0.075**	0.078	0.102	-	**0.110**	0.114	0.116	-	0.201	**0.198**	0.358
σ_ϵ = Medium	0.121	**0.118**	0.119	0.150	-	0.155	**0.154**	0.163	-	0.309	**0.306**	0.629
σ_ϵ = High	**0.161**	0.176	0.177	0.229	-	0.264	**0.257**	0.288	-	0.641	**0.635**	1.403
$\sigma_{\epsilon,t}$ = SV Low-Med	0.107	**0.087**	0.087	0.108	-	**0.121**	0.122	0.126	-	0.230	**0.225**	0.374
$\sigma_{\epsilon,t}$ = SV Low-High	0.112	**0.105**	0.109	0.132	-	**0.145**	0.147	0.172	-	**0.287**	0.289	0.567

Notes: see Table 2.

Table 6: Results for Simulation 1 (Cosine) and $T = 150$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.136	0.119	**0.109**	0.154	-	0.139	**0.133**	0.159	-	**0.244**	0.252	0.294
σ_ϵ = Medium	**0.182**	0.186	0.183	0.214	-	0.212	**0.205**	0.257	-	**0.321**	0.328	0.337
σ_ϵ = High	**0.337**	0.354	0.344	0.377	-	**0.393**	0.396	0.568	-	**0.569**	0.584	0.617
$\sigma_{\epsilon,t}$ = SV Low-Med	0.150	**0.148**	0.149	0.183	-	0.162	**0.158**	0.208	-	**0.273**	0.277	0.297
$\sigma_{\epsilon,t}$ = SV Low-High	**0.200**	0.212	0.216	0.252	-	0.244	**0.242**	0.338	-	**0.376**	0.385	0.385
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.200	0.142	**0.136**	0.187	-	0.170	**0.170**	0.215	-	0.358	0.368	**0.347**
σ_ϵ = Medium	0.228	**0.219**	0.222	0.291	-	**0.252**	0.256	0.305	-	0.418	0.432	**0.406**
σ_ϵ = High	**0.360**	0.383	0.374	0.396	-	**0.432**	0.436	0.591	-	**0.624**	0.633	0.712
$\sigma_{\epsilon,t}$ = SV Low-Med	0.203	0.175	**0.173**	0.227	-	**0.209**	0.212	0.253	-	0.387	0.400	**0.365**
$\sigma_{\epsilon,t}$ = SV Low-High	**0.240**	0.243	0.248	0.285	-	**0.287**	0.292	0.397	-	0.465	0.485	**0.436**
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	0.262	**0.158**	0.161	0.189	-	**0.206**	0.225	0.231	-	0.492	0.511	**0.429**
σ_ϵ = Medium	0.284	**0.239**	0.247	0.270	-	**0.302**	0.316	0.352	-	0.532	0.547	**0.500**
σ_ϵ = High	**0.390**	0.421	0.431	0.433	-	**0.477**	0.481	0.678	-	**0.717**	0.737	0.722
$\sigma_{\epsilon,t}$ = SV Low-Med	0.270	**0.191**	0.197	0.241	-	**0.253**	0.270	0.269	-	0.508	0.526	**0.453**
$\sigma_{\epsilon,t}$ = SV Low-High	0.295	**0.270**	0.282	0.324	-	**0.338**	0.356	0.454	-	0.572	0.590	**0.546**

Notes: see Table 2.

Table 7: Results for Simulation 2 (Break) and $T = 150$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.083	0.085	**0.082**	0.110	-	0.162	**0.155**	0.175	-	**0.534**	0.547	0.611
σ_ϵ = Medium	**0.154**	0.162	0.160	0.188	-	0.330	**0.312**	0.449	-	**1.070**	1.073	1.283
σ_ϵ = High	**0.380**	0.396	0.399	0.472	-	0.745	**0.727**	1.059	-	**2.364**	2.430	2.378
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.115**	0.125	0.126	0.165	-	0.229	**0.214**	0.304	-	**0.766**	0.784	0.894
$\sigma_{\epsilon,t}$ = SV Low-High	**0.196**	0.213	0.216	0.268	-	0.387	**0.380**	0.592	-	1.341	1.370	**1.305**
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.084	0.084	**0.083**	0.105	-	0.164	**0.157**	0.178	-	**0.536**	0.544	0.652
σ_ϵ = Medium	**0.153**	0.162	0.162	0.205	-	0.336	**0.317**	0.445	-	**1.080**	1.081	1.180
σ_ϵ = High	**0.382**	0.385	0.385	0.502	-	0.743	**0.731**	1.021	-	**2.364**	2.426	2.411
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.113**	0.121	0.120	0.136	-	0.225	**0.213**	0.305	-	**0.770**	0.792	0.905
$\sigma_{\epsilon,t}$ = SV Low-High	**0.198**	0.215	0.220	0.240	-	0.382	**0.376**	0.580	-	**1.341**	1.385	1.379
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	**0.085**	0.089	0.086	0.099	-	0.163	**0.157**	0.167	-	0.534	**0.532**	0.717
σ_ϵ = Medium	**0.158**	0.173	0.170	0.203	-	0.330	**0.320**	0.460	-	**1.063**	1.073	1.264
σ_ϵ = High	**0.383**	0.426	0.419	0.460	-	0.751	**0.736**	0.953	-	**2.353**	2.408	2.613
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.114**	0.124	0.126	0.154	-	0.214	**0.212**	0.280	-	**0.771**	0.785	0.894
$\sigma_{\epsilon,t}$ = SV Low-High	**0.198**	0.224	0.225	0.278	-	0.386	**0.373**	0.538	-	**1.362**	1.414	1.467

Notes: see Table 2.

Table 8: Results for Simulation 3 (Trend and Cosine) and $T = 150$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.149	0.096	**0.091**	0.115	-	0.159	**0.156**	0.202	-	0.405	0.415	**0.381**
σ_ϵ = Medium	0.181	0.153	**0.149**	0.182	-	0.244	**0.243**	0.342	-	**0.604**	0.624	0.628
σ_ϵ = High	**0.253**	0.275	0.276	0.313	-	**0.419**	0.421	0.680	-	**1.218**	1.248	1.235
$\sigma_{\epsilon,t}$ = SV Low-Med	0.167	0.121	**0.114**	0.140	-	**0.192**	0.192	0.246	-	0.485	0.501	**0.480**
$\sigma_{\epsilon,t}$ = SV Low-High	0.193	**0.177**	0.178	0.210	-	**0.261**	0.262	0.420	-	0.749	0.767	**0.609**
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.104	**0.077**	0.085	0.118	-	**0.138**	0.139	0.186	-	0.365	0.377	**0.342**
σ_ϵ = Medium	0.128	**0.127**	0.131	0.161	-	0.223	**0.221**	0.337	-	**0.585**	0.599	0.602
σ_ϵ = High	**0.224**	0.249	0.250	0.276	-	0.397	**0.397**	0.654	-	**1.211**	1.243	1.262
$\sigma_{\epsilon,t}$ = SV Low-Med	0.112	**0.098**	0.100	0.147	-	**0.173**	0.175	0.223	-	0.455	0.474	**0.445**
$\sigma_{\epsilon,t}$ = SV Low-High	**0.145**	0.151	0.154	0.194	-	**0.241**	0.241	0.374	-	0.726	0.743	**0.657**
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	**0.062**	0.062	0.065	0.071	-	**0.107**	0.108	0.130	-	**0.299**	0.311	0.317
σ_ϵ = Medium	**0.096**	0.103	0.107	0.116	-	0.174	**0.173**	0.279	-	**0.547**	0.559	0.604
σ_ϵ = High	**0.204**	0.221	0.221	0.259	-	0.384	**0.378**	0.552	-	**1.204**	1.234	1.280
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.073**	0.077	0.081	0.093	-	**0.126**	0.128	0.164	-	0.406	0.423	**0.404**
$\sigma_{\epsilon,t}$ = SV Low-High	**0.111**	0.121	0.126	0.162	-	**0.202**	0.202	0.323	-	0.713	0.734	**0.693**

Notes: see Table 2.

Table 9: Results for Simulation 4 (Mixture) and $T = 150$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.065	0.063	**0.061**	0.069	-	0.092	**0.091**	0.094	-	**0.277**	0.280	0.337
σ_ϵ = Medium	0.097	**0.096**	0.097	0.113	-	0.175	**0.172**	0.201	-	**0.542**	0.546	0.568
σ_ϵ = High	**0.203**	0.212	0.206	0.227	-	0.380	**0.375**	0.585	-	**1.187**	1.214	1.230
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.080**	0.082	0.081	0.089	-	0.125	**0.122**	0.144	-	**0.386**	0.392	0.404
$\sigma_{\epsilon,t}$ = SV Low-High	**0.115**	0.118	0.120	0.145	-	0.200	**0.197**	0.311	-	**0.696**	0.720	0.699
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.086	0.073	**0.071**	0.092	-	0.106	**0.104**	0.119	-	**0.290**	0.299	0.349
σ_ϵ = Medium	0.111	**0.107**	0.108	0.126	-	0.184	**0.183**	0.222	-	**0.547**	0.559	0.594
σ_ϵ = High	**0.208**	0.215	0.215	0.242	-	0.381	**0.378**	0.547	-	**1.183**	1.212	1.286
$\sigma_{\epsilon,t}$ = SV Low-Med	0.096	0.093	**0.092**	0.105	-	0.132	**0.131**	0.154	-	**0.397**	0.410	0.439
$\sigma_{\epsilon,t}$ = SV Low-High	**0.127**	0.129	0.130	0.159	-	0.210	**0.206**	0.326	-	0.707	0.733	**0.678**
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	0.106	**0.080**	0.082	0.101	-	**0.124**	0.125	0.138	-	**0.315**	0.328	0.334
σ_ϵ = Medium	0.129	0.124	**0.123**	0.142	-	0.198	**0.197**	0.227	-	**0.561**	0.579	0.640
σ_ϵ = High	**0.218**	0.224	0.225	0.258	-	0.410	**0.409**	0.583	-	**1.208**	1.225	1.343
$\sigma_{\epsilon,t}$ = SV Low-Med	0.120	**0.100**	0.105	0.129	-	**0.147**	0.149	0.170	-	**0.417**	0.434	0.473
$\sigma_{\epsilon,t}$ = SV Low-High	0.140	**0.134**	0.139	0.169	-	0.221	**0.218**	0.348	-	**0.709**	0.734	0.712

Notes: see Table 2.

Table 10: Results for Simulation 1 (Cosine) and $T = 600$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.102	0.105	**0.078**	0.102	-	0.112	0.080	**0.077**	-	0.129	0.126	**0.095**
σ_ϵ = Medium	**0.133**	0.151	0.147	0.164	-	0.139	0.135	**0.134**	-	0.149	**0.146**	0.167
σ_ϵ = High	**0.187**	0.205	0.206	0.251	-	0.204	**0.199**	0.206	-	0.218	**0.212**	0.241
$\sigma_{\epsilon,t}$ = SV Low-Med	0.118	0.124	**0.113**	0.125	-	0.127	0.115	**0.098**	-	0.134	0.131	**0.120**
$\sigma_{\epsilon,t}$ = SV Low-High	**0.130**	0.149	0.149	0.174	-	0.136	**0.134**	0.143	-	0.148	**0.146**	0.182
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	**0.087**	0.132	0.115	0.120	-	0.145	0.124	**0.105**	-	0.219	0.219	**0.109**
σ_ϵ = Medium	**0.204**	0.208	0.206	0.204	-	0.212	0.209	**0.161**	-	0.236	0.233	**0.195**
σ_ϵ = High	**0.242**	0.263	0.262	0.308	-	0.257	**0.253**	0.266	-	0.279	**0.275**	0.297
$\sigma_{\epsilon,t}$ = SV Low-Med	0.154	0.152	0.142	**0.140**	-	0.170	0.156	**0.135**	-	0.222	0.221	**0.117**
$\sigma_{\epsilon,t}$ = SV Low-High	**0.205**	0.206	0.207	0.225	-	0.211	0.210	**0.188**	-	0.234	0.231	**0.191**
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	**0.087**	0.146	0.148	0.158	-	0.180	0.183	**0.108**	-	0.339	0.340	**0.128**
σ_ϵ = Medium	0.238	**0.236**	0.239	0.266	-	0.275	0.278	**0.216**	-	0.346	0.346	**0.178**
σ_ϵ = High	**0.312**	0.338	0.336	0.350	-	0.344	**0.343**	0.346	-	0.374	**0.372**	0.407
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.115**	0.174	0.176	0.174	-	0.210	0.214	**0.162**	-	0.340	0.342	**0.141**
$\sigma_{\epsilon,t}$ = SV Low-High	**0.230**	0.233	0.241	0.258	-	0.274	0.280	**0.213**	-	0.348	0.347	**0.199**

Notes: see Table 2.

Table 11: Results for Simulation 2 (Break) and $T = 600$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.082	0.084	**0.068**	0.086	-	0.109	**0.094**	0.120	-	**0.219**	0.222	0.236
σ_ϵ = Medium	0.140	0.136	**0.128**	0.177	-	0.185	**0.182**	0.210	-	0.372	**0.365**	0.418
σ_ϵ = High	**0.213**	0.237	0.241	0.322	-	0.348	**0.347**	0.358	-	0.763	**0.743**	0.948
$\sigma_{\epsilon,t}$ = SV Low-Med	0.113	0.103	**0.090**	0.109	-	0.133	**0.123**	0.155	-	**0.265**	0.266	0.291
$\sigma_{\epsilon,t}$ = SV Low-High	0.138	0.132	**0.130**	0.229	-	**0.175**	0.179	0.233	-	**0.376**	0.385	0.514
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.085	0.109	0.096	**0.081**	-	0.140	0.127	**0.124**	-	**0.275**	0.298	0.316
σ_ϵ = Medium	0.182	0.163	**0.152**	0.169	-	**0.221**	0.230	0.251	-	**0.417**	0.431	0.515
σ_ϵ = High	**0.270**	0.274	0.280	0.414	-	**0.384**	0.397	0.411	-	0.788	**0.778**	0.908
$\sigma_{\epsilon,t}$ = SV Low-Med	0.121	0.130	0.121	**0.108**	-	0.165	**0.161**	0.162	-	**0.315**	0.341	0.355
$\sigma_{\epsilon,t}$ = SV Low-High	0.185	**0.162**	0.163	0.209	-	**0.211**	0.230	0.274	-	**0.421**	0.446	0.539
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	**0.089**	0.133	0.134	0.092	-	0.177	0.341	**0.121**	-	0.355	0.466	**0.316**
σ_ϵ = Medium	**0.151**	0.205	0.248	0.163	-	0.267	0.404	**0.218**	-	**0.505**	0.524	0.587
σ_ϵ = High	**0.285**	0.338	0.395	0.316	-	0.455	0.479	**0.449**	-	0.898	**0.817**	1.252
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.113**	0.154	0.164	0.120	-	0.203	0.375	**0.146**	-	0.402	0.483	**0.396**
$\sigma_{\epsilon,t}$ = SV Low-High	**0.147**	0.198	0.255	0.185	-	0.255	0.409	**0.246**	-	**0.498**	0.534	0.789

Notes: see Table 2.

Table 12: Results for Simulation 3 (Trend and Cosine) and $T = 600$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	**0.064**	0.085	0.089	0.093	-	0.131	0.123	**0.103**	-	**0.217**	0.218	0.239
σ_ϵ = Medium	**0.120**	0.143	0.133	0.157	-	0.188	0.188	**0.169**	-	0.264	**0.263**	0.380
σ_ϵ = High	**0.200**	0.216	0.217	0.262	-	**0.238**	0.239	0.272	-	0.433	**0.419**	0.799
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.080**	0.104	0.097	0.107	-	0.156	0.148	**0.127**	-	**0.229**	0.231	0.279
$\sigma_{\epsilon,t}$ = SV Low-High	**0.114**	0.139	0.134	0.163	-	0.183	0.187	**0.181**	-	0.273	**0.272**	0.480
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	**0.059**	0.071	0.099	0.102	-	0.113	0.123	**0.111**	-	**0.189**	0.198	0.219
σ_ϵ = Medium	0.124	**0.117**	0.119	0.150	-	0.156	0.163	**0.148**	-	**0.244**	0.247	0.321
σ_ϵ = High	**0.156**	0.180	0.180	0.201	-	**0.221**	0.225	0.237	-	0.418	**0.408**	0.698
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.084**	0.086	0.108	0.116	-	0.133	0.135	**0.119**	-	**0.202**	0.210	0.282
$\sigma_{\epsilon,t}$ = SV Low-High	0.118	**0.112**	0.121	0.145	-	**0.152**	0.163	0.163	-	**0.249**	0.258	0.484
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	**0.034**	0.039	0.041	0.040	-	0.060	0.072	**0.046**	-	**0.131**	0.167	0.212
σ_ϵ = Medium	0.073	**0.067**	0.073	0.072	-	0.101	0.125	**0.080**	-	**0.206**	0.220	0.252
σ_ϵ = High	**0.116**	0.120	0.128	0.130	-	0.187	0.203	**0.179**	-	0.400	**0.395**	0.493
$\sigma_{\epsilon,t}$ = SV Low-Med	**0.044**	0.048	0.049	0.050	-	0.072	0.088	**0.057**	-	**0.152**	0.181	0.210
$\sigma_{\epsilon,t}$ = SV Low-High	0.065	**0.064**	0.072	0.068	-	**0.096**	0.121	0.099	-	**0.217**	0.236	0.369

Notes: see Table 2.

Table 13: Results for Simulation 4 (Mixture) and $T = 600$

	$K = 6$				$K = 20$				$K = 100$			
	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR	BVAR	2SRR	GLRR	GRRRR
$\mathbf{K^*/K = 0.2}$												
σ_ϵ = Low	0.049	0.050	**0.038**	0.053	-	0.057	**0.052**	0.062	-	0.109	**0.105**	0.108
σ_ϵ = Medium	0.071	0.076	**0.070**	0.096	-	0.092	**0.091**	0.095	-	0.182	**0.178**	0.206
σ_ϵ = High	**0.107**	0.117	0.114	0.144	-	0.171	**0.169**	0.170	-	0.383	**0.371**	0.642
$\sigma_{\epsilon,t}$ = SV Low-Med	0.057	0.059	**0.051**	0.069	-	0.069	**0.066**	0.075	-	0.131	**0.127**	0.144
$\sigma_{\epsilon,t}$ = SV Low-High	**0.066**	0.069	0.068	0.087	-	**0.088**	0.090	0.107	-	0.192	**0.192**	0.292
$\mathbf{K^*/K = 0.5}$												
σ_ϵ = Low	0.062	0.061	**0.052**	0.066	-	0.078	**0.073**	0.082	-	**0.134**	0.135	0.141
σ_ϵ = Medium	0.091	0.089	**0.085**	0.109	-	**0.113**	0.114	0.117	-	0.205	**0.199**	0.221
σ_ϵ = High	**0.127**	0.134	0.136	0.179	-	0.189	**0.187**	0.190	-	0.393	**0.382**	0.532
$\sigma_{\epsilon,t}$ = SV Low-Med	0.074	0.068	**0.062**	0.075	-	0.089	**0.087**	0.097	-	0.156	**0.154**	0.163
$\sigma_{\epsilon,t}$ = SV Low-High	0.089	**0.084**	0.085	0.111	-	**0.110**	0.114	0.132	-	**0.206**	0.209	0.333
$\mathbf{K^*/K = 1}$												
σ_ϵ = Low	0.076	**0.070**	0.073	0.090	-	**0.101**	0.114	0.113	-	**0.179**	0.188	0.213
σ_ϵ = Medium	0.121	**0.108**	0.109	0.131	-	**0.141**	0.150	0.145	-	0.244	**0.238**	0.268
σ_ϵ = High	**0.158**	0.162	0.163	0.199	-	0.224	0.219	**0.218**	-	0.431	**0.411**	0.686
$\sigma_{\epsilon,t}$ = SV Low-Med	0.096	**0.083**	0.084	0.107	-	**0.114**	0.128	0.124	-	**0.200**	0.201	0.244
$\sigma_{\epsilon,t}$ = SV Low-High	0.120	**0.105**	0.110	0.146	-	**0.138**	0.148	0.160	-	0.248	**0.248**	0.381

Notes: see Table 2.

Table 14: Forecasting Results

	AR				ARDI				VAR5				VAR20			
	Plain	2SRR	GLRR	GRRRR	Plain	2SRR	GLRR	GRRRR	Plain	2SRR	GLRR	GRRRR	Plain	2SRR	GLRR	GRRRR
GDP																
$h=1$	1.00	0.98	0.99	**0.98**	1.03	1.11	1.10	1.04	1.04	1.06	1.05	1.01	1.24	2.07	1.61**	1.45*
$h=2$	**1.00**	1.10	1.00	1.00	1.05	1.77	1.06	1.08	1.08	1.39	1.16*	1.08**	1.27	1.32**	1.34**	1.70*
$h=4$	**1.00**	1.23	1.06	1.06	1.07	1.41	1.06	1.08	1.06	1.41	1.14	1.15	1.10	1.27	1.23**	1.12
UR																
$h=1$	1.00	1.11	1.15	1.15	**0.99**	1.02	1.05	1.17	1.10*	1.13	1.20	1.10	1.63	1.45	1.76*	1.37
$h=2$	1.00	1.46	1.29	1.19	**1.00**	1.80	1.48	1.03	1.11	1.44	1.44	1.39	1.40	2.11	1.76	2.21
$h=4$	**1.00**	1.59	1.34	1.13**	1.00	1.47	1.32	1.19	1.09	1.40	1.30	1.40	1.07	1.49	1.33*	1.33*
INF																
$h=1$	1.00	**0.93**	0.96	0.95	1.01	0.99	1.09	0.95	1.00	0.94	0.95	1.22	1.79	1.79	1.81	1.85
$h=2$	**1.00**	1.14	1.15	1.00	1.06	1.35	1.14	1.49	1.03	1.39	1.17	1.26	1.15	1.77	1.53	1.09*
$h=4$	**1.00**	1.09	1.10	1.12	1.06*	1.20	1.11	1.15	1.00	1.11	1.09	1.02	1.38	1.64	1.12	1.10
IR																
$h=1$	1.00	0.64***	0.71***	0.79***	0.94***	**0.64*****	0.80***	1.05	1.07	0.72***	0.82**	1.09**	1.46*	2.03	2.28	0.71***
$h=2$	1.00	0.72***	**0.66*****	0.72***	0.94**	0.86	0.74**	0.99	0.97	0.95	0.98	0.93	1.37	2.46	0.79**	1.34
$h=4$	1.00	1.03	1.06	1.12	0.93	1.04	0.91	1.13	0.97	1.12	0.96	0.93	1.08	1.24	0.93	**0.81***
SPREAD																
$h=1$	1.00	0.86***	0.86***	0.86**	0.90**	0.86**	0.85**	0.84***	0.96	**0.82*****	0.87**	0.93	2.13	2.70*	2.48*	1.94
$h=2$	1.00	1.02	1.00	0.96	0.91*	1.09	1.06	0.95	**0.88****	0.96	0.91	0.92	2.03	2.31	2.30	2.01
$h=4$	1.00	1.58	1.14	1.32	0.90**	1.35	1.27	0.95	**0.88****	1.34	1.20	1.01	0.89	1.50	1.25	1.16

Notes: This table reports $RMSPE_{v,h,m}/RMSPE_{v,h,\text{Plain AR(2)}}$ for 5 variables, 3 horizons and 16 models considered in the pseudo-out-of-sample experiment. Numbers in bold identifies the best predictive performance of the row. Diebold-Mariano tests are performed to evaluate whether the difference in predictive performance between a model and the AR(2) benchmark is statistically significant. '*', '**' and '***' respectively refer to p-values below 10%, 5% and 1%.

Table 15: Forecasting Results, *Half & Half*

	AR				ARDI				VAR5				VAR20			
	Plain	2SRR	GLRR	GRRRR	Plain	2SRR	GLRR	GRRRR	Plain	2SRR	GLRR	GRRRR	Plain	2SRR	GLRR	GRRRR
GDP																
$h=1$	1.00	**0.99**	0.99	0.99	1.03	1.01	1.00	1.04	1.04	1.02	1.03	1.02	1.24	1.59	1.34	1.24
$h=2$	1.00	1.00	0.97	1.00	1.05	1.21	**0.97**	1.05	1.08	1.10	1.06	1.07*	1.27	1.19	1.20	1.33
$h=4$	1.00	1.01	0.96	0.95	1.07	1.06	**0.94**	0.97	1.06	1.09	1.01	0.97	1.10	1.05	1.05	0.97
UR																
$h=1$	1.00	1.02	1.04	1.02	0.99	**0.97**	0.98	1.04	1.10*	1.09	1.12	1.06	1.63	1.47	1.59	1.27*
$h=2$	1.00	1.08	1.08	1.03	**1.00**	1.16	1.12	1.01	1.11	1.15	1.17	1.17	1.40	1.67	1.50	1.70
$h=4$	**1.00**	1.15	1.10	1.05*	1.00	1.08	1.04	1.02	1.09	1.16	1.14*	1.19*	1.07	1.20	1.15	1.11
INF																
$h=1$	1.00	**0.94**	0.95	0.96	1.01	0.98	1.01	0.97	1.00	0.96	0.96	1.06	1.79	1.78	1.78	1.74
$h=2$	1.00	0.91	0.92	0.94	1.06	1.12	**0.91***	1.15	1.03	1.07	0.93	1.07	1.15	1.30**	1.18	1.07
$h=4$	1.00	0.85*	0.85*	0.86*	1.06*	0.88	0.84	0.91	1.00	0.93	**0.84**	0.87	1.38	1.45	1.17	1.13
IR																
$h=1$	1.00	0.80***	0.84***	0.88***	0.94***	**0.76*****	0.85***	0.96	1.07	0.87**	0.93	1.04	1.46*	1.67	1.77	0.98
$h=2$	1.00	0.83***	0.76***	0.82***	0.94**	0.85**	**0.76*****	0.93	0.97	0.88*	0.95	0.92**	1.37	1.84	0.98	1.26
$h=4$	1.00	0.99	1.01	1.02	0.93	0.95	0.90	0.99	0.97	0.99	0.92	0.91	1.08	1.07	0.97	**0.88**
SPREAD																
$h=1$	1.00	0.90***	0.92***	0.89***	0.90**	**0.86****	0.87***	0.86***	0.96	0.87***	0.90**	0.92*	2.13	2.30	2.21	1.98
$h=2$	1.00	0.97	0.98	0.97	0.91*	0.94	0.95	0.92	0.88**	**0.88**	0.88	0.88**	2.03	2.12	2.12	1.90
$h=4$	1.00	1.21	1.03	1.12	0.90**	1.03	1.03	0.91*	**0.88****	1.03	0.98	0.92	0.89	1.08	0.95	0.98

Notes: This table reports $RMSPE_{v,h,m}/RMSPE_{v,h,\text{Plain AR(2)}}$ for 5 variables, 3 horizons and 16 models considered in the pseudo-out-of-sample experiment. TVPs of each model are shrunk to constant parameters via model averaging with equal weights for both the TVP model and its constant coefficients counterpart. Numbers in bold identifies the best predictive performance of the row. Diebold-Mariano tests are performed to evaluate whether the difference in predictive performance between a model and the AR(2) benchmark is statistically significant. '*', '**' and '***' respectively refer to p-values below 10%, 5% and 1%.

1.7.5. *Additional Graphs*

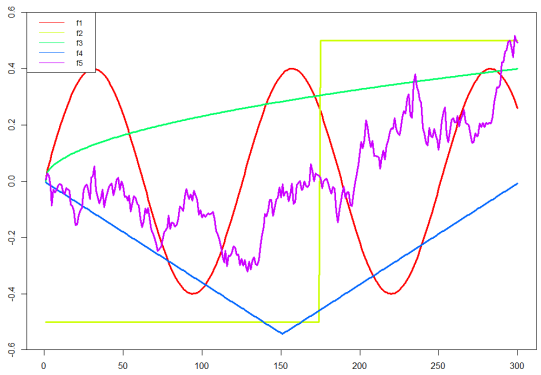
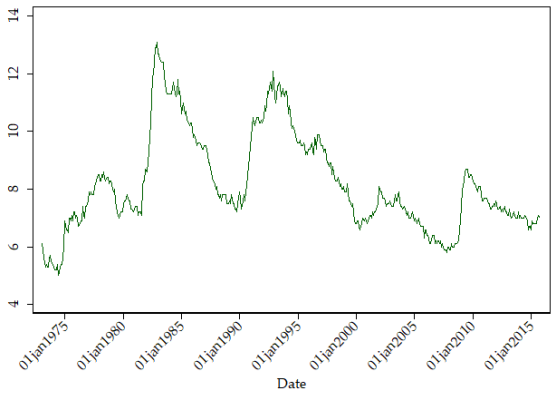
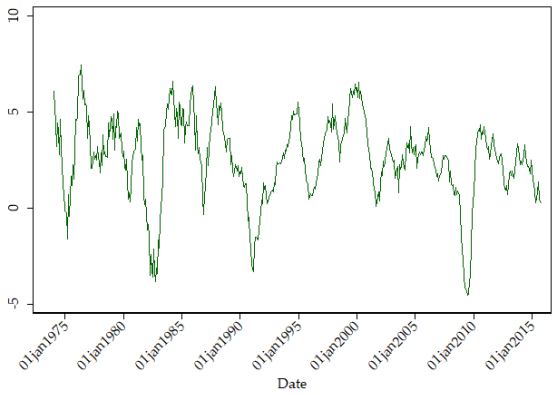


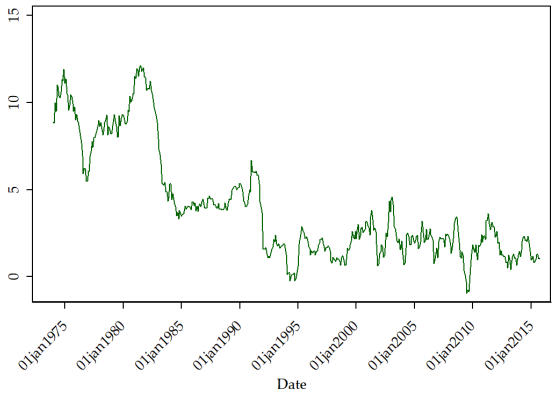
Figure 4: This graph displays the 5 paths out of which the true $\beta_{k,t}$'s will be constructed for simulations.



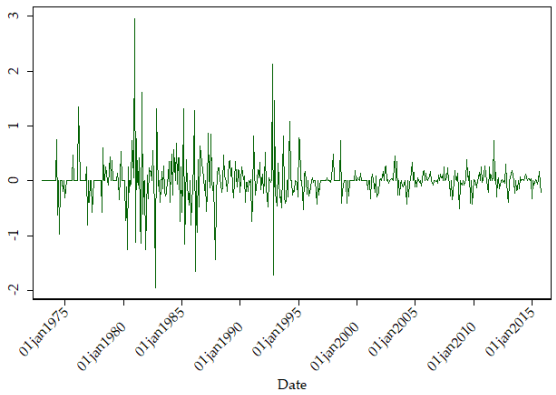
(a) Unemployment Rate



(b) Month over Month GDP growth



(c) Month over Month Inflation Rate



(d) Monetary Policy Shocks

Figure 5: Four Main Canadian Time series

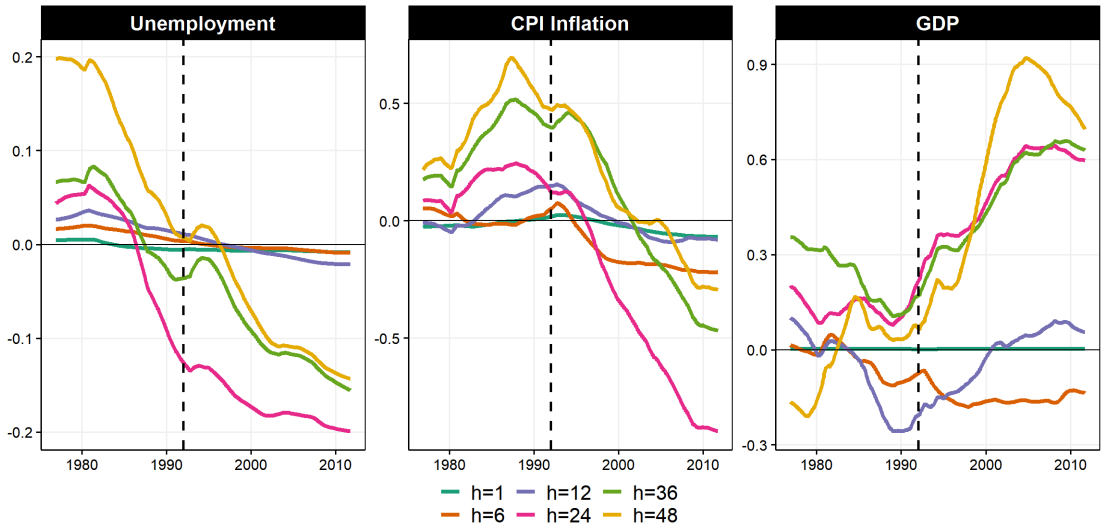


Figure 6: $\beta_t^{2SRR} - \beta^{OLS}$ for the cumulative effect of MP shocks on variables of interest. Note that for better visibility, GDP and CPI Inflation units are now percentages. Dashed black line is the onset of inflation targeting.

CHAPTER 2 : THE MACROECONOMY AS A RANDOM FOREST

2.1. Introduction

The rise of Machine Learning (ML) led to great excitement in the econometrics community. In applied macroeconomics, a first wave of papers took ML algorithms off the shelf and went hunting for forecasting gains. With the emerging consensus that some ML offerings can appreciably increase predictive accuracy, a question emerges: what is the place of economics in all that?

The conditional mean is the most basic input to any empirical macroeconomic analysis. Anything else that follows (e.g., structural analysis) depends on it. Thus, getting it right is not merely useful, it is *necessary*. Clearly, in that regard, ML can help. However, while the latter gladly delivers prediction accuracy gains (and ergo a conditional mean closer to the truth), it is much more reluctant to disclose its inherent model. Consequently, ML is currently of great use to macroeconomic forecasting, but of little help to macroeconomics. I propose a simple remedy: shifting the focus of the algorithmic arsenal away from predicting y_t into modeling β_t, which are economically meaningful coefficients in a time-varying macroeconomic equation. The newly proposed algorithm, *Macroeconomic Random Forest* (MRF) kills two coveted birds with one stone. First, in most instances, MRF forecasts better than off-the-shelf ML algorithms and traditional econometric approaches. Second, its main output, *Generalized Time-Varying Parameters* (GTVPs), can be interpreted. Their versatility comes from nesting many popular specifications (structural breaks/change, threshold effects, regime-switching, etc.) and letting the data decide whichever combination of them is most suitable. Ultimately, we get a new methodology leveraging the power of ML and big data to provide a modern take on the decades-old challenge of estimating latent states driving linear macroeconomic equations.

THE STATE OF EMPIRICAL MACRO AFFAIRS. Answering positively two questions guarantees a viable conditional mean: "are all the relevant variables included in the model?" and at a higher level of sophistication, "is linearity a valid approximation of reality?". The first one led to the successful development of factor models and large Bayesian Vector Autoregressions (VARs) over the last two decades. To address the second, applied macroeconomic researchers have proposed many non-linear time series models based on reasonable

economic intuition. Most of them amount to have regression coefficients β_t in

$$y_t = X_t\beta_t + \varepsilon_t$$

evolving through time. The β_t process can take many forms, and a choice must be made *a priori* out of many equally plausible alternatives. Notable members of the vast time-variations catalog are threshold/switching regressions (Hansen, 2011), smooth transition (Teräsvirta, 1994), structural breaks (Perron et al., 2006; Stock, 1994), and random walk time-varying parameters (Sims, 1993; Cogley and Sargent, 2001; Primiceri, 2005). While it is uncontroversial that factor models and large Bayesian VARs have gone a long way in meeting their original goals, less victorious statements are available for the various time-variation proposals. Why?

More often than not, nonlinear time series models use little data and/or restrict stringently the shape of β_t's path. While the consequences for forecasting are direct and obvious, those for analysis of macroeconomic relationships are equally problematic. Is the evolving Taylor rule characterized by switching regimes (Sims and Zha, 2006), a Volker structural break (Clarida et al., 2000), or gradually evolving parameters (Boivin, 2005; Primiceri, 2005)? This discordance interferes with our understanding of the past while impacting our expectations for tomorrow's β_t. I now divide popular time-variation approaches into two strands, discuss their shortcomings, and complete by explaining how MRF addresses them.

Observable Time-Variation Via Interaction Terms. Using interaction terms and related refinements is a parsimonious way to create time variation in a linear equation. For instance, switching regimes based on an observed regressor can be obtained by interacting the linear equation with the indicator function $I(q_t > c)$, where c is some value, and q_t is a threshold variable chosen by the researcher. However, using the FRED-QD US macro data set (McCracken and Ng, 2016) reveals an overwhelmingly large number of candidates for q_t. Additionally, there may be multiple regimes interacting together. Or the "true" q_t could be an unknown function of available regressors. And structural breaks or slow exogenous variation could get in the way. The list goes on. This renders a credible exploration of the threshold structures' space impossible and the enterprise of manually specifying the model very much compromised.

Here is an empirical example. Auerbach and Gorodnichenko (2012b) and Ramey and Zubairy (2018b) use a GDP/unemployment indicator to let the effects of fiscal stimulus (potentially) vary with the state of the economy. Batini et al. (2012) allow for additional dependence on the origin of the impulse (revenue or spending). Such honorable explorations could go on endlessly. MRF provides a hammer solution to the problem. First,

the near-universe of threshold structures can be characterized by regression trees — see section 2.2.1. Second, MRF embeds, among other things, a powerful greedy algorithm designed to explore such "structure" spaces.

LATENT TIME-VARIATION. Some methods with an aura of greater flexibility are labeled as "latent change". In this line of work, β_t either follows a law of motion (random walk, Markov process) or could be subject to discrete breaks.[1] At first glance, this appears to solve many of the problems of interaction terms approaches. By treating β_t as a state to be filtered/estimated within the model, the complexity of characterizing its path correctly out of abundant data seems to vanish. Alas, estimating β_t's path implies a great number of parameters (in fact, often greater than the number of observations, Goulet Coulombe 2020a) which inevitably necessitates strong regularization. That regularization is the law of motion itself, a choice far from innocuous – and akin to that of q_t in "observable" change models. Accordingly, whether it is latent regime-switching, exogenous breaks, or slow change, none can easily accommodate for the additional presence of the other. Yet, these models are routinely fitted *separately* on the *same data*. Consequently, methods often detect what they are designed to detect, in near-complete abstraction of imaginable interference from other nonlinearities.

Additionally, while "latent" approaches may sometimes rationalize the data well in-sample, many of them will struggle to outperform a simple benchmark *out-of-sample*. Often, the very nature of β_t's law of motion creates forecasting headaches. Classical TVPs imply a two-sided vs one-sided filtering problem. Analogously, detecting a structural break is much harder without a great amount of data on both sides of it. Moreover, there is the obvious problem of statistical efficiency. If the Phillips curve flattened because an economy became increasingly open, including an interaction term with imports/exports is wildly more efficient than obtaining the whole β_t path non-parametrically. Thus, exogenous structural change should be, in some sense, a time variation of last resort. The advantage of MRF is that it algorithmically search for "observable" low-hanging fruits, and turn to split the sample with t only if necessary. Further, it implicitly creates a forecasting function for β_t which is an RF in its own right. This is, almost in any case, much more powerful than existing alternatives – like random walks.

MECHANICS. The key difference when adding the M to MRF is the inclusion of a linear part within each of the tree leaves, rather than just an intercept. Motivated in cross-sectional applications to improve the efficiency of nonparametric estimation (in the spirit

[1] Simpler derivatives are often used in applied work. In forecasting, rolling-window estimation drops early observations. In empirical macro, pre-defined subsamples are popular (Clarida et al., 2000; Del Negro et al., 2020).

of local linear regression), trees with linear parts have been considered (among others) in Alexander and Grimshaw (1996) and Wang and Witten (1996). Friedberg et al. (2018) expand on this by considering an ensemble of them (i.e., a forest) and focusing on the problem of treatment effect heterogeneity. Of course, the difference here is that a linear part is much more meaningful when one can look at β_t as a process of its own – and as a synthesis of nonlinear time series models. Finally, it is noteworthy that the approach may come in semiparametric partially linear clothing, yet it makes no compromise on the range of nonlinearities it captures. This is a virtue of time-varying coefficients models being able to approximate any nonlinear function (Granger, 2008).

The chapter also introduces new devices enhancing MRF's predictive and interpretability potential. First, I propose Moving Average Factors (MAFs) as a simple way to compress ex-ante the information contained in the lags of a regressor entering the RF part of MRF. They boost the meaningfulness of tree splits and helps avoid running out of them quickly. The transformation is motivated by the literature on constraining/regularizing lag polynomials (Shiller, 1973). Precisely, MAFs' contribution is to induce similar shrinkage when there are no explicit coefficients to shrink. When it comes to GTVPs themselves, I provide a regularization scheme better suited for time series which procures a desirably smoother path with respect to time. It is inspired by the random walk shrinkage of the classical TVP literature and is implemented within the tree procedure by weighted least-squares. Finally, a variant of the Bayesian Bootstrap provides credible regions that are instrumental for the interpretation of GTVPs.

Results. In simulations, the tool does comparably well to traditional nonlinear time series models when the data generating process (DGP) matches what the latter is designed for. When the time-variation structure becomes out of reach for classical approaches, MRF wins. Additionally, it supplants plain RF whenever persistence is pervasive. In a forecasting application, the MRFs gains are present for almost all variables and horizons under study, a rarity for nonlinear forecasting approaches. For instance, the Autoregressive Random Forest (ARRF) almost always supplant its resilient OLS counterpart. Also, an MRF where the linear part is a compact factor-augmented autoregression generates very accurate forecasts of the 2008 downturn for both GDP and the unemployment rate (UR). Inspection of resulting GTVPs reveals they behave differently from random walk TVPs. For instance, in the UR equation, the contribution of forward-looking variables nearly doubles before every recession — including 2008 where the associated β_t is forecasted to do so out-of-sample. This reinforces the view that financial indicators and other market-based expectations proxies can rapidly capture downside risks around business cycle turning points (Adrian et al., 2019). MRF learned and applied it to great success.

Inflation is subject to a variety of time-variations, detection of which would be compromised by approaches lacking the generality of MRF. The long-run mean and the persistence evolved slowly and in an exogenous fashion — this has been repeatedly found in the literature (e.g., Cogley and Sargent 2001). More novel is the finding that the real activity factor's effect on the price level depends positively on the strength of well-known leading indicators, especially housing-related. Following this lead, I complete the analysis by looking at a traditional Phillips' curve specification. I report that the inflation/unemployment trade-off coefficient decreased significantly since the 1980s *and* also varies strongly along the business cycle. Among other things, it is extremely weak following every recession. This nuances current evidence on the flattening Phillips curve, which, by design, focused almost entirely on long-run exogenous change (Blanchard et al., 2015; Galí and Gambetti, 2019; Del Negro et al., 2020). Overall, MRF suggests inflation can rise from a positive unemployment gap, but it goes down much more timidly from economic slack. These findings are made possible by combining different tools within the new framework, such as credible intervals for the GTVPs, new variable importance measures specifically designed for MRF, and surrogate trees as interpretative devices for β_t.

Outline. Section 2.2 introduces MRF, motivates its use, considers practical aspects, and discusses relationships with available alternatives. Sections 2.3 and 2.4 report simulations and forecasting results, respectively. Section 2.5 analyzes various GTVPs of interest. Section 3.5 concludes.

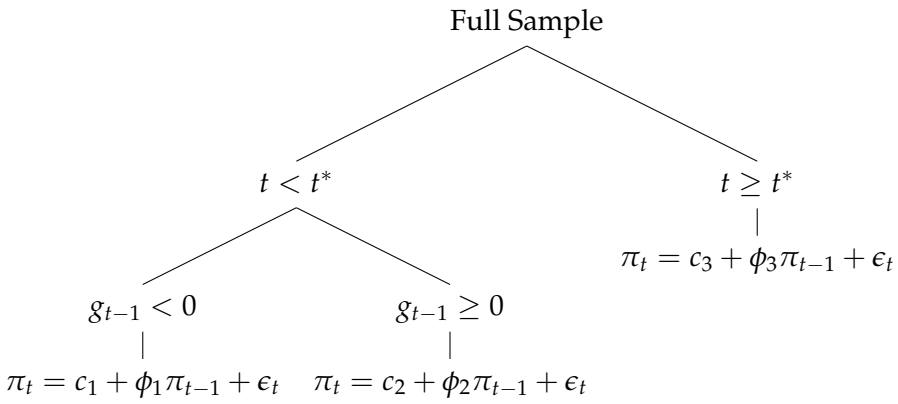
2.2. Macroeconomic Random Forests

This section introduces MRF. I first motivate the use of trees as basis functions by casting standard switching structures for autoregressions as special cases. Second, I detail the MRF mechanics and how it yields GTVPs. Third, I discuss how the approach relates to both standard RF and traditional random walk TVPs. Fourth, I discuss interpretability potential and provide a way to assess parameter uncertainty.

2.2.1. Traditional Macro Non-Linearities as Trees

Within the modern ML canon, Random Forest (RF) is an extremely popular algorithm because it allows for complex nonlinearities, handles high-dimensional data, bypasses overfitting, and requires little to no tuning. This is in sharp contrast with, for example, Neural Networks, whose ability to fail upon a bad choice of hyperparameters is largely unmatched. Thus, RF is a reasonable device to look into for constructing GTVPs. But there is more: many common time series nonlinearities fit within a tree structure. Hence, it will be all the more natural to think of MRF as a generalization of previous nonlinear offerings. Overall, it eliminates the arbitrary search for a specification. By creating a unified view, the myriad of time-variations suggested separately can now be tackled jointly.

I now present two examples displaying how common time series nonlinearities imply a tree structure for an AR process. Let us consider the inflation process in a country where inflation targeting (IT) was implemented at a publicly known date (like in Canada). Let π_t be inflation at time t and t^* is the onset date of IT. Additionally, g_t is some measure of output gap. A plausible model is reported in the tree graph below. The story is straightforward. Inflation behaved differently before vs after IT. After IT, it is a simple AR process. Before IT, it was a switching AR process which dynamics and mean depended on the sign of the output gap.[2]



This is one story out of many that trees can characterize. In practice, none of the above is known. The structure, the splitting variables, and the splitting points could be different. This is both good and bad news. It highlights the flexibility of trees. It also suggests that designing the "true" one from economic deduction is a daunting task — equally plausible alternatives are easily imaginable. Fortunately, algorithms can point out which trees in better agreement with the data.

A global grid search is computationally unfeasible if either S_t is large or if we want to consider more than a few splits (examples above included 2 and 3, respectively). A natural way forward is recursive partitioning of the data set via a *greedy* algorithm (Breiman et al., 1984).[3] A greedy algorithm optimizes functions by iteratively doing the best local update, rather than directly solving for a global optimum. As a result, it is prone to high variance (Friedman et al., 2001). Hence, considering a diversified portfolio of trees appears as the most sensible route. To achieve that, it is highly effective to use Bootstrap Aggregation (*Bagging*, Breiman 1996) of many de-correlated trees. This is the famous Random Forest proposition of Breiman (2001).

[2]Note that a standard regression tree would set all ϕ's to 0.

[3]A single autoregressive tree was proposed in Meek et al. (2002).

2.2.2. Generalized Time-Varying Parameters

The general model is

$$y_t = X_t\beta_t + \epsilon_t$$
$$\beta_t = \mathcal{F}(S_t)$$

where S_t are the state variables governing time variation and $\mathcal{F}$ a forest. S_t is oberved macroeconomic data which composition is motivated in section 2.2.6 and laid out explicitly in section 2.4. X determines the *linear* model that we want to be time-varying. Typically, $X_t \subset S_t$ is rather small (and focused) compared to S_t. For instance, an autoregressive random forests (ARRF) – which generalizes the cases of the previous section – uses lags of y_t for X_t. The tree fitting procedure underlying *plain* RF is not adequate, as it sets $X_t = \mathbf{1}$ by default. Thus, analogously to Friedberg et al. (2018), it is modified to

$$\min_{j\in\mathcal{J}^-,\ c\in\mathbb{R}} \left[\min_{\beta_1} \sum_{\{t\in l | S_{j,t}\leq c\}} (y_t - X_t\beta_1)^2 + \lambda\|\beta_1\|_2 + \min_{\beta_2} \sum_{\{t\in l | S_{j,t}> c\}} (y_t - X_t\beta_2)^2 + \lambda\|\beta_2\|_2 \right]. \tag{2.1}$$

The purpose of this problem is to find the optimal variable S_j (so, finding the best j out of the random subset of predictors indexes $\mathcal{J}^-$) to split the sample with, and at which value c of that variable should we split.[4] It outputs j^* and c^* which are used to split l (the parent node) into two children nodes, l_1 and l_2. We start with the leaf l being the full sample. Then, we perform a split according to the minimization problem, which procures us with 2 subsamples. Within each of these two newly created subsamples, we run (2.1) again. Repeating this process recursively constructs an ever-growing set of l's which are of ever-shrinking size. Doing so until a stopping criteria is met generates a tree.

Let the Trees Run Deep. Recursively splitting β_0 into β_1 and β_2 eventually leads to β_t. However, β_t, by construction, has very little company within its terminal node/leaf. As result, a single tree has low bias, but also very high variance for β_t. When fitting a single tree, the (early) stopping point must be tuned to avoid overfitting. However, this is not necessary when a sufficiently diversified ensemble of trees is considered. Originally, Breiman (2001) himself provided a bound on the generalization error that grows with the

[4]Note that, unlike Friedberg et al. (2018), S_t and X_t will differ, which is natural when motivated from a TVP perspective (but not so much from local linear regression one). Forcing their equivalence is not feasible nor desirable in a macro environment.

correlation between trees.[5] In Chapter 3, I go further by showing that RF's out-of-sample prediction is equivalent to the optimally "stopped" or "pruned" one, provided sufficiently diversified trees. The desirable property is attributed to the peculiar behavior of "randomized greedy algorithms", which are often overlooked as mere computational necessities. Those insights are of even greater use when it comes to time series since dependence and structural change pose challenges to hyperparameter tuning. Given a large enough B, a reasonable `mtry` (see "De-Correlation" below on this) and standard subsampling rate, we can be confident that the out-of-bag prediction and β_t's exclude fitted noise. In our specific context, it means the sample will not be over-split, and we are not going to see time variation when it is not there. Naturally, the credible regions proposed in section 2.2.7 will also help in that regard. The property will be illustrated in section 2.3.2.

(M)RF prediction is the *simple average* from those of its single trees. Same goes for β_t. RF is a clever diversification scheme which generates sufficient randomization for that average to inherit the above properties. To achieve that, it mixes elements of re-sampling and model averaging: Bagging and de-correlated trees.[6]

BAGGING. Each tree is "grown" on a bootstrapped sample (or a random subsample) (Breiman, 1996).[7] When the base learner is highly nonlinear in observation and/or unstable, gains from Bagging can be large (Breiman, 1996; Grandvalet, 2004). Nonparametric (or "pairs" MacKinnon 2006) bootstrap is being used — i.e., we are *not* shuffling residuals.[8] Rather, we are randomly selecting many observations triples $[y_t \;\; X_t \;\; S_t]$ (or pairs $[y_t \;\; S_t]$ for Plain RF), and then fit a tree on them. To deal with the dependence inherent to time series data and other reasons detailed in section 2.2.7, a slightly more sophisticated bootstrapping/subsampling procedure (involving blocks) will be used for MRF.

DE-CORRELATION. The second ingredient, proposed in Breiman (2001), is to consider "de-correlated" trees. RF is an average of many trees, and any averaging scheme reduces variance at a much faster rate if its components are uncorrelated. In our context, this is obtained by growing trees semi-stochastically. In equation (2.1), this is made operational by using $\mathcal{J}^- \subset \mathcal{J}$ rather than $\mathcal{J}$. In words, this means that at each step of the recursion, a different subsample of regressors is drawn to constitute candidates for the split. This pre-

[5] Also, Duroux and Scornet (2016) derive a formula (for a "median" forest) linking tuning parameters related to the depth of the trees and that of diversification.

[6] See Chapter 3 for a discussion on how RF compares and contrast with the forecast combinations/averaging literature.

[7] This does not preclude from obtaining β_t for all t's since β_t's attached to the excluded observations are simply generated by applying the tree on the "out-of-bag" data.

[8] Nonetheless, Bagging in itself is not estranged to macro forecasting (Inoue and Kilian, 2008; Hillebrand and Medeiros, 2010; Hillebrand et al., 2020). However, nearly all studies consider the more common problem of variable selection via hard-thresholding rules – like t-tests (Lee et al., 2020).

vents the greedy algorithm (which, as we know, only "thinks" locally) to always embark on the same optimization route. As a result, trees are further diversified and computing time, reduced. The fraction of randomly selected predictors is a tuning parameter typically referred to as `mtry` in the literature (and all software), with a default value of $\frac{1}{3}$ for regression settings. This, other algorithmic parameter settings, and some practical aspects are discussed in appendix 2.7.4.

Plain RF has many qualities readily transferable to MRF. It is easy to implement and to tune. That is, it has few tuning parameters that are usually of little importance to the overall performance – robustness. It is relatively immune to the adverse effects of including many irrelevant features (Friedman et al., 2001). Given the standard ratio of regressors to observations in macro data, this is a non-negligible advantage. Furthermore, with a sufficiently high `mtry`, it can adapt nicely to sparsity and discard useless predictors (Olson and Wyner, 2018). Finally, its vanilla version already shows good forecasting performance for US inflation (Medeiros et al., 2019) and macro data in general (Chen et al., 2019; Goulet Coulombe et al., 2019).

2.2.3. Random Walk Regularization

Equation (2.1) uses Ridge shrinkage which implies that each time-varying coefficient is implicitly shrunk to 0 at every point in time. λ and the prior it entails can exert a significant influence. For instance, if a process is highly persistent (AR coefficient lower than 1 but nevertheless quite high) as it is the case for SPREAD (see section 2.4), shrinking the first lag heavily to 0 could incur serious bias. Fortunately, this can easily be refined to a Minnesota-style prior if X_t corresponds to a Bayesian VAR equation. If X_t is low-dimensional (as it will often be), a simpler alternative consists in using OLS coefficients as prior means. Nonetheless, the specification of previous sections implies that if λ grows large, $\forall t \;\; \beta_t = 0$ (or whatever the prior mean is). $\beta_i = 0$ is a natural stochastic constraint in a cross-sectional setting, but its time series translation $\beta_t = 0$ can easily be suboptimal. The traditional regularization employed in macro is rather the random walk

$$\beta_t = \beta_{t-1} + u_t.$$

Thus, it is desirable to transform (2.1) so that it implements the prior that coefficients evolve smoothly (at least, to minimal extent), which is just shrinking β_t to be in the neighborhood of β_{t-1} and β_{t+1} rather than 0. This is in line with the view that economic states (as expressed by β_t here) last for at least a few consecutive periods. Note that unlike traditional TVP methods which rely extensively on smoothness regularization – as it is the sole regularizer, MRF makes only an very mild use of it to get rid of high-frequency noise that may be left in β_t. The main benefit is to facilitate the interpretation of resulting GTVPs.

I implement the desired regularization by taking the "rolling-window view" of time-varying parameters, which has been exploited recently to estimate large TVP-VARs (Giraitis et al., 2018; Petrova, 2019). That is, the tree, instead of solving a plethora of small ridge problems, will rather solve many weighted least squares problems (WLS) which includes close-by observations. The latter are in the neighborhood (in time) of observations within current leaf. They are included in estimation, but are allocated a smaller weight.

For simplicity and to keep computational demand low, the kernel used by WLS is rather rudimentary: it is a symmetric 5-step Olympic podium. Informally, the kernel puts a weight of 1 on observation t, a weight of $\zeta < 1$ for observations $t-1$ and $t+1$ and a weight of ζ^2 for observations $t-2$ and $t+2$. Since some specific t's will come up many times (for instance, if both observations t and $t+1$ are within the same leaf, podiums overlap), I take the maximal weight allocated to t as the final weight $w(t;\zeta)$.

Formally, define l_{-1} as the "lagged" version of leaf l. In other words, l_{-1} is a set containing each observation from l, with all of them lagged one step. l_{+1} is the "forwarded" version. l_{-2} and l_{+2} are two-steps equivalents. For a given candidate subsample l, the podium is

$$w(t;\zeta) = \begin{cases} 1, & \text{if } t \in l \\ \zeta, & \text{if } t \in (l_{+1} \cup l_{-1})/l \\ \zeta^2, & \text{if } t \in (l_{+2} \cup l_{-2})/\left(l \cup (l_{+1} \cup l_{-1})\right) \\ 0, & \text{otherwise} \end{cases}$$

where $\zeta < 1$, a tuning parameter guiding the level of time-smoothing. Then, it is only a matter of how to include those additional (but down weighted) observations in the tree search procedure. The usual candidate splitting sets

$$l_1(j,c) \equiv \{t \in l | S_{j,t} \leq c\} \quad \text{and} \quad l_2(j,c) \equiv \{t \in l | S_{j,t} > c\}$$

are expanded to include all observations of relevance to the podium

$$\text{for } i = 1,2: \quad l_i^{RW}(j,c) \equiv l_i(j,c) \ \cup \ l_i(j,c)_{-1} \ \cup \ l_i(j,c)_{+1} \ \cup \ l_i(j,c)_{-2} \ \cup \ l_i(j,c)_{+2}.$$

The splitting rule becomes

$$\min_{j\in\mathcal{J}^-,\ c\in\mathbb{R}}\left[\min_{\beta_1}\sum_{t\in l_1^{RW}(j,c)} w(t;\zeta)\left(y_t - X_t\beta_1\right)^2 + \lambda\|\beta_1\|_2 + \min_{\beta_2}\sum_{t\in l_2^{RW}(j,c)} w(t;\zeta)\left(y_t - X_t\beta_2\right)^2 + \lambda\|\beta_2\|_2\right]. \tag{2.2}$$

Note that the Ridge penalty is kept in anyway, so the final model has in fact two sources of regularization. With $\zeta \to 0$, we are heading back to pure Ridge.

Although not considered in the main applications of this chapter, models with a larger linear part X_t are possible. For instance, one could estimate, equation by equation, a high-dimensional VAR. In practice, this simply requires harsher regularization via higher values of λ, ζ and a larger minimum leaf size. Nevertheless, the forecasting benefits from this strategy could prove limited: MRF is "high-dimensional" whenever S_t is large. The time-varying constant in MRF is a RF in its own right. It can be seen as a complex misspecification function (in the deep learning jargon, it is effectively called the bias) that adaptively controls for omitted variables in a way that is both non-linear and strongly regularized via randomization. Consequently, the cost from omitting a regressor of minor importance in X_t is low since it can be picked up by the time-varying intercept.

Of course, the small X_t strategy treats the extra regressors as exogenous, which could be at odds with some researchers' will to investigate a large web of impulse response functions. Anyhow, both approaches are possible. The dynamic coefficients of a (large) GTVP-VARs can be estimated by either fitting MRF equation by equation, or modifying the splitting rule in (2.2) to be multivariate so that each tree is fitted jointly for all equation – pooling time-variation across equations. Finally, elements of the covariance matrix of residuals can be fitted separately with a plain RF, which is very fast.

2.2.4. Relationship to Random Walk Time-Varying Parameters

GTVPs have many advantages over classical TVPs. While it is known that any nonlinear model can be approximated by a linear one with TVPs (Granger, 2008), nothing is said about how efficient that estimation is going to be. As it turns out, efficiency crucially matters in a macro context, and random-walk TVPs can be quite inefficient (Aruoba et al., 2017). For example, if the true β_t follows a recurrent switching mechanism, random walk parameters already have two strikes against them. Some dimensionality reduction techniques – like reduced-rank restrictions (de Wind and Gambetti, 2014; Stevanovic, 2016; Chan et al., 2018; Goulet Coulombe, 2020a) – can help, but nothing in that paradigm can come close to the parsimony of simply interacting X_t with relevant variables. In contrast,

MRF considers all time-variations options, and choose the "obvious thing", which may or may not be splitting on t. Also, it is absolutely possible that the resulting $\mathcal{F}$ pools *both* latent and observable time variation.

Even though MRF is remarkably flexible, its variance remains low thanks to the diversified portfolio of trees. The variance of classical TVPs can be controlled by cross-validation (Goulet Coulombe, 2020a) or via an elaborate hierarchical prior (Amir-Ahmadi et al., 2018). A number of applications opt for a "manual" approach (D'Agostino et al., 2013). However, it is understood that no tuning, however careful it may be, can overcome the hardship of fitting random-walks when the true β_t's look nothing like it.

Econometrically, one way to more formally connect this paradigm to recent work on TVPs is to adopt the view that RF are adaptive kernel estimators (Meinshausen, 2006; Athey et al., 2019; Friedberg et al., 2018). That is, the tree ensemble is a machine generating kernel weights. Once those are obtained, estimation amounts to weighted least squares (WLS) problem with a Ridge penalty. By running (2.1) recursively, one obtains terminal nodes/leaves $L_b()$ to construct kernel weights

$$\alpha_t\left(\mathrm{x}_0\right) = \frac{1}{B}\sum_{b=1}^{B}\frac{1\left\{\mathrm{X_t} \in L_b\left(\mathrm{x}_0\right)\right\}}{\left|L_b\left(\mathrm{x}_0\right)\right|}$$

to use in

$$\forall t \quad : \text{argmin}_{\beta_t}\left\{\sum_{\tau=1}^{T}\alpha_t\left(\mathbf{s}_\tau\right)\left(Y_\tau - X_\tau\beta_\tau\right)\right)^2 + \lambda\|\beta_t\|_2\right\}. \tag{2.3}$$

As shown in Chapter 1, standard random walk TVPs are in fact a smoothing splines problem, and for those, a reproducing kernel exists (Dagum and Bianconcini, 2009a). Giraitis et al. (2014) drop the random walk altogether and proposed to use kernels directly. Anyhow, in both cases, the only variable entering the kernel is t. In other words, only proximity in time is considered for the clustering of observations. This makes the seemingly flexible estimator in fact quite restrictive – and dependent on its inherent smoothness prior. Moreover, standard kernel methods are known to break down even in medium dimensions (say <10 variables) (Friedberg et al., 2018). Therefore, augmenting t – the sole variable in the kernel (implicit of explicit) of traditional TVP methods – with additional regressors is not an option. No such constraints bind on the RF approach.

2.2.5. Relationship to Standard Random Forest

The standard RF is a restricted version of MRF where $X_t = \iota$, $\lambda = 0$ and $\zeta = 0$. In words, the only regressor is a constant and there is no within-leaf shrinkage. Previous sections motivated MRF as a natural generalization of non-linear time-series models. At this point,

a reasonable question emerges from a ML standpoint. Why should we prefer the partially linear MRF to the fully nonparametric RF? One reason is statistical efficiency. The other is potential for interpretation.

Smooth Relationships are Hard Relationships (to estimate)

In finite samples, plain RF can have a hard time learning smooth relationships – like a AR(1) process. This is bad news for time series applications. For prediction purposes, estimating

$$y_t = \phi y_{t-1} + \varepsilon_t$$

by OLS implies a single parameter. However, approximating the same relationship with a tree (or an ensemble of them) is far more consuming in terms of degrees of freedom. To get close to the straight line once parsimoniously parametrized by ϕ, we now need a succession of many step functions.[9] With short time series, modeling smooth/linear relationships in such a way is a luxury one rarely can afford. The mechanical consequence is that RF will waste many splits on capturing the linear part, and may run out of them before it gets to focus more subtle nonlinear phenomena.[10] In a language more familiar to economists, this is simply running out (quickly) of degrees of freedom. MRF provides a workaround. Modeling the linear part concisely leaves more room to estimate the nonlinear one. By its more strategic budgeting of degrees of freedom, the resulting (estimated) partially linear model could be, in fact, more non-linear than the fully nonparametric one.

This chapter is not the first to recognize the potential need for a linear part in tree-based models. For instance, both Alexander and Grimshaw (1996) and Wang and Witten (1996) proposed linear regressions within a leaf of a tree, respectively denominated "Treed Regression" and "Model Trees". More focused on real activity forecasting, Woloszko (2020) and Wochner (2020) blend insights from macroeconomics to build better-performing tree-based models.[11] On a different end of the econometrics spectrum, Friedberg et al. (2018) proposed to improve the nonparametric estimation of treatment effect heterogeneity by combining those ideas developed for trees into a forest.[12] To my knowledge, this chapter is the first to exploit the link between this strand of work and the sempiternal search for the "true" state-dependence in empirical macroeconomic models.

[9] In a standard regression setup, nobody would model a continuous variable as an ordinal one unless some wild nonlinearities are suspected.

[10] One necessary (but not sufficient) symptom is AR terms being flagged as really important by typical RF variable importance measures (one example is Borup et al. (2020b)).

[11] Specifically, Wochner (2020) also note that using trees in conjunction with factor models can improve GDP forecasting. An analogous finding will be reported in section 2.4.

[12] More broadly, this is extending to trees and ensemble of trees the "classical" non-parametrics literature's knowledge that local linear regression usually has much better properties (especially at the sample boundaries) than the Naradaya-Watson estimator.

A Note on Interpretability

The interpretation of ML outputs is now a field of its own (Molnar, 2019). RF is widely regarded as a black box model which needs to be interpreted using an external device. Indeed, it usually averages over 100 trees of substantial depth, which makes individual inspection impossible. MRFs partially circumvent the problem by providing time series β_t which can be examined, and have a meaning as time-varying parameters for the linear model. Thus, whatever one may do with TVPs, it can be done with GTVPs. There are also some new avenues. For instance, Variable Importance (VI) measures usually deployed to dissect RF's prediction can be used to inspect what is driving β_t's. Those will be used in section 2.5.3.

A popular approach to dissect a standard RF is to use interpretable surrogate tree models to partially replicate the black box model's fit. The idea can be transferred to MRF (Molnar, 2019). In fact, partial linearity facilitates such an exercise. The linear part in MRF splits the nonparametric atom into different pieces ($X_{t,k}\beta_{t,k}$) which can be analyzed separately. Each time series $\beta_{t,k}$ can be dissected with its own surrogate model, and meaningful combination/transformations of coefficients can be considered.

2.2.6. Engineering S_t

This section discusses principles guiding the composition of S_t, which is the raw material for $\mathcal{F}$ in both MRF and plain RF. Macroeconomic data sets (e.g. FRED, McCracken and Ng 2020) typically contains many regressors and few observations. After incorporating lags for each variable, it can easily be the case that predictors outnumber observations. The curse of dimensionality has both computational and statistical ramifications. The former is mostly avoided in RF since it does not rely on inverting a matrix. However, the statistical curse of dimensionality, a feature of the regressors/observations ratio, remains a difficulty to overcome.

There are two extreme ways of reducing dimensionality: sparse or dense. The former selects a small number of features out of the large pool in a supervised way (e.g. LASSO), the latter compresses the data in a set of latent factors that should span most of the original regressors space. This is often seen as a necessity to choose *one* of them.[13] However, in a regularized model, both can be included, and we can let the algorithm select an optimal combination of original features and factors.[14] This is useful — it is not hard to imagine a situation where opting for one or the other would prove suboptimal to a more nuanced solution.

[13] In macro forecasting work using RF, Goulet Coulombe et al. (2019) follow a dense approach by only including factors while Borup et al. (2020a) opt for sparsity by proposing a Lasso pre-selection step.

[14] A more detailed discussion of this can be found in Appendix 2.7.1.

LAG POLYNOMIALS. From a predictive standpoint, residuals autocorrelation implies there is forecasting power left on the table. To get rid of it, many lags might be necessary. In multivariate contexts (like that of a VAR), doing so quickly pushes the model to overfit. A standard solution is Bayesian estimation and the use of priors in the line of Doan et al. (1984), which are specially designed for blocks of lags structures. Outside of the VAR paradigm, there is an older literature estimating restricted/regularized lag polynomials in Autoregressive Distributed Lags (ARDL) models (Almon, 1965; Shiller, 1973). More recently, these methods have found new applications in mixed-frequency models (Ghysels et al., 2007) where the design of the model leads to an explosion of lag parameters.

(M)RF experiences an analogous situation. A tree may waste many splits trying to efficiently extract information out of a lag polynomial: for instance, splitting on the first lag, then the 7th one, then the 3rd one. In linear parametric models, the above methods can extract the relevant information out of a lag polynomial without sacrificing many degrees of freedom. A significant roadblock to this enterprise in the RF paradigm is that there are no explicit lag polynomials to penalize. An alternative route is to exploit the insight that RF can choose for itself relevant restrictions. We just have to construct regressors that embodies those, and include them in S_t.

MOVING AVERAGE FACTORS. To extract the essential information out of the lag polynomial of a specific variable, a linear transformation can do the job. Consider forming a panel of P lags of variable j:

$$X_{t,j}^{1:P} \equiv [X_{t-1,j} \ \ldots \ X_{t-P,j}] \ .$$

We want to form weighted averages of the P lags so that it summarizes most efficiently the temporal information of the feature indexed by j.[15] The weighted averages with that property will be the first few factors (extracted by PCA) of $X_{t,j}^{1:P}$.[16] This can be seen as the time-dimension analog to the traditional cross-sectional factors. The latter are defined such as to maximize their capacity to replicate the cross-sectional distribution of $X_{t,j}$ fixing t while the Moving Average Factors (MAFs) proposed here seek to represent the temporal distribution of $X_{t,j}$ for a fixed j in a lower-dimensional space.[17] By doing so, our goal to summarize the information of $X_{t,j}^{1:P}$ without modifying the RF algorithm (or any other) is

[15] P is a tuning parameter the same way the set of included variables in a standard factor model is one.

[16] While I work directly with the latent factors, a related decomposition called singular spectrum analysis works with the estimate of the *summed* common components. Since this decomposition naturally yields a recursive formula, it has been used to forecast macroeconomic and financial variables (Hassani et al., 2009, 2013).

[17] In the spirit of the Minnesota prior, one can assign decaying (in p) weights to each lag before running PCA. This has the analogous effect of shrinking more heavily the distant lags and less so the recent ones.

achieved: rather than using the numerous lags as regressors, we can use the MAFs which compress information ex-ante. As it is the case for standard factors, MAF are designed to maximize the explained variance in $X_{t,j}^{1:P}$, not the fit of the final target. It is the RF part's job to select the relevant linear combinations among S_t so to maximize the fit. Finally, it is noteworthy that MAFs facilitate interpretation. As these are moderately sophisticated averages of a single time series, they can be viewed as a smooth index for a specific (but tangible) economic indicator. This is arguably much easier to interpret than a plethora of lags coefficients.

The take-away message from this subsection can be summarized in three points. First, there is no need to choose ex-ante between sparse and dense when the model performs selection/regularization. We can let the algorithm find the optimal balance. Second, to make the inclusion of many lags useful, we need to regularize the lag polynomial. Third, such compression can be achieved most easily by generating MAFs and using those as regressors in RF – or any algorithm.

2.2.7. Quantifying Uncertainty of β_t's Estimates

Taddy et al. (2015) and Taddy et al. (2016) interpret RF's prediction as the posterior mean of a tree functional $\mathcal{T}$ (the splitting algorithm) obtained by an approximate Bayesian bootstrap.[18] Through those lenses, each tree is a posterior draw. Seeing $\mathcal{T}$ as a Bayesian non-parametric statistic (independently of the DGP) is of even greater interest in the case of MRF.[19] It provides inference for meaningful time-varying parameters β_t rather than an opaque conditional mean function. Such techniques, originating from Ferguson (1973), have seldomly found applications in econometrics, such as Chamberlain and Imbens (2003) for instrumental variable and quantile regressions.

While the Bayesian Bootstrap desirably does not assume many things about the data, it yet makes the assumption that $Z_t = [y_t \;\; X_t \;\; S_t]$ is an *iid* random variable. Thus, it cannot be used directly as a proper theoretical motivation for using the bag of trees directly to conduct inference. I propose a block extension to make Taddy et al. (2015)'s convenient approach amenable to this chapter's setup.

Block Bayesian Bootstrap (BBB) is a simple redefinition of Z so that it is plausibly *iid*. Hence, in the spirit of traditional frequentist block bootstrap (MacKinnon, 2006), blocks

[18] The connection between Breiman (1996)'s bagging and Rubin (1981)'s Bayesian Bootstrap was acknowledged earlier in Clyde and Lee (2001).

[19] An alternative (frequentist) inferential approach is that of Friedberg et al. (2018). However, their asymptotic argument requires estimating the linear coefficients and the kernel weights on two different subsamples. This is hard to reconciliate with our goal of modeling time-variation and different regimes throughout the entire sample. Furthermore, when the sample size is small, splitting the sample in such a way carries binding limitations on the complexity of the estimated function.

of a well-chosen size will be exchangeable. Thus, a new variable can be defined $Z_{\mathfrak{b}} \equiv [y_{\underline{\mathfrak{b}}:\bar{\mathfrak{b}}} \;\; X_{\underline{\mathfrak{b}}:\bar{\mathfrak{b}}} \;\; S_{\underline{\mathfrak{b}}:\bar{\mathfrak{b}}}]$. There will be a total of $\mathfrak{B} = {}^{T}/_{\text{block size}}$ fixed and non-overlapping blocks. Under covariance stationarity, $\tilde{Z}_{\mathfrak{b}} = vec(Z_{\mathfrak{b}})$ are *iid*, for a properly chosen block length.[20] Analogously to Taddy et al. (2015), block-subsampling is preferred to BBB in implementations since it is faster and gives nearly identical results. Details of BB and BBB are available in Appendix 2.7.2.

It is reasonable to wonder how the above procedure deals with the possible presence of heteroscedasticity. Fortunately, the nonparametric bootstrap/subsampling that RF uses is in fact the "pairs" bootstrap of Freedman et al. (1981) which is valid under general forms of heteroscedasticity (MacKinnon, 2006). From a Bayesian point of view, Lancaster (2003) show that the obtained variance for OLS from using such a bootstrap is asymptotically equivalent to that of White's sandwich formula.[21] Hence, in the spirit of heteroscedasticity-robust estimation, no attempt will be made at directly evolving volatility (which is a GLS approach). Rather, it will be reflected in larger bands for periods of smaller signal-to-noise ratio.

2.3. Simulations

Simulations are divided in two parts. The first shows that Autoregressive Random Forest (ARRF) delivers forecasting gains over standard nonlinear time series model when the true DGP mixes both endogenous and exogenous time-variation. Moreover, the former is very resiliant against traditional approaches, even when the DGP matches the latter's restrictive assumptions. Additionally, those simulations will numerically document the superiority of ARRF over RF when the AR part is pervasive (as discussed in section 2.2.5). Overall, this helps rationalizing forecasting results from section 2.4, where ARRF supplants ∼TARs for the vast majority of targets.

The second simulations section considers simpler linear parts and look at how the algorithm behaves when S_t is large. Further, I focus on β_t itself and its credible regions. The main point is to visually show that (i) GTVPs adapts nicely to a wide range of DGPs and (ii) are not prone to discover inexistent time-variation.

2.3.1. Comparison of ARRF to Traditional Nonlinear Autoregressions

I consider 3 DGPs: Autoregression (AR), Self-Exciting Threshold ARs (SETAR), and a SETAR model that collapse to an AR (via a structural break). Those DGPs include two types of time variations, endogenous (y_{t-1}) and exogenous (t).[22] They are meant to encapsulate

[20] In practice, I will use block of two years for both quarterly or monthly data.

[21] Poirier (2011) propose better priors and Karabatsos (2016) incorporate such ideas into a generalized ridge regression.

[22] Since a structural break is just a threshold effect with respect to variable t, one can conclude without loss of generality that similar results would be obtained using different additional switching variables.

compactly the usual nonlinearities considered in empirical studies, like dependence on the state of the business cycle (Auerbach and Gorodnichenko, 2012a; Ramey and Zubairy, 2018b) and exogenous time variation (Clarida et al., 2000).

For all DGPs, $X_t = [1 \;\; y_{t-1} \;\; y_{t-2}]$. The simulated series sample size is either $T = 150$ or $T = 300$. The last 40 observations of each sample consist the hold-out sample for evaluation. I forecast 4 different horizons: $h = 1, 2, 3, 4$. Models are estimated once at the last available data point.

MODELS. SETAR, Rolling-Window (RW) AR, Random Forest (RF) and Autoregressive Random Forest (ARRF) are included. Iterated SETAR forecasts are obtained via the standard bootstrap method (Clements and Smith, 1997) and all the others are generated via direct forecasting. That is, in the latter case, I fit the model directly on y_{t+h} rather than iterating forward the one-step ahead forecast. To certify that the observed differences between SETAR and other models is not merely due to the choice of iterated vs direct forecasts – a non-trivial choice in many environments (Chevillon, 2007) –, I also include SETAR-d where "d" means its forecasts were alternatively obtained by direct forecasting.

In all simulations, MRF's S_t includes 8 lags of y_t and a time trend, which match what will be referred to in section 2.4 as "Tiny ARRF". Thus, unlike ∼TARs, it is "allowed" to split on what we know (by the DGP choices) to be useless regressors (especially at horizon $h = 1$). The specified linear part for all models matches that of the true DGP ($X_t = [1 \;\; y_{t-1} \;\; y_{t-2}]$).

PERFORMANCE METRIC. Performance is evaluated using the mean squared prediction error (MSPE). In simulation s, for the forecasted value at time t made h steps ahead, I compute

$$RMSE_{h,m} = \sqrt{\frac{1}{40 \times 100} \sum_{s=1}^{100} \sum_{t \in \text{OOS}} (y_t^s - \hat{y}_{t-h}^{s,h,m})^2}.$$

100 different simulations are considered, which means the total number of squared errors being averaged for a given horizon and model is 100×40=4000. To provide a visually useful normalization, bar plots report $RMSE_{h,m}$'s relative to that of the oracle, who knows perfectly the law of motion of time-varying parameters β_t.[23] Formally, the metric is $\Delta_o RMSE_{h,m} = {}^{RMSE_{h,m}}/_{RMSE_{h,o}} - 1$.

SETAR MORPHING INSTANTLY INTO AR(2). The two sources of time-variation are com-

[23] Precisely, if the model has a break and a switching variable, it knows exactly the break points, thresholds and AR parameter values in each regime. The only things the oracle does not know are the future shocks (ϵ_{t+h}), and the out-of-sample evolution of parameters (β_{t+h}) – unless the latter is purely deterministic.

bined to display MRF's edge in this not so implausible situation. Further,

$$\text{DGP 1} = \begin{cases} \text{SETAR}, & \text{if } t < T/2 \\ \text{AR}, & \text{otherwise} \end{cases}$$

can rightfully be hypothesized for some economic time series: complex dynamics up until the mid-1980's followed by a very simple autoregressive structure during the Great Moderation.[24] In Figure 7a, MRF comes out as the best model for all horizons in the smaller sample. RF fails particularly at short horizons because it attempts to model all dynamics nonparametrically. Doubling the sample size helps, but its disimprovement with respect to the oracle remains at least twice as large as that of MRF. SETAR and AR both focus on dynamics but are misspecified. Their increase in relative RMSE is about thrice that of MRF at longer horizons for the shorter sample. For horizon 1, RW-AR does equally well as MRF, which is expected in this DGP since it discards earlier observations we only know ex-post to be harmful. Thus, in this DGP much akin to that of the hypothetical inflation tree of section 2.2.1, MRF comes out as the clear winner.

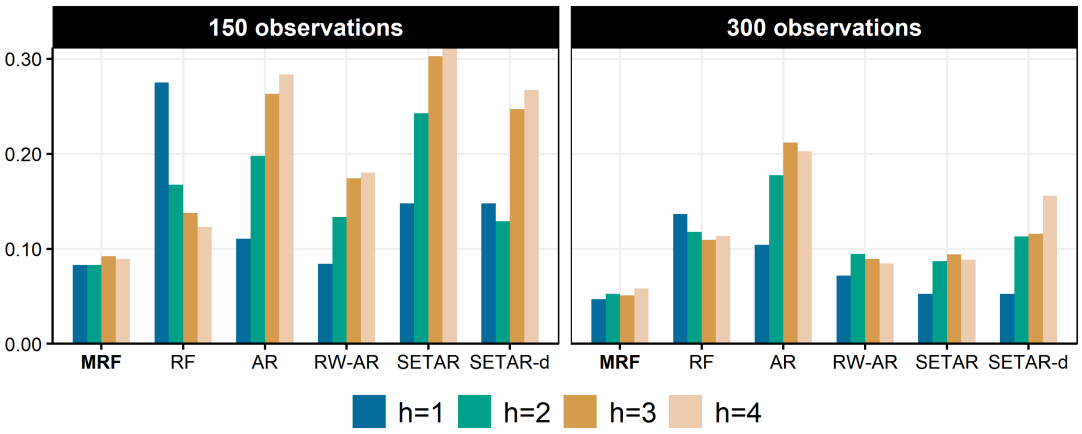
PERSISTENT SETAR. DGP 2, with $\beta_t = I(y_{t-1} \geq 0)[2 \;\; 0.8 \;\; -0.2] + I(y_{t-1} < 0)[0.25 \;\; 1.1 \;\; -0.4]$, represents an endogenous switching process which may suit well real activity variables: it includes high/low regimes, and mildly different dynamics in each of them. Can MRF match traditional nonlinear times series model when the world is nonlinear, yet simple? The broad answer from Figure 7b is yes. For all horizons and sample sizes considered, MRF is practically as good as SETAR, the optimal model in this context. Because of its capacity to control overfitting, MRF will be competitive even if nonlinearities turn out to be as simple as often postulated in the empirical macroeconomics literature. With the relative importance of changing persistence, RF cannot match MRF's performance and is trailing behind with RW-AR.[25] Nevertheless, RF improves substantially at shorter horizons when the sample size increase. Finally, AR is resilient at longer horizons but is much worse than MRF and SETAR at shorter ones.

NO TIME-VARIATION. Given the widespread worry that ML-based algorithms can overfit, a time-invariant DGP is a natural check.[26] Can MRF still deliver competitive performance if reality equates simple linear dynamics? Results for DGP 3, an AR(2) process with $\beta = [0 \;\; 0.7 \;\; -0.2]$, are reported in Figure 7c. As expected, AR is the best model for all

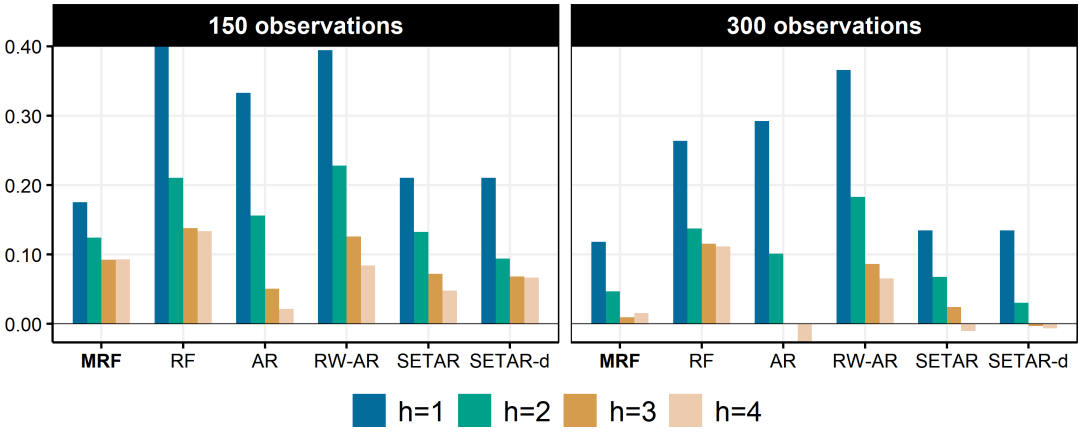
[24] The AR has $\beta = [0 \;\; 0.7 \;\; -0.2]$ and the SETAR has $\beta_t = I(y_{t-1} \geq 1)[2 \;\; 0.8 \;\; -0.2] + I(y_{t-1} < 1)[0.25 \;\; 0.4 \;\; -0.2]$. Results for the latter in isolation are in Appendix 2.7.5.

[25] In Appendix 2.7.5, the case where the changing persistence is less important is considered.

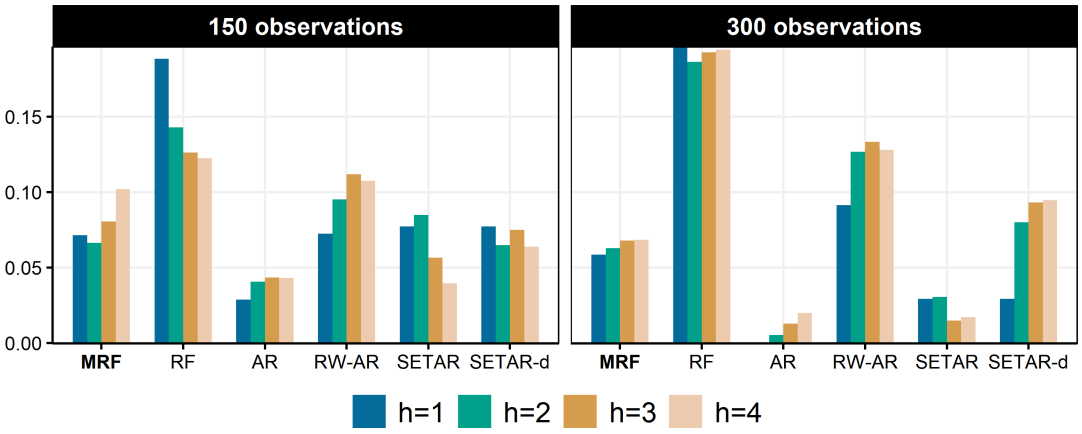
[26] Also, the incredible resilience of linear AR models is well documented in the macroeconomic forecasting literature (see Kotchoni et al. (2019) and references therein).



(a) DGP is SETAR morphing into AR(2).



(b) DGP is Persistant SETAR.



(c) DGP is Plain AR(2).

Figure 7: Displayed are increases in relative RMSE with respect to the oracle.

horizons and both sample sizes. The RW-AR suffers from high variance and it is assumed that tuning the window length in a data-driven way would help. Plain RF struggles irrespective of the sample size. For the smaller sample, MRF performs as well as the tightly parametrized SETARs. Their marginal increases in RMSE with respect to the oracle are typically less than 10%, which is small in contrast to that of previous DGPs. More observations generally helps AR, the iterated SETAR, and MRF especially at longer horizons.

About Misspecification of X_t. Most of the reported gains from using MRF come from avoiding misspecification when a more complex DGP arises. What happens if the arbitrary linear part in MRF, X_t, is itself misspecified? Figure 20 in the appendix report corresponding results. For all DGPs under consideration, a "Bad" MRF, where X_t is composed of two white noise series (instead of the first two lags of y_t), performs similarly well (or bad) as plain RF.[27]

Summary. First, when the true DGP is *not* that of the tightly parametrized classical nonlinear time series model, the more flexible MRF does better. Second, when classical nonlinear time series model are fitted on their corresponding DGPs, those perform better than MRF – but only marginally. Third, when there are pervasive linear autoregressive relationships, plain RF struggles. Fourth, MRF and RF relative performance both increase with the number of observations but MRF's one increases faster if the linear part is well-chosen. In Appendix 2.7.5, results for 3 additional DGPs are reported: another SETAR, AR with a structural break, and SETAR morphing in another SETAR (through a break). Again, MRF is shown to have an edge when other models are misspecified and almost as good when those are not.

2.3.2. A Look at GTVPs when S_t is Large

A notable difference between the simulations presented up to now and the applied work being carried in later sections is the size of S_t. In many macro applications, there is no shortage of variables to include in MRF's $\mathcal{F}$. For instance, the FRED-QD data base (McCracken and Ng, 2020) contains over 200 potential predictors that can join lags of y and a time trend within S_t. As a result, there is now considerable interest in allowing for time variation in empirical models using large information sets. For instance, Koop and Korobilis (2013) propose large TVP-VARs while Abbate et al. (2016) extend Bernanke et al. (2005)'s factor-augmented VAR to be time-varying. Interestingly, those papers (and the corresponding literature) almost exclusively focus on a setup where, in MRF notation, X_t

[27] This result may not hold, however, when the law of motion for the intercept is highly complex and requires a great number of split (unlike what is considered here). This is due to the linear part restricting the depth of trees (with to what plain RF could allow for), especially if observations are scarce. In that regard, increasing the ridge penalty (via λ) will help. Nevertheless, in practice, it is a safer bet to use a small linear part if uncertainty around its composition is high. More on this and the effect of hyperparameters can found in appendix 2.7.4.

is large and S_t is extremely small (usually just t). Of course, MRF could deal with this case (as discussed in section 2.2.3), but its edge will be more apparent when we let the RF part deal with large data and keep X_t concise. Indeed, in addition to lessened misspecification concerns, RFs also benefits from more data through increased randomization – which prevents fully grown trees from overfitting (Breiman, 2001; Goulet Coulombe, 2020c).

The additional simulations go as follow. First, I simplify the analysis by looking at a static model with mutually orthogonal but autocorrelated regressors X_1 and X_2, both driving y_t according to some process. I simulate each of them for 1000 periods and estimate the models with the first 400 observations. The remaining 600 are used to evaluate the out-of-sample performance. The signal to noise ratio is calibrated to $2/3$ which is about what is found (out-of-sample) for most models in the empirical section.

The only remaining questions are that of the constitution of S_t and the generation of β_t's. I create two autocorrelated (but not cross-correlated) factors. Out of each of them, I create 50 series with a varying amount of additional white noise.[28] Joining those 100 series with lags of y_t and a time trend, the final size of S_t is slightly above 100. Finally, β_t's are functions of the underlying *first* factor which (like the second) is not directly included in the data set. In certain DGPs, some β_t's will also be a pure function of t (like random walks, structural breaks).[29] Table 16 summarizes the six DGPs in words.

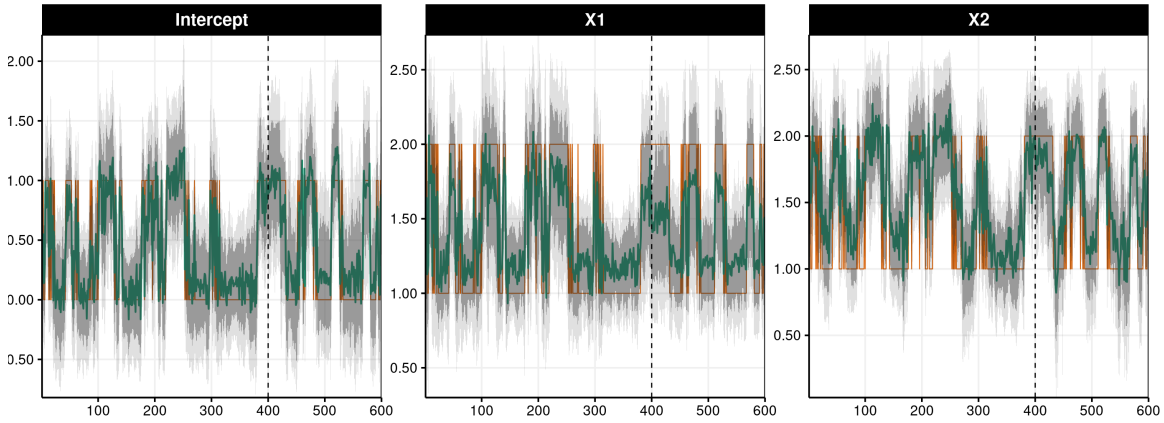
Table 16: Summary of Data-Rich Simulations DGPs

DGP #	Intercept	$\beta_t^{X_1}$	$\beta_t^{X_2}$	Residuals Variance
1	Switching	Switching	Switching	Flat
2	Flat	Random Walk	Random Walk	Flat
3	Flat	Latent factor directly	Slow Change (function of t)	Flat
4	Flat	Switching	Slow Change (function of t)	Flat
5	Flat	Switching	Structural Break	Flat
6	Flat	Flat	Flat	Stochastic Volatility

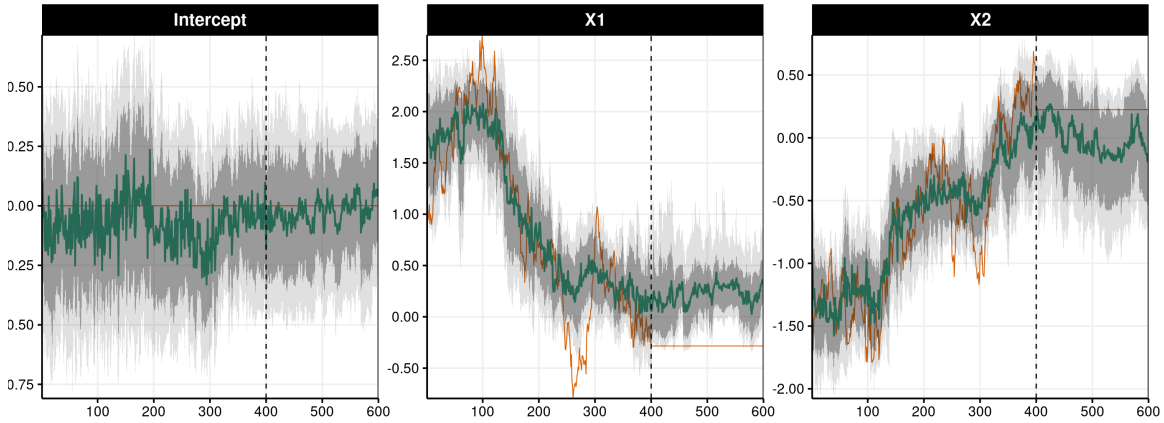
More illustratively, Figures 8 and 21 (appendix) plots one example of each DGPs as well as the estimated GTVPs and their credible region (as discussed in section 2.2.7). It is visually obvious that GTVPs are adaptive in the sense that it can discover which kind of time-variation is present in the data while estimating it. In Figure 8a, MRF successfully estimate the rather radical switching regimes present in all coefficients. In Figure 8b, MRF realizes that almost all of S_t is useless because true β_t follow random walks. Rather, it manages to fit β_t's nicely by relying on a multitude of t splits. In Figure 8c, things get "easier" for the

[28] To be precise, their standard deviation is $U[0.5, 3]\%$ that of the original factor standard deviation.

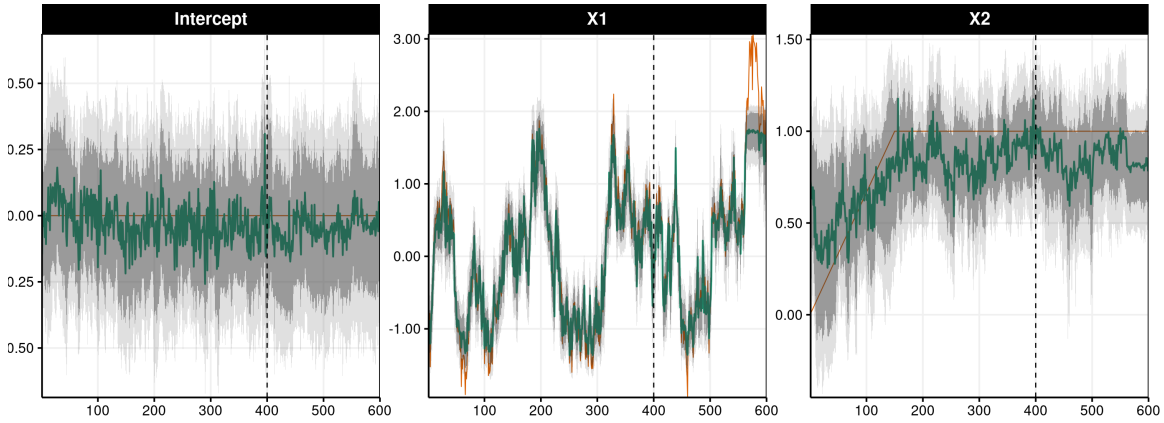
[29] To clarify, the second factor and underlying series are completely useless to the true DGP – arguably mimicking the inevitable when using a data base of the size of FRED-QD.



(a) DGP 1



(b) DGP 2



(c) DGP 3

Figure 8: The grey bands are the 68% and 90% credible region. After the blue line is the hold-out sample. Green line is the posterior mean and orange is the truth. The plots include only the first 400 observations for visual convenience.

true $\beta_{X_1,t}$ as it is driven directly by the first latent factor. MRF discovers that and leverage it to have a very tight fit for it, both in-sample and out-of-sample. This is merely a reflection that if time variation can be constructed by simple interaction terms, this is certainly the easiest statistical route to by – and MRF chooses it thanks to its inherent ability to perform "time variation selection".

Figure 22 reports distributions of RMSE differentials with respect the oracle (the forecast that knows the β_t's law of motion). MRF performance is compared to OLS, Rolling-Window OLS (RW-OLS) and plain RF. As expected, MRF outperforms all alternatives by wide margins for most DGPs. By construction, RW-OLS and OLS also perform well for DGP 5 (random walks) and DGP 6 (constant parameters). Nonetheless, it is reassuring to see that MRF either performs much better than OLS or worse by a thin margin (in cases with no time-variation).

2.4. Macroeconomic Forecasting

In this section, I present results for a pseudo-out-of-sample forecasting experiment at the quarterly frequency using the dataset FRED-QD (McCracken and Ng, 2020). The latter is publicly available at the Federal Reserve of St-Louis's web site and contains 248 US macroeconomic and financial aggregates observed from 1960Q1. The forecasting targets are real GDP, Unemployment Rate (UR), CPI Inflation (INF), 1-Year Treasury Constant Maturity Rate (IR) and the difference between 10-year Treasury Constant Maturity rate and Federal funds rate (SPREAD). These series are representative macroeconomic indicators of the US economy which is based on Goulet Coulombe et al. (2019) exercise for many ML models, itself based on Kotchoni et al. (2019) and a whole literature of extensive horse races in the spirit of Stock and Watson (1998a). The series transformations to induce stationarity for predictors are indicated in McCracken and Ng (2020). For forecasting targets, GDP, UR, CPI and IR are considered $I(1)$ and are first-differenced. For the first two, the natural logarithm is applied before differencing. SPREAD is kept in "levels". Forecasting horizons are 1, 2, 4, 6 and 8 quarters.

The pseudo-out-of-sample period starts in 2003Q1 and ends 2014Q4. I use expanding window estimation from 1961Q3. Models are estimated (and tuned, when applicable) every two years. For all models except SETAR and STAR, I use direct forecasts, meaning that $\hat{y}_{t+h}$ is obtained by fitting the model directly to y_{t+h} rather than iterating one-step ahead forecasts. $\sim$TAR iterated forecasts are calculated using the block-bootstrap method which is standard in the literature (Clements and Smith, 1997).

Following standard practice, the quality of point forecasts is evaluated using the root Mean Square Prediction Error (MSPE). For the out-of-sample (OOS) forecasted values at time t

of variable v made h steps ahead, I compute

$$RMSE_{v,h,m} = \sqrt{\frac{1}{\#\text{OOS}} \sum_{t \in \text{OOS}} (y_t^v - \hat{y}_{t-h}^{v,h,m})^2}.$$

The standard Diebold and Mariano (2002) (DM) test procedure is used to compare the predictive accuracy of each model against the reference AR(4) model. $RMSE$ is the most natural loss function given that all models are trained to minimize the squared loss in-sample.

It has been argued in section 2.2.6 that feature engineering matters crucially when the number of regressors exceeds the sample size. S_t, the set of variables from which RF can select, is motivated by such concerns. Its exact composition is detailed in Table 17. Among other things, it includes both cross-sectional and moving average factors, which are compressing information along their respective dimensions. The usefulness of MAFs is further studied in Goulet Coulombe et al. (2020a) and found to help, mostly with tree-based algorithms. However, it is supplanted by a more computationally demanding (but more general) transformation of the raw data that Goulet Coulombe et al. (2020a) propose specifically for ML-based macroeconomic forecasting.

Table 17: Composition of S_t

What	Why	How
8 lags of y_t	Endogenous SETAR-like dynamics	–
t	Exogenous "structural" change/breaks	–
2 lags of FRED	Fast switching behavior	–
8 lags of 5 traditional factors F	Compress cross-sectional information ex-ante	Usual PCA
2 MAFs for each variable j	Compress lag polynomial information ex-ante	PCA on 8 lags of j

MODELS. To better understand where the gains from MRF are coming from, I include models that use different subsets of ideas developed in earlier sections. Those are summarized in Table 18. The competitive data-rich models are in the benchmarks group. Non-linear time series models are also included as they share an obvious familiarity with ARRF. "Tiny" versions of both ARRF and RF are considered to gauge the effect from only having access only to a small S_t — this could be the case for many non-US applications. Conversely, this helps quantify how a data-rich environment contributes to the success of ARRF versus its plain flexibility. Indeed, Tiny ARRF corresponds to what was shown in the "data-poor" simulations (section 2.3) to be a generalization of ∼TARs and related models.

Here are some remarks motivating some inclusions and specifications choices. To assess

the marginal effects of MAFs alone, Lasso, Ridge and RF are considered using S_t — those are known to handle high-dimensional feature space. When it comes to FA-ARRF, I opt for a parsimonious linear specification including one lag of the first two factors. First, concise models make interpretation easier. Second, considering compact linear specifications within MRF is usually the better strategy. Parameters (including the intercept) are all RFs in their own right and can palliate to the omission of marginally important features, if need be. Consequently, it is desirable to fix a humble linear part and let β_t's take care of the rest.[30] Finally, as discussed in McCracken and Ng (2020), the first factor mostly loads on real activity variables while the second is a composite of forward-looking indicators like term spreads, permits and inventories. They are baptized and interpreted accordingly.

Table 18: Forecasting Models

Name	Acronym	Linear Part (X_t^m)	RF part
Autoregression	**AR**	$[1,\ y_{t-\{1:4\}}]$	$\emptyset$
Factor-Augmented Autoregression	**FA-AR**	$[1,\ y_{t-\{1:4\}},\ F_{1,t-\{1:2\}},\ F_{2,t-\{1:2\}}]$	$\emptyset$
Plain Random Forest	**RF**	$\emptyset$	Raw data[31]
Low-Dimensional Plain RF	**Tiny RF**	$\emptyset$	$[y_{t-\{1:8\}},\ t]$
Plain RF but using S_t	**RF-MAF**	$\emptyset$	S_t
RF-MAF on de-correlated y_t	**AR+RF**	Filter y_t first with an AR(2), then RF	S_t
Autoregressive Random Forest	**ARRF**	$[1,\ y_{t-\{1:2\}}]$	S_t
Low-Dimensional Autoregressive RF	**Tiny ARRF**	$[1,\ y_{t-\{1:2\}}]$	$[y_{t-\{1:8\}},\ t]$
Factor-Augmented Autoregressive RF	**FA-ARRF**	$[1,\ y_{t-\{1:2\}},\ F_{1,t-1},\ F_{2,t-1}]$	S_t
Vector Autoregressive RF[32]	**VARRF**	$[1,\ y_{t-\{1:2\}},\ GDP_{t-1},\ IR_{t-1},\ INF_{t-1}]$	S_t
Self-Exciting Threshold AR	**SETAR**	$[1,\ y_{t-\{1:2\}}]$	$\emptyset$
Smooth Transition AR[33]	**STAR**	$[1,\ y_{t-\{1:2\}}]$	$\emptyset$
10 years Rolling-Window AR	**RW-AR**	$[1,\ y_{t-\{1:2\}}]$	$\emptyset$
Time-Varying Parameters AR[34]	**TV-AR**	$[1,\ y_{t-\{1:2\}}]$	$\emptyset$
LASSO using S_t	**LASSO-MAF**	S_t	$\emptyset$
Ridge using S_t	**Ridge-MAF**	S_t	$\emptyset$

Notes: models are classified in 3 categories: benchmarks, MRFs (and related prototypes), and misc (non-linear time series models, other reasonable additions). The main analysis in section 2.4.1 omits the 3rd club for parsimony.

2.4.1. Main Quarterly Frequency Results

Violin plots are used throughout to summarize dense RMSEs tables (like Table 20). I report the distribution of $RMSE_{v,h,m}/RMSE_{v,h,AR}$. This is informative about the overall ranking

[30] Further backing a parsimonious choice (with MRF), McCracken and Ng (2020) report that the first two factors account for 30% of the variation in the data while adding two more only bumps it up to 41%, making the last two presumably more disposable in our context.

[31] Precisely, this means 8 lags of FRED-QD, after usual transformations for stationarity have been applied.

[32] Note that the VAR appellation refers to the linear equation consisting of typical "small monetary VAR". The model remains univariate and direct forecasts are used.

[33] The state variable is y_{t-1}, as in SETAR.

[34] Estimated and tuned via the Ridge approach proposed in Chapter 1.

and versatility of considered models. Of course, being ranked first does not imply being the best model for every h and v. Rather, it means that it performs better on average, over all targets.

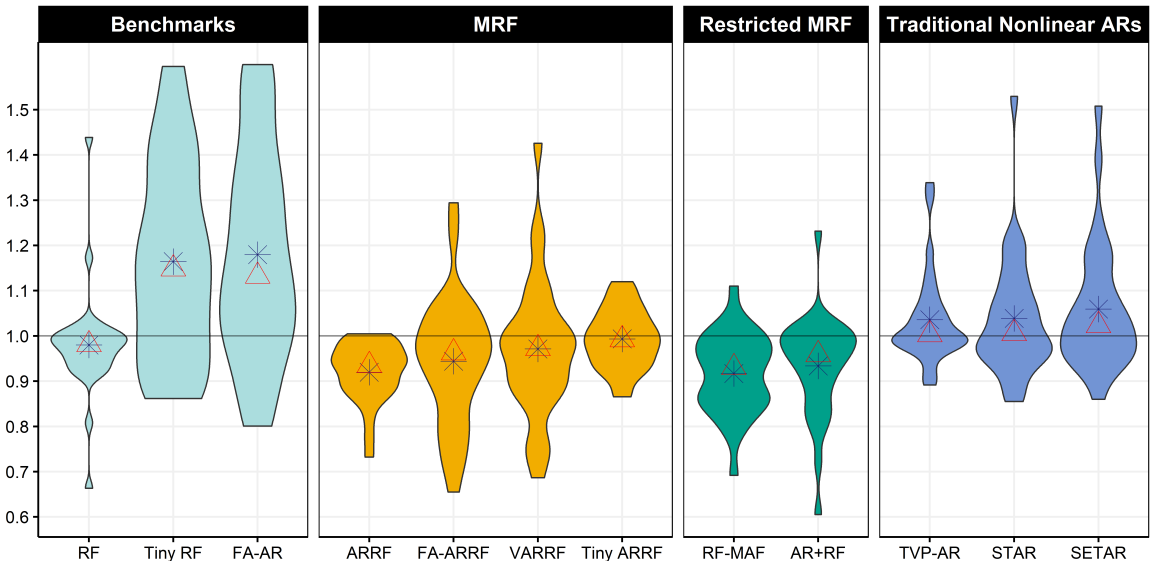


Figure 9: The distribution of $RMSE_{v,h,m}/RMSE_{v,h,AR}$. The star is the mean and the triangle is the median.

Here are interesting observations from Figure 9. Clearly, MRFs deliver important gains over both the AR and FA-AR benchmarks (the latter is second to last). ARRF has a noticeably small mass above the 1 line. In other words, there are no targets for which ARRF does significantly worse than its OLS counterpart, which makes it atypically adaptable among nonlinear autoregressions. A look at Table 20 confirms this observation also extends to FA-ARRF vs FA-AR. The simplification AR+RF, ranks third with a performance that is much more volatile. This suggests that imposing time-invariant dynamics can sometimes help (see one example in Figure 11), but can also be highly detrimental (as reported for inflation). Of course, that we do not know ex-ante, and it is why AR+RF does not inherit ARRF's "off-the-shelf" quality.

MAFs are useful: RF-MAF does much better than RF which uses the raw data. The latter only exhibits conservative gains over the benchmark. Thus, it is understood that a fraction of MRFs' forecasting gains emanates from considering more sensible transformations of time series data – and which are trivially implementable. The relevance of MAFs is studied more systematically in Goulet Coulombe et al. (2020a).

FA-ARRF provides very substantial improvements, but can also fail. This is the linear part's doing: FA-AR will mostly work well for real activity variables while AR is a jack of all trades. Thus, it is not surprising to see FA-ARRF inherit some of these uneven proper-

ties, albeit to a much milder extent. For instance, in Table 20, FA-AR is noticeably worse than AR for all inflation horizons, while FA-ARRF beats AR for all of them. This phenomenon is well summarized by FA-AR being second to last *overall*, well behind FA-ARRF. VARRF has a behavior similar to that of FA-ARRF, but with less highly noticeable gains.

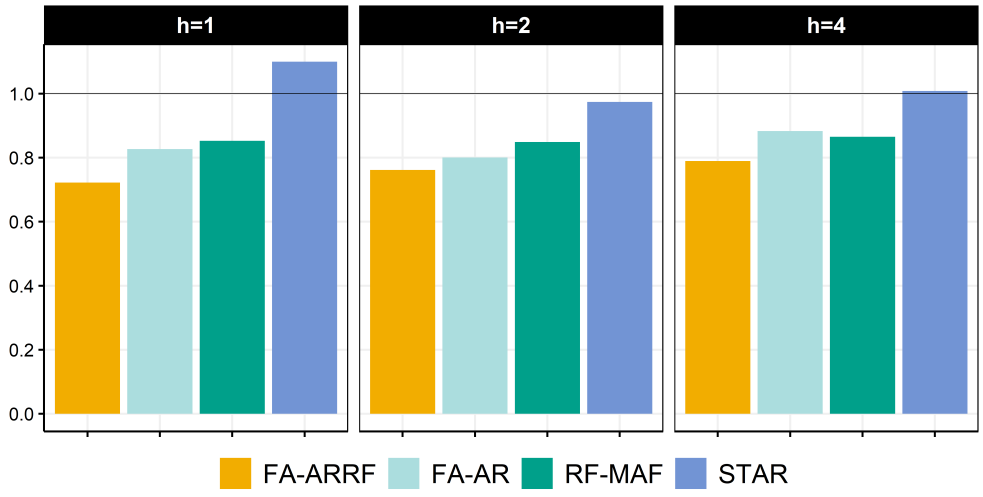
Does a large S_t pay off? Most of the time, yes. It is worth re-emphasizing that restricting S_t restricts the space of time-variations possibilities as well as the potential for trees diversification. Nonetheless, if the restrictions are "true", gains are possible.[35] This is reported to be a rare occurrence, with ARRF $\succ$ Tiny ARRF (and RF $\succ$ Tiny RF) for almost any target. Thus, we can safely conclude that a rich S_t is desirable, with $\mathcal{F}$ being tasked with selection of relevant items.

As discussed in earlier sections, ARRF connects to the wider family of nonlinear autoregressive models. It clearly does better on average than SETAR and Smooth-Transition TAR. This advantage is attributable to both a more flexible law of motion and a large S_t. Tiny ARRF is better than the $\sim$TAR group, while ARRF is *much* better. Linking this result to those of simulations, this means that no $\sim$TAR is likely the true model.

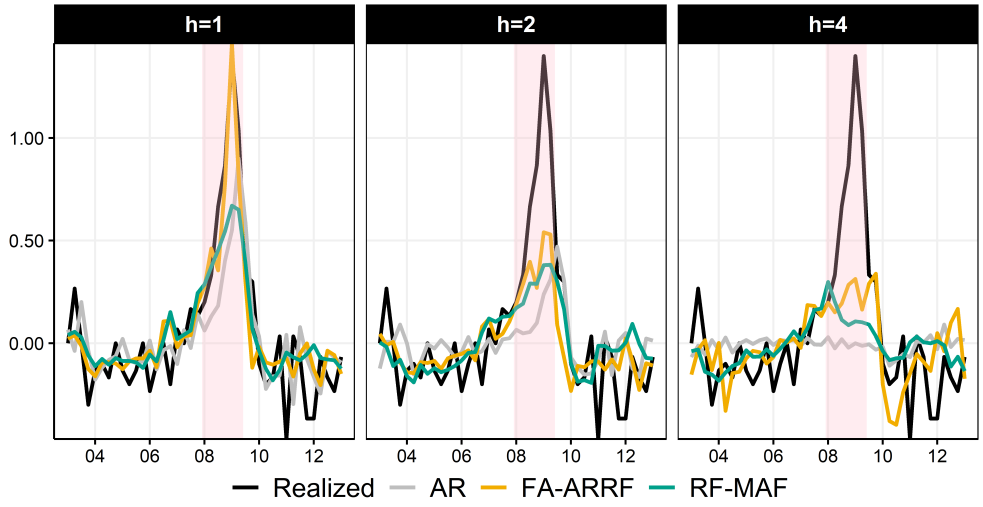
Real Activity Targets. Figure 10 reports results for UR. FA-ARRF dominates strongly. Table 20 confirms it is the best model for all horizons but the last one (8 quarters ahead, where the encompassed RF-MAF is the best). Clearly, at an horizon of one quarter, the preferred model successfully predicts the drastic rise in unemployment during the Great Recession. Rather than responding with a lag to negative shocks (which is what we observe from AR and ARRF), the model visibly predicts them. As a result, improvements in RMSE are between 25% and 30% over AR for all horizons. Specifically, predicting UR (change) with FA-ARRF at $h = 1$ yields an unusually high out-of-sample R^2 of about 80%. The nearly perfect overlap of the yellow and black lines highlight the absence of a one-step ahead shock around 2008. Note that FA-AR and STAR forecasts are omitted from Figure 10b to enhance visibility. STAR forecasts are either similar or worse than the benchmark (as often found for nonlinear time series models). FA-AR forecasts at $h = 1$ follows a proactive pattern similar to the yellow line, but with a 1 to 2 quarter delay – hence the inferior results.

For $h = 2$, the quantitative rise is nowhere near the realized one, but it reveals 6 months ahead the arrival of a significant economic downturn. Additionally, ARRF and FA-ARRF both flag one year ahead the arrival of a rise in unemployment, which is a quality shared by very few models. The barplot in Figure 10 (and Table 20) provides a natural decomposition

[35] An interesting specific case is Tiny ARRF being close behind ARRF for inflation. This is intuitive given that INF has often been associated with exogenous time variation.



(a) $RMSPE_{UR,h,m}/RMSPE_{UR,h,AR(4)}$



(b) A look at some forecasts

Figure 10: Zooming on best model within each group for UR (change)

of FA-ARRF's gains. Adding the MAFs to an otherwise plain RF procures an improvement of roughly 15% across all horizons (RF-MAF $\succ$ RF, in Table 20). The linear FA-AR part and the rest of algorithmic modifications discussed in section 2.2 provide an additional reduction of 10% to 15% depending on the forecasted horizon (FA-ARRF $\succ$ RF-MAF and FA-ARRF $\succ$ FA-AR). It is noteworthy that good results for $h = 1$ are mechanically close to impossible with a plain RF since it cannot extrapolate – i.e., predict values of y_t that did not occur in-sample. In contrast, this is absolutely feasible within MRF thanks to the linear part.

GDP is known to have a lower signal-to-noise ratio. In Figure 24, FA-ARRF exhibits a bit less than a 20% drop in RMSE over the AR and nicely grasp the 2008 drop one quarter

ahead.[36] However, FA-ARRF performance does not stand apart as much as it did for UR. One reason can be traced visually to predicting higher post-recession growth than its competitors. Finally, RF-MAF closing in on ARRF will be investigated on its own in section 2.5.2. In short, this occurs because once the time-varying intercept is flexibly modeled by RF, there is very little room left for autoregressive behavior (at the quarterly frequency).

SPREAD AND INFLATION. VARRF shines for SPREAD (Figure 25) by capturing key movements, even up to a year ahead. The simpler AR+RF also does remarkably well. FA-ARRF provides successful one-year ahead forecasts. Overall, these results highlight the common importance of the autoregressive part, which is no surprise given SPREAD's persistence. For INF, Table 20 displays that RF-MAF is the leading model (with ARRF close behind) reducing RMSEs by 12-15% for all horizons. I investigate this with GTVPs in section 2.5.2.

2.5. Analysis

Based on forecasting results, I concentrate on FA-ARRF's GTVPs. Additionally, its parameters are easier to interpret (given factors are labeled) since regressors are mechanically orthogonal. First, I look at β_t and analyze their behavior around the Great Recession. Second, I compare GTVPs to random walk TVPs, ex-post vs ex-ante, with a focus on the recessionary episode. Finally, I use a surrogate model approach to explain of the parameters' paths in terms of observed variables.

2.5.1. Forecast Anatomy

β_t's characterize completely MRF's forecasts. Thus, we can investigate GTVPs to understand results from the previous section. The FA-ARRF forecasting equation is

$$y_{t+h} = \mu_t + \phi_{1,t} y_t + \phi_{2,t} y_{t-1} + \gamma_{1,t} F_{1,t} + \gamma_{2,t} F_{2,t} + u_{t+h}.$$

and naturally $\beta_t = [\mu_t \;\; \phi_{1,t} \;\; \phi_{2,t} \;\; \gamma_{1,t} \;\; \gamma_{2,t}]$. To avoid overfitting, $\hat{\beta}_t$'s are (as in section 2.3.2) the mean over draws that did not include observations $t-4$ to $t+4$ (a two-year block) in the tree-fitting process. Intuitively, this mimics in-sample the real out-of-sample experiment that starts here in 2007Q2.[37]

Figure 11 displays GTVPs underlying the successful one-step ahead UR *change* forecast. The intercept clearly alternates between at least two regimes and the "increasing UR" one is in effect circa 2008. In levels, this translates to UR alternating between a positive and negative (albeit small) trend. Overall persistence is strikingly time-invariant, and marginally smaller than for OLS estimates. The effect of F_1, the real activity factor, is generally within

[36] Diebold and Rudebusch (1994) proposed an empirically sucessful regime-switching factor model. Given that line of work and more recent results in Wochner (2020), the FA-ARRF's success is not an anomaly.

[37] Note that this is partially different from what gave the results reported in section 2.4.1, where the model was re-estimated every 2 years. Here, estimation occurs once in 2007Q2.

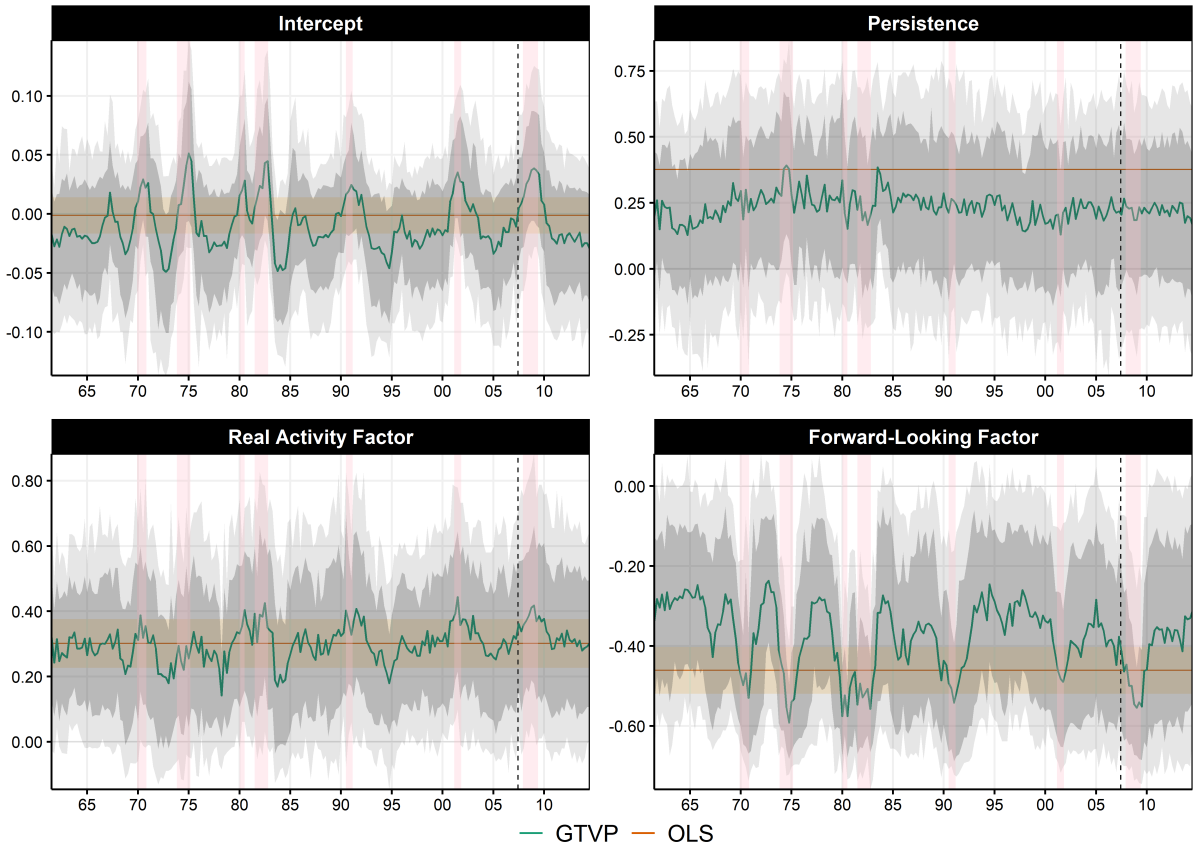


Figure 11: GTVPs of the one quarter ahead UR forecast. Persistence is defined as $\phi_{1,t} + \phi_{2,t}$. The gray bands are the 68% and 90% credible region. The pale orange region is the OLS coefficient $\pm$ one standard error. The vertical dotted blue line is the end of the training sample. Pink shading corresponds to NBER recessions.

OLS confidence intervals, suggesting that while $\gamma_{1,t}$ almost doubles around recessions, this is subject to great uncertainty.

What is less uncertain, however, is the magnified contribution of the forward-looking factor F_2 during recessionary episodes, which stands out as the key difference with OLS. $\gamma_{2,t}$ smooth-switching behavior can be best interpreted by remembering that F_2 is highly correlated with capacity utilization, manufacturing sector indicators, building permits and financial indicators (like spreads) (McCracken and Ng, 2020). Many of those variables are considered "leading" indicators and have often been found to increase forecasting performance, mostly before and during recession periods (Stock and Watson, 1989; Estrella and Mishkin, 1998; Leamer, 2007). Recently, there has been renewed attention on the matter, with financial indicators highlighted as capable of capturing economic activity downside risk (Adrian et al., 2019; Delle Monache et al., 2020). This brand of nonlinearity can translate to a more active $\gamma_{2,t}$ around business cycle turning points. MRF learns that, while OLS

provides a clumsy average of two regimes. In Figure 11, the obvious consequence of OLS' rigidity is being over-responsive to leading indicators during tranquil economic times, and under-responsive when it matters.

Section 2.5.3 will investigate formally the underlying variables driving this time variation. Figure 27 displays equivalent β_t for GDP one quarter ahead. The pattern $\gamma_{2,t}$ is also visible for GDP, but it is quantitatively weaker and more uncertain – which is is no surprise given GDP being generally noisier than UR. Additionally, slow and relatively mild long-run change is observed. Interestingly, $\gamma_{1,t}$ has been shrinking since the mid 1980s, and its regime dependence exhibited in the first four recessions is no more.

2.5.2. Comparing Generalized TVPs with Random Walk TVPs

The relationship between random walk TVPs and GTVPs was evoked earlier. I compare them for the small factor model. I estimate standard TVPs using the ridge regression technique developed Chapter 1. Conveniently, the procedure incorporates a cross-validation step that determines the optimal level of time variation in the random walks.[38]

As Figure 11 suggested for μ_t and $\gamma_{2,t}$, parameters can be subject to recurrent, rapid and statistically meaningful shifts. Such behavior creates difficulties for random-walk TVPs, which put the accent on smooth and slow structural change. Figure 12 confirms this conjecture. Standard TVPs look for long-run change when regime-switching behavior is the main driving force. As a result, they are flat and within OLS confidence bands, as often reported in the literature (D'Agostino et al., 2013). Of course, more action will mechanically be obtained for TVPs when considering a smaller amount of smoothness than what cross-validation proposed. In appendix 2.7.7, I report the same figures, but using the optimal smoothing parameters (as picked by CV) divided by 1000. This provides much more volatile random walk TVPs that are inclined, at certain specific moments, to follow the GTVPs. However, it is clear in Figure 12 that the end-of-sample/revision problem is worsen by the forced lack of smoothing.

It is known in the traditional TVP literature that there is a balance between flexible (but often erratic) β_t paths and very smooth ones where time variation may simply vanish.[39] Since random-walk TVPs are unfit for many forms of the time-variation present in macroeconomic data, high bias estimates are usually reported as only them can keep variance at a manageable level. This can have serious implications. Relying too much on time-smoothing can create a mirage of long-run change and/or dissimulate parameters that

[38] I show with simulations that this much easier approach performs similarly well (and sometimes better) to traditional Bayesian TVP-VAR, for model sizes that the latter is able to estimate.

[39] In the case of ridge regression-based TVPs, cross-validation is just a data-driven way of backing this necessary empirical choice.

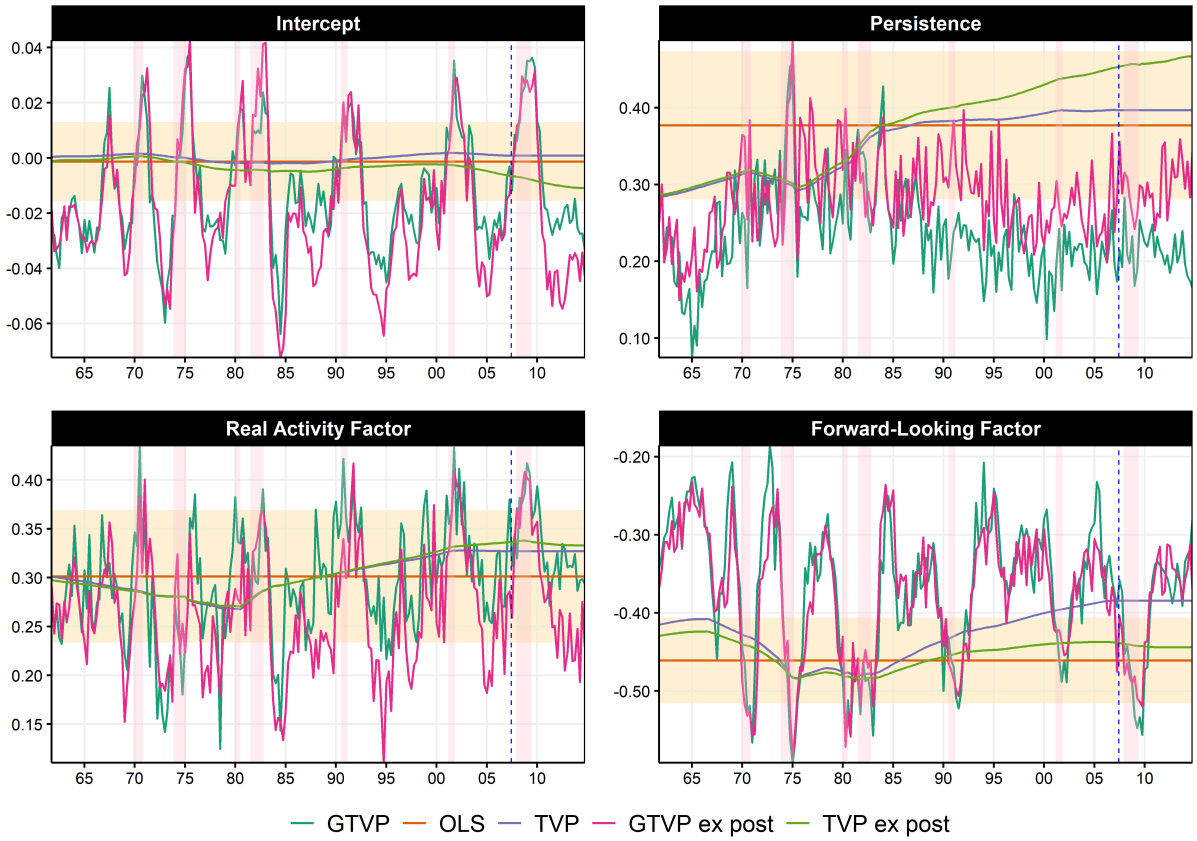


Figure 12: UR equation β_t's obtained with different techniques. Persistence is defined as $\phi_{1,t} + \phi_{2,t}$. TVPs estimated with a ridge regression as in Chapter 1 and the parameter volatility is tuned with k-fold cross-validation — **see Figure 32a for a case where TVP parameter volatility is forced to be higher**. Ex Post TVP means using the full sample for estimation and tuning as opposed to only using pre-2002 data as for GTVPs. The pale orange region is the OLS coefficient $\pm$ one standard error. Pink shading corresponds to NBER recessions.

mostly (but not solely) vary according to expansions/recessions.

Another concern, particularly consequential for forecasting, is the boundary problem. As discussed earlier, random-walk TVP models forecasts can suffer greatly from it because by construction, forecasts are always made at the boundary of the variable on which the kernel is based – i.e., time. One can deploy a 1-sided kernel, but this only alleviate a few pressing symptoms without attacking the heart of the problem. In sharp contrast, GTVPs use a large information set S_t to create the kernel, which implies that the likelihood of making a forecast at the boundary is rather low, unless the RF part constantly selects t as splitting variable.

Figures 12 and 28 show, for both random walk and generalized TVPs, their full-sample versions (up to the end of 2014, "ex post") and their version with a training sample ending

in 2007Q2 (the dashed blue line). There are two main observations. First, GTVPs are much less prompt to rewrite recent history than random-walk TVPs. Indeed, the green line and the magenta one closely follows each other all the way up to the end of the training sample. Second, while GTVPs can change many quarters after 2007Q2 (like the GDP constant), they are generally very close to each other at the boundary – especially when the time variation is statistically meaningful (like that of μ_t and $\gamma_{F_2,t}$), which is what matters for forecasting. This is much less true of random walk TVPs as there are clear examples where the two version differ for a long period of time (for instance, the intercept and the coefficient on F_2 in the GDP equation), and this often culminates at the boundary.[40]

Why and When MRF Can Fail to Deliver Better Forecasts

MRF can sometimes be outperformed by simpler alternatives, like standard RF that incorporate MAFs. When that occurs, it is usually due to the inadequacy of the linear part rather than GTVPs themselves. Unlike traditional TVPs, GTVPs rarely provides a model worse than OLS.

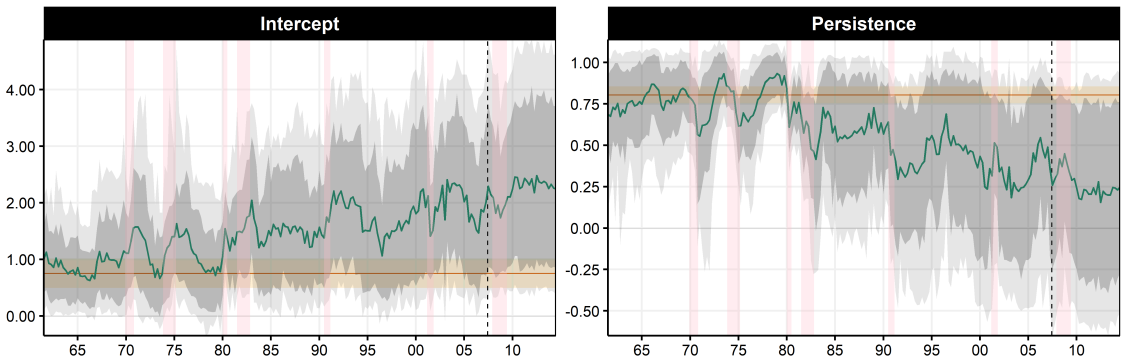
Trivially, β_t helps understanding relative performance. For instance, in the case of forecasting inflation with the *quarterly* data set, ARRF does not supplant RF-MAF. The critical difference between ARRF (reported in Figure 13a) and its restricted analog is that the two autoregressive coefficients of the former are shut to 0.[41] In Figure 13a, the estimates of ARRF broadly agree with the view that inflation persistence has substantially decreased during and following Volker disinflation (Cogley and Sargent, 2001; Cogley et al., 2010).

In terms of anticipated forecasting performance, such decline in persistence suggests a constrained version simply including μ_t may do better. The OOS evaluation period corresponds to the region of Figure 13a where $\phi_{1,t} + \phi_{2,t}$ is the nearest to 0. Given that observation, RF-MAF mildly improving upon ARRF is less surprising. An analogous finding emerges for GDP at many horizons. ARRF does not outperform RF-MAF like FA-ARRF and larger VARs versions of MRF do. GTVPs showcased in Figure 13b provide a simple explanation. There is only a limited role for persistence when allowing for a forest-driven μ_t. $\phi_{1,t} + \phi_{2,t}$ is below the OLS counterpart most of the time and the credible 68% credible region frequently includes 0. The ensuing forecast is essentially a time-varying constant, which is what RF-MAF does.[42] In sum, unlike many ML offerings, MRF successes and

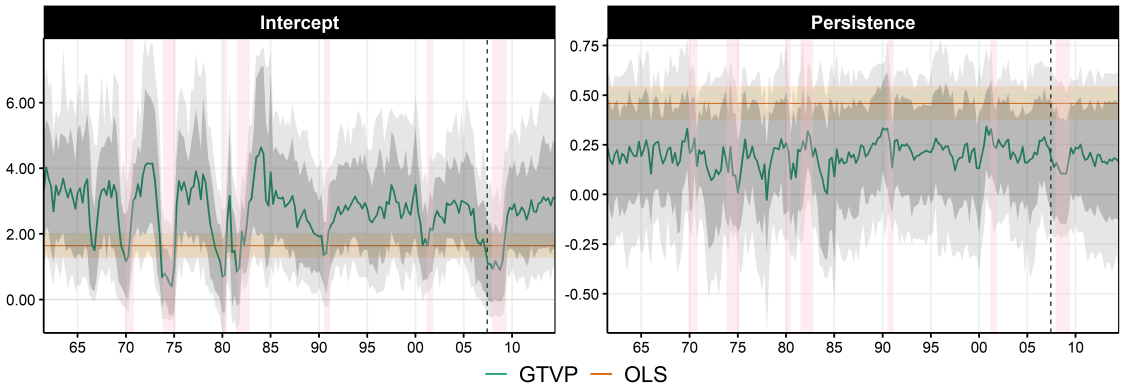
[40] In (real) practice, all models would be re-estimated each quarter. However, it is worth pointing out that re-estimating every period is much more important for random-walk TVP than it is for GTVPs. For such reasons, the TV-AR in section 2.4 was the sole model estimated every period rather than every two years.

[41] Of course, lags of INF can still enter the forest part for μ_t, so RF-MAF does not suppress entirely the link between current and recent inflation.

[42] This result is largely in accord with the reported sufficiency of a switching intercept (without additional autoregressive dynamics) to model US GDP in Camacho and Quiros (2007). However, Figure 13b suggests that there are rather 3 regimes: recession, expansion before 1985 (growth rate $\approx$ 3.5%), expansion after 1985



(a) Inflation



(b) GDP

Figure 13: GTVPs of the one-quarter ahead forecasts using ARRF. Persistence is defined as $\phi_{1,t} + \phi_{2,t}$. The gray bands are the 68% and 90% credible regions. The pale orange region is the OLS coefficient $\pm$ one standard error. The vertical dotted line is the end of the training sample. Pink shading corresponds to NBER recessions.

failures can be understood via a time-varying parameter interpretation. The helpfulness of this attribute cannot be overstated when thinking about future model improvements.

2.5.3. Cutting Down the Forest, One Tree at a Time

Evolving β_t can limit macroeconomists in their ability to use the model for counterfactuals. Complementarily, policy-makers will complain about the limited use for a model in which tomorrow's parameters are unknown (random walks). Fortunately, GTVPs may be the result of an opaque ensemble of trees, but they are made out of observables rather than a multiplicity of latent states. That is, they change, but according to a *fixed* structure. Hence, the reduced-form coefficients could easily change, and yet remain completely predetermined as long as $\mathcal{F}$ itself is stable. In this paradigm, a changing β_t is not necessarily empirical evidence supporting Lucas (1976)'s critique – rather, a changing $\mathcal{F}$ could be.

(growth rate slightly below 3%). The sufficiency of the switching intercept has also been documented in Markov-switching dynamic factor models for Norway (Aastveit et al., 2016) and Germany (Carstensen et al., 2020).

Hence, dissecting $\mathcal{F}$ is inherently interesting. One way to get started on this is to use well-established measures of Variable Importance (VI), originally proposed in Breiman (2001). Those extract features driving the *prediction*. Conveniently, they can be adapted to inquire β_t. Then, one can capitalize on VI's insights to build interpretable small trees parsimoniously approximating $\beta_{t,k}$'s path.

The construction of upcoming graphs consists in two steps. I start by computing 3 different VI measures: VI_{OOB} (out-of-bag predictive performance), VI_{OOS} (out-of-sample predictive performance) or VI_{β} (for a specific coefficient rather than the whole prediction). Appendix 2.7.3 contains a detailed explanation those and a discussion on how the current approach relates to recent work in the ML interpretability literature. As a potential data set for the construction of a surrogate tree, I consider the union of the 20 most potent predictors as highlighted by any of the three VIs. The tree is pruned with a cost-complexity factor (usually referred to as `cp`) of 0.075. That tuning parameter is set such as to balance its capacity to mimic the original GTVP and potential for interpretation.

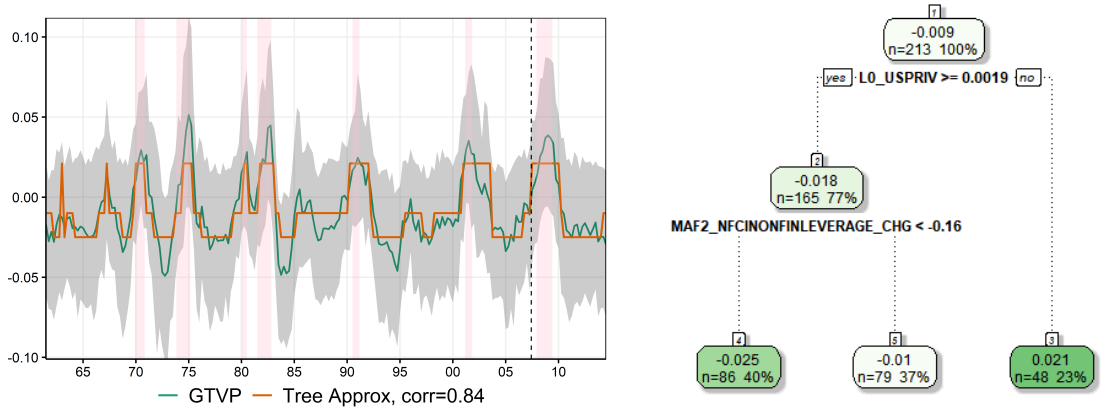
Unemployment Equation

I limit the attention μ_t and $\gamma_{2,t}$ paths, which were argued of greater importance to FA-ARRF's success in forecasting UR. Also, the nature of their variation is easier to characterize with a single tree (ex-post). Figures 14b and 14d show that paths can sometimes be summarized succinctly using a handful of predictors.

Most of μ_t can be captured by two states which are determined by a cut-off on total private sector employees (USPRIV): 0.021 (increasing unemployment) and -0.018 (decreasing). This first layer basically classifies recession vs expansions in a very parsimonious way, which is inevitably crude and imperfect. The additional split on a MAF of non-financial leverage provides a more refined classification: there are more of less three states. The time series plot shows the alternation between two symmetrically opposed states of 0.021 and -0.025 (respectively entering and exiting a recession) and a transitory (and seldomly visited) middle ground around 0.

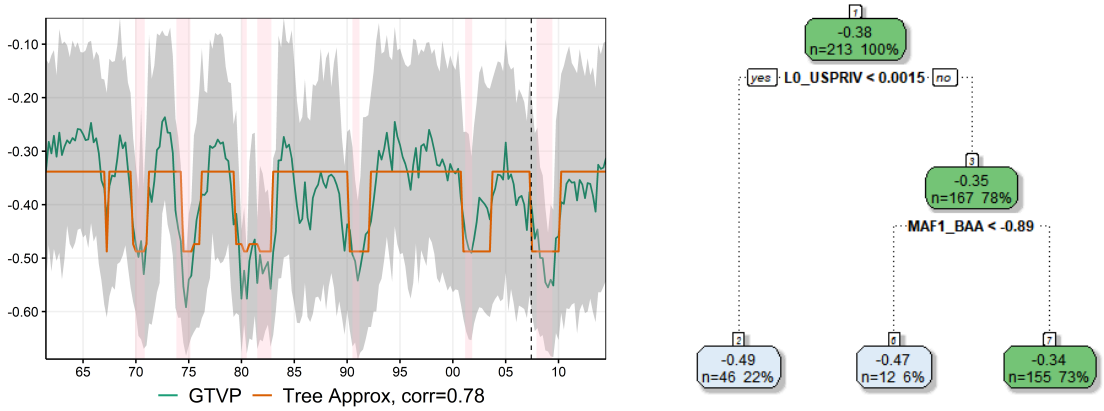
The impact of F_2 on UR switches significantly, and most of the action can be summarized by a private sector employees dummy (USPRIV). The indicator's movement downwards – which usually commence from the *onset* of a recession – can double the effect of F_2 on UR in absolute terms. However, some high (absolute) $\gamma_{2,t}$ episodes would be left behind when merely using USPRIV. Those are retrieved by an additional split with a MAF of average corporate bonds yield with a BAA rating (lower medium grade).

The GTVP (green line) often plunges earlier than the ex-post surrogate tree's replica (or-



(a) $\mu_t^{UR,h=1}$: Surrogate Model Replication

(b) $\mu_t^{UR,h=1}$: Corresponding Tree



(c) $\gamma_{t,F_2}^{UR,h=1}$: Surrogate Model Replication

(d) $\gamma_{t,F_2}^{UR,h=1}$: Corresponding Tree

Figure 14: Surrogate $\beta_{t,k}$ Trees. Shade is 68% credible region. Pink shading is NBER recessions.

ange). This is important, especially from a forecasting perspective. In Figure 29b, it is clear that leading indicators (especially financial ones) play a prominent role in driving the GTVP $\gamma_{2,t}$ – well before USPRIV starts showing signs of an imminent downturn. Since F_2 is already composed mostly of forward-looking variables, this hints at a convex effect of market-based expectations proxies.

Lastly, a word of caution. Given the points raised earlier in section 2.2.1, it is more appropriate to see these surrogate trees as suggestive of one potential explanation. It is an open secret that their exact structure is sensitive to small changes in the estimated path. For instance, little variation in β_t is needed to observe a change in the exact choice of variables itself. As a result, some of them may rightfully seem exotic when singled out in such a simple tree. GTVPs, as the product of a forest, will more often than not rely on a multitude of indicators from a specific group (which we observe in Figure 29a) rather than a single indicator.

Monthly Inflation Equation

As detailed in Appendix 2.7.6, FA-ARRF is a very competitive model for *monthly* inflation at all horizons. By its use of F_1, the real activity factor, it has the familiar flavor of a Phillips' curve (PC).[43] This is of interest given PCs have at best a very uneven forecasting track record (Atkeson et al., 2001; Stock and Watson, 2008; Faust and Wright, 2013). For instance, simple autoregressive/random walk/historical mean benchmarks often do much better.

Given its paramount importance within New Keynesian models, many explanations have been proposed for PC forecasts failures. The curve could be time-varying in a way that annihilates its forecasting potential (Stock and Watson, 2008). Closely related, some have stipulated the PC is nonlinear (Dolado et al., 2005; Doser et al., 2017; Lindé and Trabandt, 2019; Mineyama, 2020). If that were to be true, this should be exploitable. Lastly, an adjacent point of view, which became increasing popular following the Great Recession, is that the PC has irreversibly flattened to the point of predictive desuetude (Blanchard et al., 2015; Blanchard, 2016; Del Negro et al., 2020). Unlike the first two propositions, this one is, by nature, terminal.

Of course, all those explanations amount to hypotheses on the nature of $\gamma_{1,t}$'s time variation, of which MRF provides a very flexible account. It is worth emphasizing that MRF is estimated up to 2007Q2, unlike many of the above models explaining the "missing disinflation" after observing that it took place.[44] The variable importance measures reported in Figure 30 showcase a "consensus" subset of variables that matters for inflation time variation. Three popular ones are the trend, MAF of building permits and MAF of housing starts. The leading role for the trend suggests that exogenous time variation is important to explain inflation – to no one's surprise (Cogley and Sargent, 2001). Studying β_t-specific VI's suggest that this is mostly a feature of the intercept and persistence.

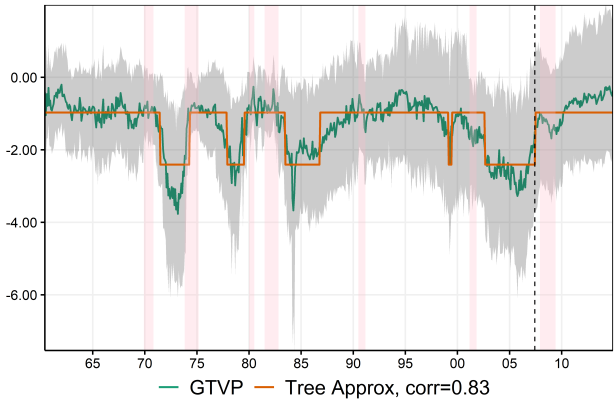
Figures 15, 31a and 31b allow to re-conciliate PC forecasting evidence. For instance, a visible PC death zone spans all of the 90s, which constitutes most of the sample used in Atkeson et al. (2001).[45] It also includes the post-2008 period, which motivated Blanchard et al. (2015)'s inquiry. Most interestingly, for the latter era, $\gamma_{1,t}$ is predicted to head toward 0 *out-of-sample*. To clarify, the parameter is driven by post-2008 data, but the structure itself ($\mathcal{F}$) is not re-evaluated past the dotted line.

By looking at predictive performance results *ex-post*, Stock and Watson (2008) report that

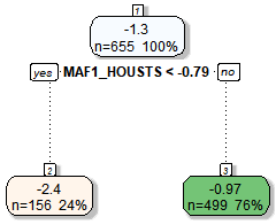
[43] As noted in Stock and Watson (2008), the plethora of output gap indicators used in literature makes the use of a common statistical factor a credible alternative.

[44] Indeed, they do so either by fitting the post-2008 data directly, or by choosing a specification (or building a theoretical model) directly inspired by the experience of the Great Recession.

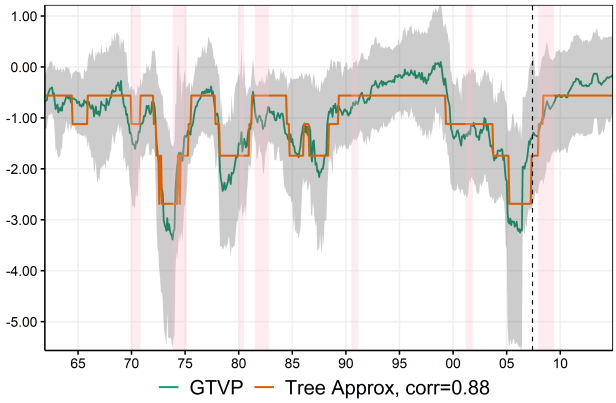
[45] The decade-long wedge between the OLS estimate and GTVP in Figure 31b nicely explains PC failures.



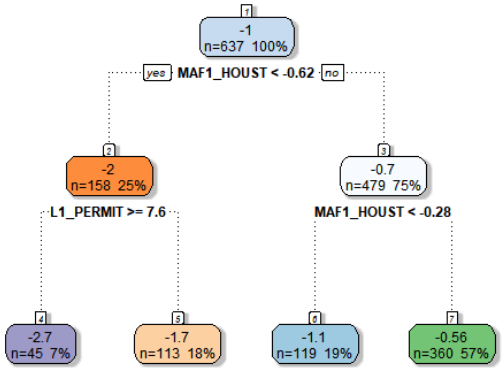
(a) $\gamma_{1,t}^{INF,h=1}$: Surrogate Model Replication



(b) $\gamma_{1,t}^{INF,h=1}$: Corresponding Tree



(c) $\gamma_{1,t}^{INF,h=12}$: Surrogate Model Replication



(d) $\gamma_{1,t}^{INF,h=12}$: Corresponding Tree

Figure 15: Surrogate $\beta_{t,k}$ Trees for Inflation. Shade is 68% credible region. Pink shading is NBER recessions.

Phillips' curve forecasts usually outperform univariate benchmarks around turning points, but suffer a reversal of fortune when the output/unemployment gap is close to 0. They note that the finding "cannot yet be used to improve forecasts" because their gap relies on a two-sided filter. More recently, Kotchoni et al. (2019) reinforce this view by showing an ARMA(1,1) is triumphant for inflation *except* in recessionary periods, where a data-rich environment can be helpful. But to capitalize on this, one needs a recession/expansion forecast. MRF recognize this potential and relies on leading indicators of the housing market to activate $\gamma_{1,t}$ in a timely manner. This is particularly evident from looking at $\gamma_{1,t}$'s VI measure in Figure 30 and its resulting GTVP in Figure 15. Overall, we see that the relationship between inflation and economic activity is episodic, as conjectured by Stock and Watson (2008), and often prevails before recessions (but not all). Figure 15 proposes a clear-cut answer: inflation responds to the real activity factor when the housing market is

booming.

For a long time, housing sector indicators have been known as predictors of future economic activity (Stock and Watson, 1998b; Leamer, 2007). However, when it comes to forecasting inflation itself, including leading indicators (like permits) does not remedy Phillips' curve forecasts failures (Stock and Watson, 2007). FA-ARRF differs by *not* using housing permits/starts as a replacement and/or additional output gap proxy. Rather, its role is to increase the curvature when the time is right. As mentioned above, one explanation is that housing starts and permits are proxying for future economic activity, resolving the conundrum posed by Stock and Watson (2008). Overall, this implies a PC which would be highly nonlinear in real activity, as further inquired in section 2.5.4. Another hypothesis is that MRF discovers – through aggregate data – how to leverage Stock and Watson (2019)'s insights that some components of inflation are much more cyclically sensitive than others. Stock and Watson (2019) show that the most cyclical component of inflation is *housing*, followed closely by food components. Accordingly, MRF activating $\gamma_{1,t}$ with building permits and housing starts is the algorithm's way of predicting when more cyclically sensitive components take the front stage – and by doing so, revive the Philipps' curve. In sum, nonlinearities would be a consequence of aggregation.

The predictive PC studied here differs in many aspects from those studied, for instance, in Blanchard et al. (2015). Importantly, F_1 summarizes mostly variables in first differences (or growth rates). A typical gap measure, being a deviation from a trend, will be much more persistent. Also, it remains negative for many years following a downturn. In contrast, F_1, which is strongly correlated with UR change, will go back up as soon as UR stops growing. To validate current insights and obtain new ones, I now study a prototypical Phillips' Curve.

2.5.4. The Phillips' Curve: Not Dead Yet?

The behavior of inflation since the Great Recession – starting with the missing disinflation and followed by "missing inflation" of recent years – sparked renewed interest in the Phillips curve. Much attention has been given to its hypothesized flattening (Blanchard et al., 2015; Galí and Gambetti, 2019; Del Negro et al., 2020). This body of work supports the view that the PC coefficient (either reduced-form or semi-structural) has substantially declined over the last decades. The focus on slow structural change is operationalized by the modeling strategy – either random walk TVPs or sample splitting at a specific date. Coibion and Gorodnichenko (2015) show less worry about PC's health. They rationalize post-2008 inflation with a simple OLS PC where expectations are based on consumer survey data rather than lags or professional forecasters. Del Negro et al. (2015) demonstrate that a standard DSGE (which encompasses a structural New Keynesian PC) is not

baffled by post-2008 inflation since it relies on model-based forward-looking expectations of future marginal cost. More recently, Lindé and Trabandt (2019) and Mineyama (2020) articulate theories supporting a nonlinear specification for the reduced-form PC, which could also account for the inflation puzzles punctuating the last 12 years. Given this background and forecasting results reported earlier, a traditional PC must be a fertile ground for MRF-based detective work.

I contribute to the literature by fitting an MRF which linear part corresponds to an expectations-augmented Phillips' curve. X_t is inspired by what Blanchard et al. (2015) (henceforth BCS) considers:

$$\pi_t = \theta_t \hat{\pi}_t^{LR} + (1 - \theta_t)\hat{\pi}_t^{SR} + \phi_t u_t^{GAP} + \psi_t \pi_t^{IMP} + \epsilon_t, \tag{2.4}$$

where π_t stands for CPI inflation, $\hat{\pi}_t^{LR}$ and $\hat{\pi}_t^{SR}$ respectively for long-run and short-run inflation expectations. u_t^{GAP} represents the (negative) unemployment gap and π_t^{IMP} is import prices inflation. I translate this to the MRF framework by making $\mu_t = \theta_t \hat{\pi}_t^{LR}$ the time-varying intercept, letting $\beta_{t,1} = 1 - \theta_t$ and by obtaining u_t^{GAP} by means of Hodrick-Prescott filtering.[46] As in BCS, $\hat{\pi}_t^{SR}$ is the average inflation over the last four quarters. Hence, the estimated equation

$$\pi_t = \mu_t + \beta_{1,t}\hat{\pi}_t^{SR} + \beta_{2,t} u_t^{GAP} + \beta_{3,t}\pi_t^{IMP} + \varepsilon_t \tag{2.5}$$

does not impose the constraint implied by θ_t in equation (2.4). However, estimation results will desirably have $\beta_{1,t} \in [0, 1]$ at almost any point in time. S_t is the same as that considered in the forecasting section. The data set runs up to 2019Q4.

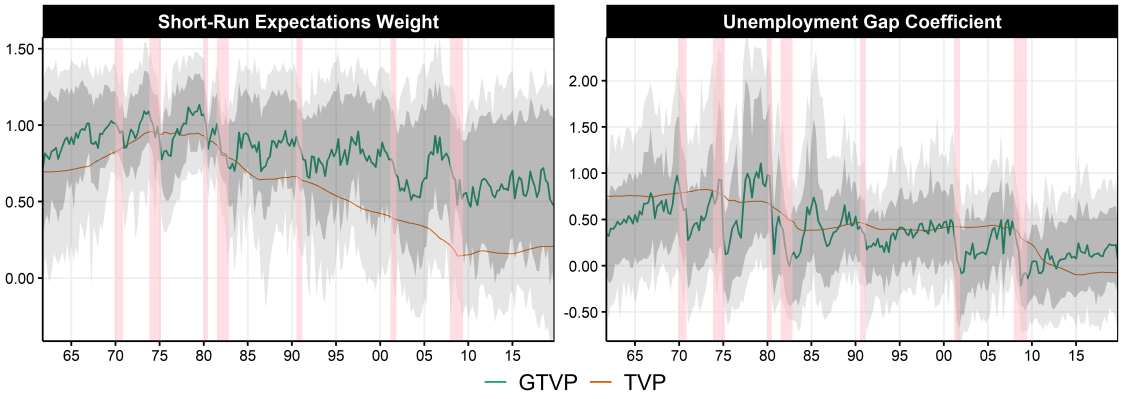


Figure 16: The gray bands are the 68% and 90% credible regions. Pink shading corresponds to NBER recessions.

[46] Specifically, both this gap and that of BCS get out of negative territory around 2014.

Figure 16 reports GTVPs of interest: the weight on short-run expectations and the output gap coefficient. Additionally, it contains traditional TVP estimates as means of comparison. The latter convey the usual wisdom: inflation expectations slowly start to be more anchored from the mid 1980s. Around the same time, the unemployment/inflation tradeoff begins its slow collapse. The updated data shows that the TVP-based Phillips' curve has further flattened to plain 0 in the last decade.

For $\beta_{1,t}$, the weight on short-run expectations, both methods agree that it has been decreasing steadily after the 1983 recession. But GTVPs highlight an additional pattern for the importance of $\hat{\pi}_t^{SR}$: it tends to increase during economic expansions, collapse during recessions then start increasing again until the next downturn. Note that the phenomenon is also observed in Figure 13b for the simpler ARRF on quarterly inflation. The decrease in the coefficient (usually of about 0.25) is observed for *every* recession and usually last for some additional quarters after the end of it. The linear rise in the coefficient occurs for all expansions except those preceding the early 90s and 2000s recessions, where the pattern is punctuated with additional peaks and troughs. The increased importance of short-run expectations with the age of the expansion is also observed for recent expansionary periods. Hence, the phenomenon is not merely a matter of the 70s and 80s recessions being preceded by a sharp acceleration of inflation.

From a more statistical point a view, the sharp decline in $\beta_{1,t}$ following every recession suggests that in the aftermath of an important downward shock, the long-run inflation expectation is a more reliable predictor as it is minimally affected by recent events. As the expansion slowly progress (and recessionary data points get out of the short-run average), $\hat{\pi}_t^{SR}$ becomes a more up to date and reliable barometer of future inflation conditions. This narrative is corroborated by variable importance (Figure 33) for $\beta_{1,t}$, which highlights the importance of the trend, but also recent lags of inflation.

When it comes to the low-frequency movement of the unemployment gap coefficient, both methods agree about a significant decline starting from the 80s. However, GTVPs uncover additional heterogeneity. **First** and most strikingly, $\beta_{2,t}$ gets very close to 0 following every recession. This suggests a nonlinear Philipps' curve where inflation responds strongly to a very positive u_t^{GAP} but not so much to a negative one. **Second**, the 70s and early 80s are characterized as a series of peaks (preceding the first three recessions of the sample) rather than a sustained high coefficient. Traditional TVPs, by excessive time-smoothing, dissimulate the effects of inflationary spirals on $\beta_{2,t}$. Such pre-recession accelerations still occur during the Great Moderation but in a much milder way.

Third, VI measures (in Figure 33) confirm the importance of activity indicators (like Total

Capacity Utilization (TCU)) in driving $\beta_{2,t}$ itself. The correlation between $\beta_{2,t}$ and TCU is 0.81, and the correspondence between the two variables is striking in Figure 17. Many notable increases in $\beta_{2,t}$ are nicely matched (between the two 70s recessions and before 2008). Of course, this simple characterization remains imperfect since it misses some highs (like the end of the 70s) and predicts a higher $\beta_{2,t}$ in the years following the 2008-2009 recession. Generally, given the strong co-cyclicality between TCU and u_t^{GAP}, this is evidence of a *convex* PC.

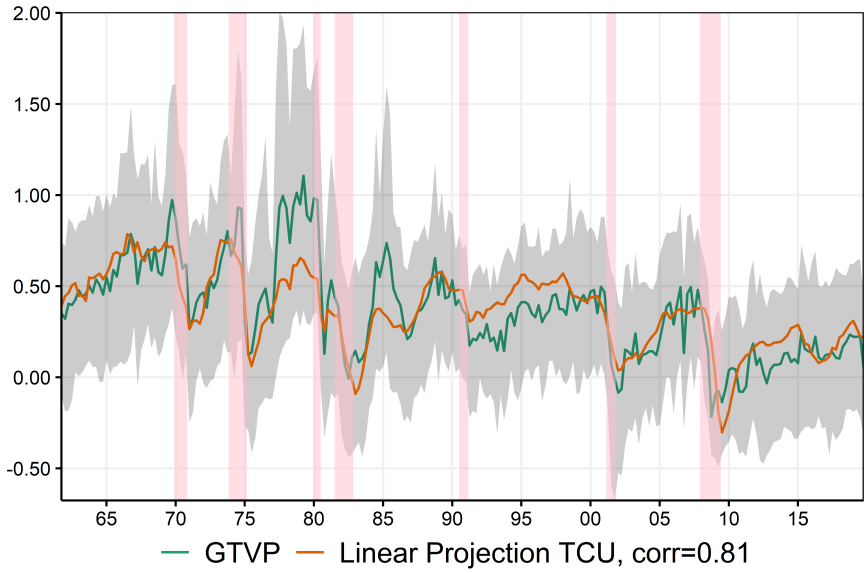


Figure 17: "What Goes Around Comes Around": Capacity Utilization is substantially correlated with the inflation-unemployment trade-off. The gray band is the 68% credible region. Pink shading corresponds to NBER recessions.

The collapse of $\beta_{2,t}$ following recessions is not unique to 2008: it happened following *every* recession since 1960. As a result, inflation will rise when the economy is running well above its potential, but much more timidly will it go down from economic slack. Recently, Lindé and Trabandt (2019) have shown that such a phenomenon can be rationalized by a New Keynesian DSGE model. Indeed, by allowing for additional strategic complementarity in firms price- and wage-setting behavior and solving the nonlinear model (rather than considering the linear approximation around the steady state), the authors obtain a state-dependent PC which becomes very flat during large downturns. This can explain both the small coefficient during recessions and its subsequent timid increase. Theoretically, convexity can also emerge from downward wage rigidities (Mineyama, 2020), but its plausibility for the post-2008 era has been contested (Coibion and Gorodnichenko, 2015).

This pattern remains when adding controls in the linear part for supply shocks and monetary policy shocks. Those are the usual confounding factors suspected of blurring the

relationship by introducing a positive correlation between unemployment and inflation.[47] The economic suspicion particular to this application is that omitting them could create a downward bias in $\beta_{2,t}$ that only occurs locally, generating the cyclical pattern. As it turns out, controls make cyclicality even more obvious in Figure 34, especially for the later part of the sample.[48] However, the overall strength of the coefficient is smaller (especially for the 70s).

Many hypotheses can be accommodated by a model estimated on two disjoint samples, like in Del Negro et al. (2020). Much fewer of them are compatible with the richer $\beta_{2,t}$ path extracted by MRF. This is important: learning the type of nonlinearity, rather than partially imposing it, helps in discriminating economic suppositions. Figure 17 and recent theoretical developments both suggest that much of the PC's decline is attributable to upward nonlinearities being less solicited in the last 3 decades. This is in accord with the policy hypothesis: since Paul Volker's chairmanship the monetary authority has responded much more aggressively to inflationary pressures, limiting the spirals that gave rise to high $\beta_{2,t}$'s in the 70s. Two conclusions emerge from this observation. First, exogenous change cannot so simply be ruled out. Second, knowing what were MRF beliefs about PC nonlinearities at different points in time could be enlightening.

Conditional Coefficient Forecasting

$\beta_{2,t}$'s lows are getting lower, and longer. Should we have known? Much of the recent work on PC is directly inspired by Great Recession aftermath, and aims at explaining it. Whether it is theoretical or empirical work, much of it could be overfitting: a model can replicate one or two facts it is trained to replicate, but fails to generalize. That is, even if models are tested out-of-sample (which is itself not so often the case in the literature), the choice of nonlinearity itself is often determined in attempt to match the OOS. Beyond the linear part being a PC, MRF does not assume much — and its nonlinearities are certainly not "personalized" to the recent inflation experience. Thus, it is interesting to ask: what was MRF "thinking" about $\beta_{2,t}$ in 2007? in 1995? Did it know something we did not, or did it learn (as most economists) of PC's collapse from the post-2008 experience? I conduct a $\beta_{2,t}$ dynamic learning exercise to find out.

To make this operational, MRF is estimated up to 1995, 2007 and 2019, and GTVPs are projected out-of-sample from those dates (when applicable). To be clear, $\hat{\beta}_{2,t|1995} = \hat{\mathcal{F}}_{1995}(S_t)$ means the coefficient *predictive structure* is last estimated in 1995. Coefficients keep moving

[47] While the time-varying constant can go a long way at controlling for such factors – being a RF in itself, including them in the linear part makes them "stand out" as everything going through the intercept is inevitably heavily regularized.

[48] Results being similar for both curves is reminiscent of Galí and Gambetti (2019) who report little differences between paths of reduced-form and semi-structural wage PCs (although they focus on long-run change).

out-of-sample because S_t does. $\hat{\mathcal{F}}_{1995}(S_t)$ and $\hat{\mathcal{F}}_{2007}(S_t)$ will differ for two main reasons. The first is estimation error – both in terms of precision and re-evaluating which nonlinearity seems more appropriate.[49] The second is structural change, perhaps completely exogenous or triggered by policy interventions.

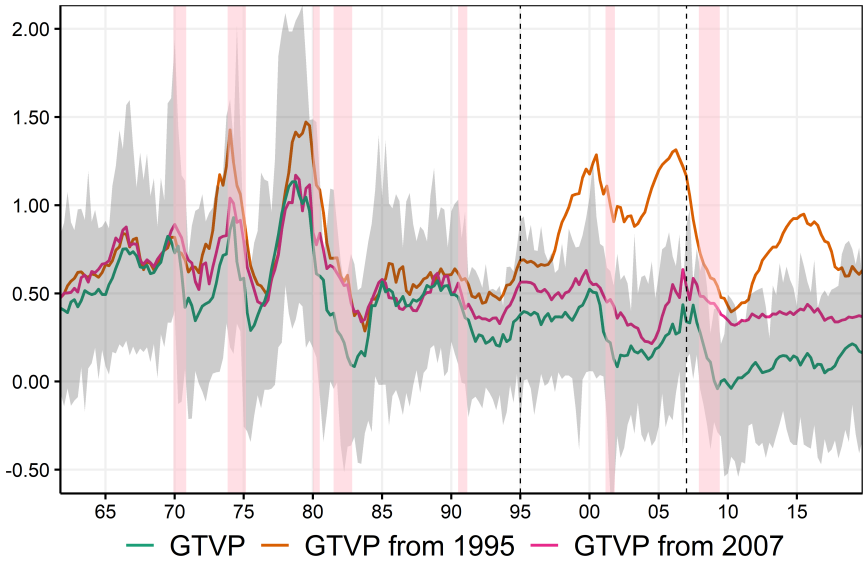


Figure 18: Conditional $\beta_{2,t}$ Forecasting. The gray band is the 68% credible region for GTVPs estimated up to 2019Q4. Pink shading corresponds to NBER recessions. For enhanced visibility, GTVPs are smoothed with 1-year moving average. The vertical dotted lines are the end of the training samples.

Much can be learned from Figure 18. First, GTVPs are all very alike for the pre-1995 period, suggesting little was observed post-1995 that made MRF change its reading of the past. Similarly, the green and the magenta line, which both share the 1995-2007 period within their training sets, are close to one another. Overall, this indicates that OOS difference between paths are very unlikely due to a better re-estimation and/or a completely new choice of $\mathcal{F}$.

Second, unlike what we have seen for the unemployment equation (Figure 11), there are important disparities between the ex-ante and the ex-post paths *out-of-sample*. Thus, one can rightfully hypothesize that structural change got in the way, making $\hat{\mathcal{F}}_{1995}$'s attempt of replicating the strong nonlinearities of the 70s into the 2000s go wildly off course. An analogous (yet far less noticeable gap) punctuates the post-2007 period. This suggests that while $\beta_{2,t}$ was expected to fall marginally following the crisis and stay low thereafter (according to $\hat{\mathcal{F}}_{2007}$), it was not expected to go *that* low. Indeed, only $\hat{\mathcal{F}}_{2019}$ hits 0 and stays in its vicinity.

Of course, by design, exogenous structural change cannot be captured out-of-sample –

[49] The second part has the flavor of model selection "error".

with the results that we know ($\hat{\mathcal{F}}_{1995}$). This dismal predicament does not apply to cyclical behavior: it has been forecastable at least since 1995. Indeed, $\hat{\mathcal{F}}_{1995}$ propose a $\beta_{2,t}$ for 2000 and 2008 that is very similar to that of 70s inflation spirals. Moreover, $\hat{\beta}_{2,t|1995}$'s collapse following 2008 is of a magnitude only seen during Arthur Burns' days. Hence, a much weaker PC following large downturns is hardly new. However, what $\hat{\beta}_{2,t|2007}$ and $\hat{\beta}_{2,t|2019}$ tell us is that the overall amplitude (and level) of those variations has evolved exogenously, forcing MRF to update $\mathcal{F}$ repeatedly.

This exercise may rightfully seem exotic, with no obvious analog in the literature. The simple explanation is that traditional time variations only give "trivial" parameter forecasts by construction, and there is no clear "learning" process to analyze. For example, the "forecasted" random walk TVP would be a straight line over the whole OOS. Doing so with a threshold model would only inform us of the increasing precision of estimation as sample size grows – i.e., the model itself cannot be re-evaluated. Unlike traditional nonlinear methods, MRF provides non-trivial β_t paths out-of-sample — and discovers exogenous structural change instead of imposing it.

2.6. Conclusion

I proposed a new time series model that **(i)** expands multiple nonlinear time series models, **(ii)** adapts Random Forest for macro forecasting and **(iii)** can be interpreted as Generalized Time-Varying Parameters. On the empirical front, the methodology provides substantial empirical gains over RF and competing non-linear time series models. The resulting Generalized TVPs have a very distinct behavior vis-à-vis standard random walk parameters. For instance, they adapt nicely to regime-switching behavior that seems pervasive for unemployment – while not neglecting potential long-run change. This finding is facilitated by the fact that GTVPs lend themselves much more easily to interpretation than either standard RF or random-walk TVPs. Indeed, rather than trying to open the black box of an opaque conditional mean function (like one would with plain RF), MRFs can be compartmentalized in different components of the small macro model. Furthermore, GTVPs can be visualized with standard time series plots and credible intervals are provided by a variant of the Bayesian Bootstrap.

When looking at Phillips' curves in general, MRF finds both structural change in the persistence and regime-dependent behavior in the economic activity/inflation trade-off. In particular, a recurrent theme across all specifications is that the slowly decaying curve is also much steeper when the economy is overheating – in line with the convexity/nonlinearity hypothesis. Hence, MRF can be of great help sorting out what is plausible and what is not when it comes to macroeconomic equations with a history of controversy. Since there is no shortage of those, MRF holds many possibilities for future research.

2.7. Appendix

2.7.1. More on Engineering S_t

To appreciate the point that various factors *and* the raw data can both be included together, let us put RF aside for a moment, and look at a high-dimensional linear regression problem. Suppose we define $S_t = [X_t \;\; F_t]$ and by construction the factors are some linear combination of original features ($F_t = X_t R$).[50] We can estimate

$$y_{t+1} = X_t\beta + X_t R\gamma + u_t \tag{2.6}$$

using LASSO. Of course, this would not run with OLS because of perfect collinearity, which is the standard motivation for not mixing dense and sparse approaches. By Frisch-Waugh-Lowell theorem and the factor model

$$X_t = \Lambda F_t + e_t,$$

(2.6) above is equivalent to

$$y_{t+1} = e_t\beta + F_t\gamma + u_t.$$

At first sight, this has more parameters than either the dense or sparse approach. However, with some adequate penalization of β and γ, the model can balance a proper mix of dense and sparse. For instance, activating some β's "corrects" the overall prediction when the factor model representation is too restrictive for the effect of a specific regressor X_k on y_{t+1}.[51] This representation has been studied in Hahn et al. (2013) and Hansen and Liao (2019) to enhance hard-thresholding methods' performance (like LASSO) in the presence of highly correlated regressors. Coming back to RF, this means its strong regularization/selection allows for both the original data and its rotation to be included in S_t. This also suggests it is relatively costless to explore alternative rotations of X_t.

2.7.2. Block Bayesian Bootstrap Details

BBB is a conceptual workaround to reconcile time series data with multinomial sampling. For completeness, I briefly review the *standard* Bayesian Bootstrap. Let all the available data be cast in the matrix $Z_t = [y_t \;\; X_t \;\; S_t]$. Z is considered as a discrete *iid* random variable with T support points. Define $N_t = \sum_{\tau=1}^{T} I(Z_\tau = z_t)$, which is the number of occurrences of z_t in the sample. The goal is to conduct inference on the data weight vector $\theta_{1:T}$, and then obtain credible regions for the posterior functional $\beta_t = \mathcal{T}(\theta_{1:T})$. To do so, we need to

[50] Note that in this section only, X_t denotes generic raw regressors rather than MRF's linear part. This switch allows for the use of familiar-looking notation.

[51] That problem has been documented in Bai and Ng (2008) and others.

characterize the posterior distribution of vector θ (stripped of its subscript for readability)

$$\pi(\theta|\mathbf{z}) = \frac{f(\mathbf{z}|\theta)\pi(\theta)}{\int f(\mathbf{z}|\theta)\pi(\theta)d\theta}.$$

Conditional on θ, the likelihood of the data is multinomial. The prior is Dirichlet. Since Dirichlet is the conjugate prior of the multinomial distribution, the posterior is also Dirichlet. That is, it can be shown that combining the likelihood

$$f(\mathbf{z}|\theta) = \frac{N!}{N_1! \cdots N_T!} \prod_{t=1}^{T} \theta_t^{N_t} \quad \text{with prior distribution} \quad \pi(\theta) = \frac{1}{B(\alpha_{1:T})} \prod_{t=1}^{T} \theta_t^{N_t+\alpha_t-1}$$

gives rise to the posterior distribution

$$\pi(\theta|\mathbf{z}) = \frac{1}{B(\bar{\alpha}_{1:T})} \prod_{t=1}^{T} \theta_t^{N_t+\alpha_t-1} \quad .$$

where $\bar{\alpha}_t = \alpha_t + N_t$ and $B(\bar{\alpha}_{1:T}) = \frac{\prod_{t=1}^{T} \Gamma(\bar{\alpha}_t)}{\Gamma\left(\sum_{t=1}^{T} \bar{\alpha}_t\right)}$. Using the uninformative (and improper) prior $\alpha_t = 0 \ \forall t$, we can simulate draws from the (proper) posterior using $\theta_t \sim \text{Exp}(1)$. The object of scientific interest is typically not θ *per se* but rather a functional of it. In Taddy et al. (2015), the functional of interest is a tree and inference is obtained by computing $\mathcal{T}(\theta_{1:T})$ for each $\theta_{1:T}$ draw. BBB considers a different Z_t so that it is plausibly *iid* when used with stationary time series data. The derivations above can be carried by replacing t by $\mathfrak{b}$ and T by $\mathfrak{B}$. Practically, this implies drawing $\theta_{\mathfrak{b}} \sim \text{Exp}(1)$ which means observations within the same block $(\underline{\mathfrak{b}} : \bar{\mathfrak{b}})$ share the same weight. As an alternative to this BBB that would also be valid under dependent data, Cirillo and Muliere (2013) provide a more sophisticated urn-based approach with theoretical guarantees. It turns out their approach contains the well-known non-overlapping block bootstrap as a special case, which the above is only its Bayesian rendition.

2.7.3. More on Surrogate β_t Trees

The approach described in section 2.5.3 belongs to a family of methods usually referred to as "surrogate models" (Molnar, 2019). Attempting to fit the whole conditional mean obtained from a black-box algorithm using a more transparent model is a global surrogate. An obvious critique of this approach is that if the complicated model justifies its cost in interpretability with its predicting gains, it is hard to believe a simple model can reliably recreate its predictions. Conversely, if the surrogate model is quite successful, this casts some doubts about the relevance of the black box itself. In this line of work, a more promising avenue is a local surrogates model as proposed in Ribeiro et al. (2016), which fits interpretable models *locally*. By following Granger (2008)'s insights, we already have

this: by looking at the β_t paths directly, we effectively have a local model – in time. The purpose of surrogate models is to learn about the model, not the data. The former is much easier in MRF than in standard RF since the vector β_t fully characterizes the prediction at a particular point in time.[52] Moreover, the coefficients are attained to predictors that can have themselves a specific economic meaning. Considering this and the earlier discussion of section 2.2.1, it is natural in a macro time series context to fit surrogate models to time-varying parameters themselves – a blatant divide-and-conquer strategy.

About VI_{OOB}, VI_{OOS} and VI_{β}

I now explain the motivation and mechanics behind the different VI measurements. The first measure, VI_{OOB}, is the standard out-of-bag (hence OOB) VI permutation measure widely used in RF applications (Wei et al., 2015). It consists of randomly permuting one feature S_j and comparing predictive accuracy to the full model on observations that were not used to fit the tree.[53] This pseudo evaluation set is convenient because it is a direct byproduct of the construction of the forest. Under a well-specified model that includes enough lags of y_t, autocorrelation of residuals will not be an issue. This condition is likely to be met here since the analysis focuses on results for $h = 1$. [54] VI_{OOS} considers a different testing set more natural for time series data: the real OOS, which in this section spans from 2007q2 to the end of 2014. By construction, this measure focuses on finding variables which contribution paid off during a specific forecasting experiment, rather than throughout the whole sample. This is not bad *per se* but is a different concept that can be of independent interest. Finally, both VI_{OOB} and VI_{OOS} focus on overall fit. VI_{β} implements the same idea as VI_{OOB} but is calculated using a different loss function. That is, $VI_{\beta_{k,j}}$ reports a measure of how much the path of β_k is altered (out-of-bag) when variable S_j is randomly permuted in the forest part. Finally, I use the various VI measurements as devices to narrow down the set of predictors for the construction of intuitive trees.

I restrict the number of considered variables (for the next step) to be 20 for each VI criteria. When VI suggest that a parsimonious set of variables matter, it is very rarely more than 3 or 4 variables. Thus, restricting it to 20 is a constraint that only binds if all variables contribute, but marginally, in the spirit of a Ridge regression (Friedman et al., 2001). When it comes to that, the cut-off is simply the natural reflection of a trade-off between

[52] More generally, any partially linear model in the spirit of MRF has a potential for local surrogate analysis along the linear regression space rather than the observations line.

[53] This is thought as the equivalent for a black-box model to setting a specific coefficient to 0 in a linear regression and then comparing fits. However, VI as implemented here (and in most applications) does not re-estimate the model after dropping S_j. This differs from a t-test since it is well known that the latter is equivalent to comparing two R^2's – the original one and that of a re-estimated model, under the constraint.

[54] Notwithstanding, at longer horizons, VI_{OOB} could paint a distorted picture in the presence of autocorrelation – the same way K-fold cross validation can be inconsistent for time series data (Bergmeir et al., 2018). This worry can be alleviated by using a block approach like in section 2.2.7.

interpretability and fit.

2.7.4. On Tuning Parameters

The bulk of the discussion on the algorithm's specifics is deferred to the `R` package. None of the RFs reported in the text were tuned. This is not heresy, as minuscule performance gains from doing so (like optimizing `mtry`) are the norm rather than the exception. Additionally, restraining the terminal nodes size can only alter performance very mildly and it is now clear why (Goulet Coulombe, 2020c). Nonetheless, reviewing some of those untuned tuning parameters can be insightful about MRFs inner workings. "Algorithm" 4 below summarizes when and where those enter the MRF procedure.

- `RWR`: stands for Random Walk Regularization strength as discussed in 2.2.3. It is the ζ in equation (2.2).
- `RL`: stands for Ridge Lambda (λ) in equation (2.1). Prior means are OLS estimates.
- `Minimal Node Size`: Minimal parent leaf size to consider a new split. Set to 10 for quarterly data and 15 for monthly.
- `MLF`: stands for Minimum Leaf Fraction. It is the parameter in MRF that has a role complementary to that of minimum node size. The so-called "fraction" is the ratio of parameters in the linear part to that of observations in any node (which includes most importantly the terminal ones). Here is an example. Set `MLF` $= 2$, the linear part has 3 parameters, and we are trying to split a subset of 15 observations. This setting implies that any split that results in having less than 6 observations in the children note will not be considered. This specific setting ensures that the ratio of parameters to observations never exceeds $1/2$ in any node. This ensure stability, especially if the two aforementioned HPs are set to 0. However, when `RWR` and `RL` are active, it is possible to consider `MLF` $= 1$ or even lower. The extra regularization allows in the latter case to have base regressions that have parameters/observations ratio exceeding 1 (high-dimensional setting). This is desirable with quarterly data because setting `MLF` > 2 or higher seriously restricts the potential depth of the trees.
- `mtry`: how many S_j's do we consider as potential "splitter" at each split? It is easier to think about it as a fraction of the total number of predictors. For regression settings, the suggested value is $1/3$. The lower it gets, the more random tree generation gets, and better diversification may ensue. Moreover, `mtry` directly impacts computational burden. It is often found, in a macro context, that lowering `mtry` to 0.2 does not alter performance noticeably, while reducing appreciably computations. In fact, running RF-MAF with `mtry`$\in \{0.1, 0.2, 0.33, 0.5\}$ delivers nearly identical

performance for all variable/horizon pairs of the quarterly exercise. This is likely attributable to macro data having a factor structure. If S_j is "not available" for a split when it would in fact maximize fit locally, there is another strongly correlated $S_{j'}$ ready for the task. For instance, if the unemployment rate is discarded by `mtry`, then there are more than 20 other labor indicators that can possibly substitute for it. If those 20 variables are all a noisy representation of the same latent variable the model wants to split on, then the probability of having none to split with at a given point is $\left(1 - \frac{\texttt{mtry}}{\#\text{regressors}}\right)^{20} \approx 0$.

- `Trend Push`: Some minorities may end up being underrepresented as a result of `mtry`'s discriminating action. While there are 20+ labor indicators in the data base, there is only one trend. Since exogenous change should most certainly not be underrepresented, its "personalized" probability of inclusion can be pushed beyond what `mtry` suggests.

- `Subsampling Rate`: is set at 75%.

A scaled down quarterly forecasting exercise was conducted to see whether tuning any of those could help. Precisely, horizons 1, 2, and 4 quarters were considered and models (ARRF,FA-ARRF,VARRF) were estimated once at the beginning of the OOS period (2002). Tuning parameters were optimized targeting 1998-2002 data as an artificial test set. Possible values were `RWR`$\in \{0, 0.5, 0.95\}$, `RL`$\in \{0.1, 0.5\}$, `mtry`$\in \{0.2, 0.33, 0.5\}$ and `min.node.size`$\in \{10, 40\}$. It is found that results are largely invariant to pre-optimized HPs. As mentioned earlier, what matters most in the linear part. It is observed that optimizing tuning parameters can help reduce marginally RMSEs of MRFs that were sometimes struggling (like VARRF). Results are available upon request.

2.7.5. Additional Simulations Results

DGP 4: SETAR. In this second SETAR example

$$y_t = X_t\beta_t + \epsilon_t, \quad \epsilon_t \sim N(0, 0.5^2)$$
$$\beta_t = \begin{cases} [2 \;\; 0.8 \;\; -0.2], & \text{if } y_{t-1} \geq 1 \\ [0 \;\; 0.4 \;\; -0.2], & \text{otherwise,} \end{cases}$$

AR models are doing badly by not capturing the change in mean and dynamics. It is noteworthy that in this DGP, predictive power quickly vanishes after $h = 1$, which is why we observe little performance heterogeneity at longer horizons in Figure 19a: those are dominated by the unshrinkable prediction error. Specifically tailored for this class of DGPs, the two SETARs are offering the best performance. A less trivial observation is that

Algorithm 4 How the key tuning parameters enter MRF, and other practical aspects

1: Draw blocks of some size (8 for quarterly, 24 monthly), that makes for `Subsampling Rate`% of the sample. To simply get the mean prediction, 100 trees are usually more than enough. To get credible regions to stabilize, 200-300 trees are typically needed.

2:
- For each subsample: run (2.2) recursively on that sample given λ and ζ values until each (potential) parent nodes are smaller than `Minimal Node Size`.
- A total of `mtry` predictors are considered at each splitting step $\mathcal{J}^-$ is randomly picked out of $\mathcal{J}$. Those probabilities are all $1/\dim(\mathcal{J})$ by default. `Trend Push` pushes that of the trend further if judged appropriate for a given data set.
- When evaluating potential splits, discard those that would not meet `MLF`'s requirements on resulting children nodes.
- This outputs one tree structure $\mathcal{T}$.

3: When inputted with new observations of X_t and S_t, each tree produces a forecast. MRF forecast is the mean of the those.

4: Same goes for β_t: each tree predicts its own β_t out-of-sample and the posterior mean is the average of all those.

5: In-sample β_t's need an extra step: only draws that did not use observation t to construct the tree (that is, for which t was left out of the subsample) are used to characterize the distribution of β_t.

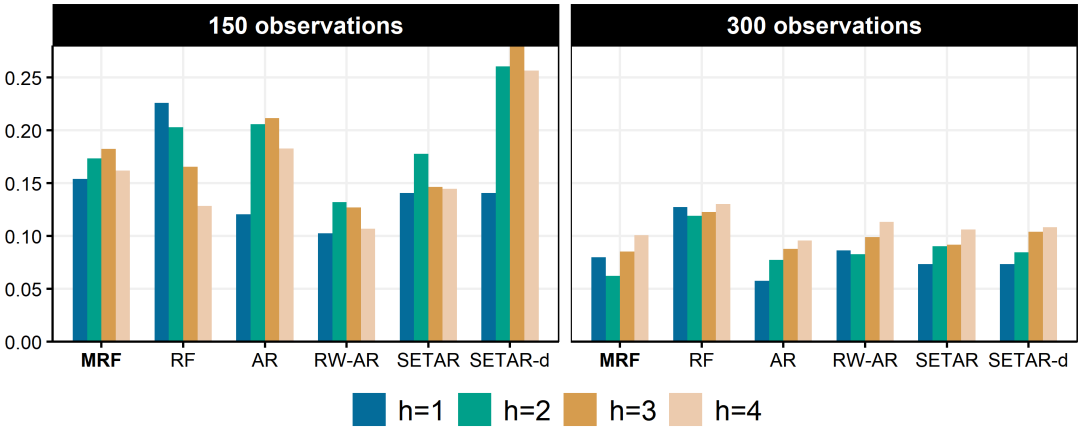
MRF and RF, while much more general, perform only marginally worse than SETARs. The tie between MRF and RF is attributable the importance of the switching constant in the current DGP, which both models allow for.

DGP 5: AR(2) WITH A BREAK. Results for

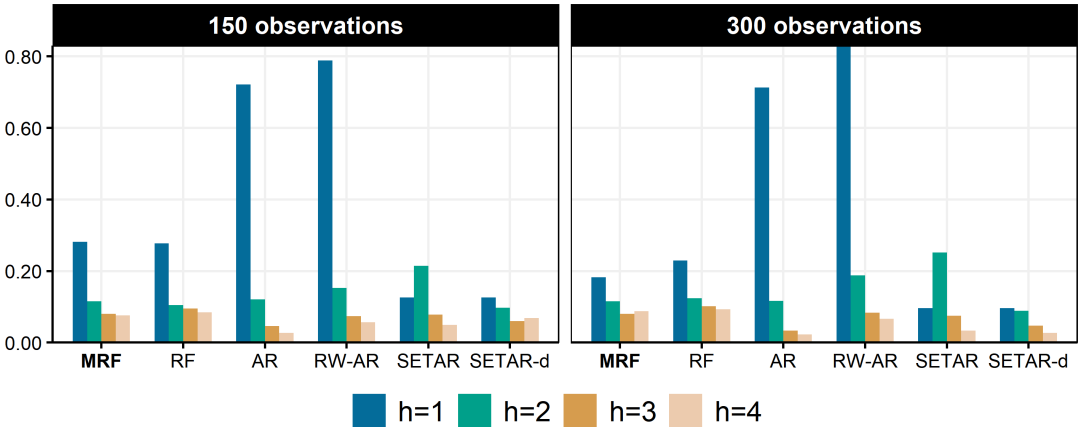
$$y_t = X_t\beta_t + \epsilon_t, \quad \epsilon_t \sim N(0, 0.3^2)$$
$$\beta_t = \begin{cases} [0 \;\; 0.7 \;\; -0.35], & \text{if } t < T/2 \\ [0.15 \;\; 0.6 \;\; 0], & \text{otherwise} \end{cases}$$

are reported in Figure 19b. In this setup, RW-AR is expected to have an edge, with the estimation window excluding pre-break data. At horizon 1, both RW-AR and ARRF are the best model, beating the robust AR by a thin margin. For $h > 1$, ARRF emerges as the best model at both 150 and 300 sample sizes. Naturally, RW-AR is always close behind.[55] As expected, the two models are better than the remaining alternatives by allowing for exogenous structural change (which SETARs and AR do not) and explicitly modeling the autoregressive part (which RF does not).

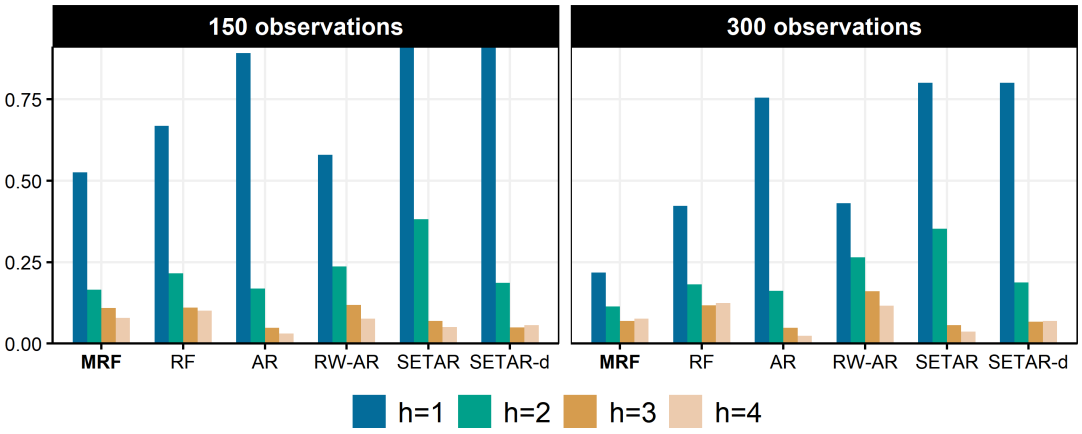
[55] Although not reported here, I considered a simple linear model where I search for a single break (in time) and use the data after the break for forecasting. This option does as well as ARRF for this particular DGP.



(a) DGP is SETAR.



(b) DGP is AR(2) with structural break.



(c) DGP is SETAR with structural break.

Figure 19: Displayed are increases in relative RMSE with respect to the oracle.

DGP 6: SETAR WITH A STRUCTURAL BREAK. This is slight complications of DGP 1. Again, SETARs are expected to fail because they are not designed to catch breaks. RW-AR is also expected to fail because it does not model switching. RF is general enough, but is anticipated to be inefficient. All these heuristics for

$$\text{DGP 4} = \begin{cases} \text{DGP 2}, & \text{if } t < T/2 \\ \text{DGP 3}, & \text{otherwise} \end{cases}$$

are verified in Figure 19c: MRF is the better model followed closely by RW-AR and RF for short horizons. With 300 observations, the lead of ARRF, as well as the second position of RF, are both strengthened. At longer horizons, all models perform poorly (including the oracle) due to the fundamental unpredictability of the law of motion for β_t. For these horizons, misspecification only plays a minor role in total forecast error variance, explaining the small and homogeneous decrease in performance with respect to the oracle.

2.7.6. Monthly Forecasting Results

I run a similar exercise as in Goulet Coulombe et al. (2019) which is very close to what has been precedently conducted for quarterly data. FRED-**M**D is now used. It contains 134 monthly US macroeconomic and financial indicators observed from 1960M01 (McCracken and Ng, 2016). To match the experimental design of Goulet Coulombe et al. (2019) for ML methods, Industrial Production (IP) replaces GDP and IR is dropped. The horizons of interest are $h = 1, 3, 9, 12, 24$ months. The forecast target is the average growth rate $\sum_{h'}^{h} y_{t+h'}^{v}/h$ which is much less noisy than the monthly growth rate. For example, for inflation 24 months ahead, I target the average inflation rate over the next two years – rather than the monthly inflation rate in 2 years. The OOS period is the same as before.

In Figure 23, VARRF is now doing much better on average, ranking first in terms of mean improvement over AR. ARRF still provides great insurance against doing worse than a plain AR counterpart (here AR(12)).[56] FA-ARRF remains very competitive. The models that do not have the MAFs (benchmarks) are clearly outperformed by the rest that do. This unsurprisingly indicates that lag polynomial compression can be of even greater use at the monthly frequency.

Table 19 reports specific $RMSE_{v,h,m}/RMSE_{v,h,AR}$'s with Diebold-Mariano tests. Broadly, they show that (i) MAFs are without any doubt the major improvement for the first three variables (IP, UR, SPREAD), (ii) simpler approaches like RF-MAF and AR+RF do well (*except* for INF) (iii) *all* MRFs do very well for inflation. Particularly, for (iii), ARRF and Tiny ARRF provide significant gains of 33% and 45% over the benchmark at $h = 12$ and $h = 24$,

[56] This is also true for the more parsimonious AR, see Table 19.

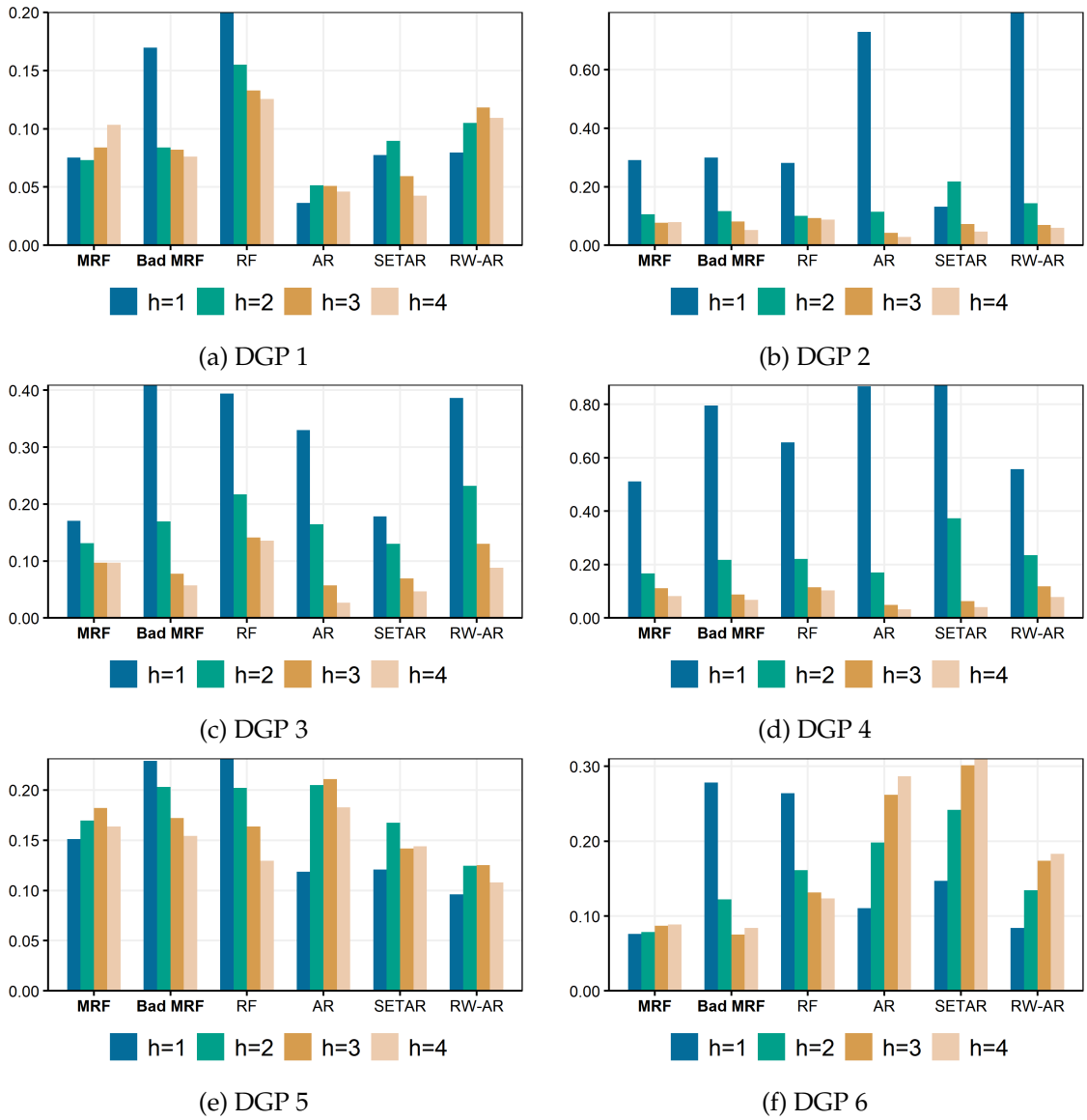
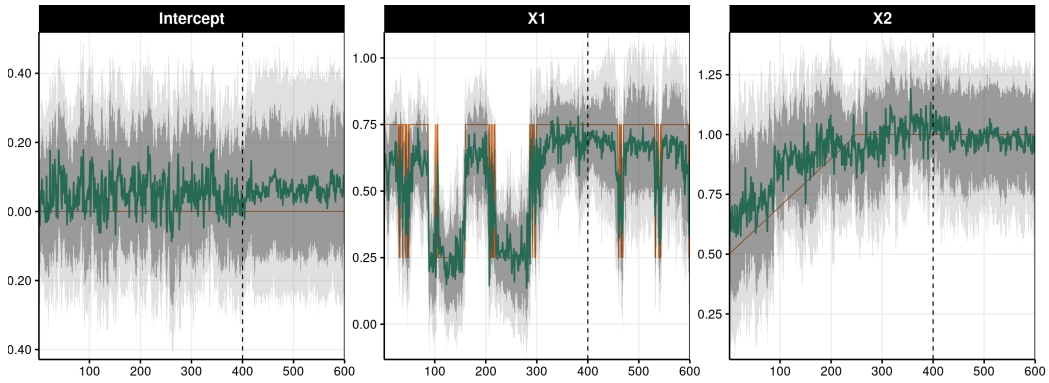
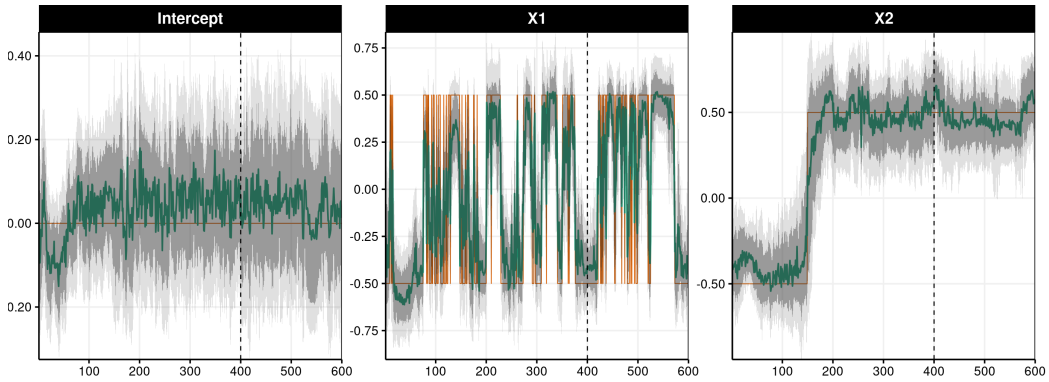


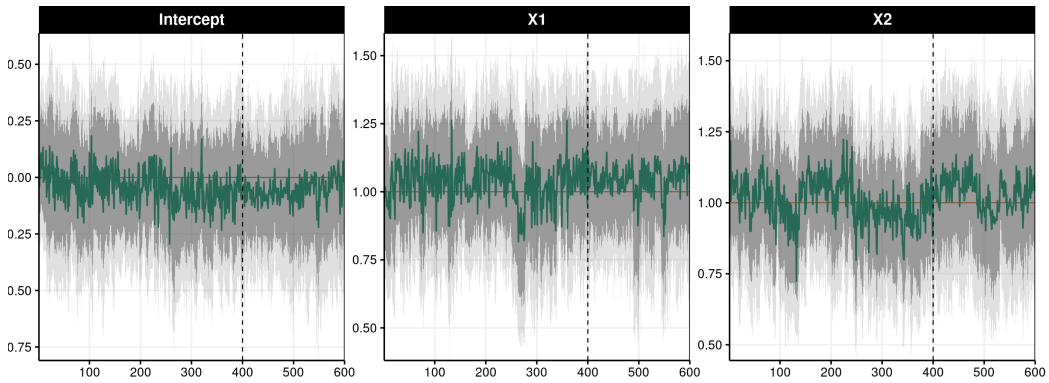
Figure 20: Investigation of the consequences of X_t's misspecification, as exemplified by "Bad ARRF". Instead of the first two lags of y_t, X_t is replaced by randomly generated *iid* (normal) variables. Total number of simulations is 50, and the total number of squared errors is thus 2000.



(a) DGP 4



(b) DGP 5



(c) DGP 6

Figure 21: The grey bands are the 68% and 90% credible region. After the blue line is the hold-out sample. Green line is the posterior mean and orange is the truth. The plots include only the first 400 observations.

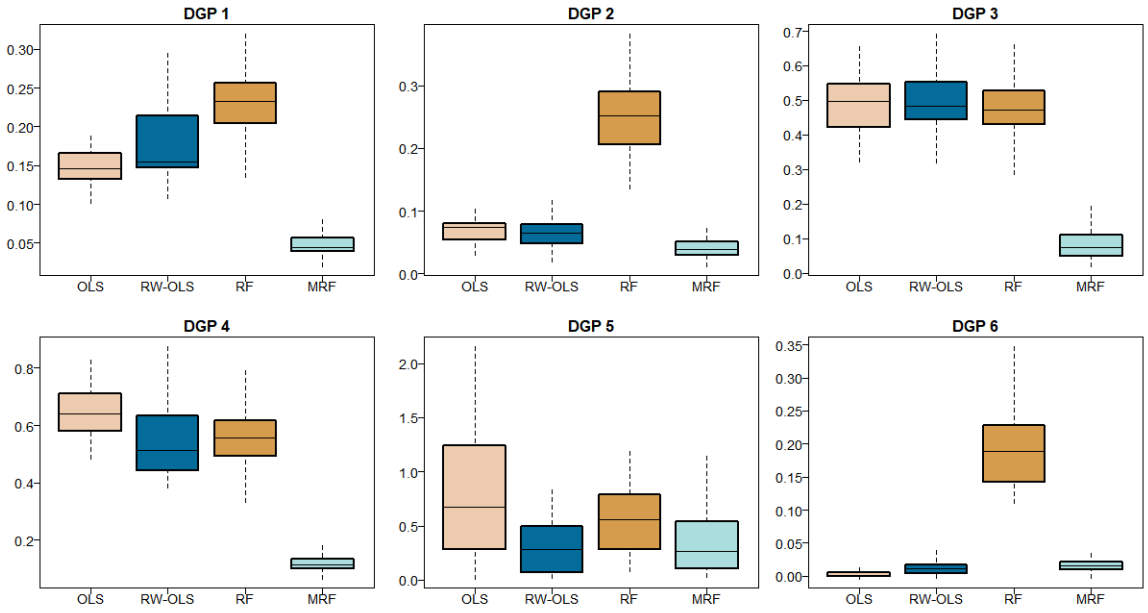


Figure 22: The distribution of RMSE dis-improvements with respect to the oracle's forecast for 4 models: OLS, Rolling-Window OLS, plain RF, MRF. 50 simulations of 750 OOS forecasts each.

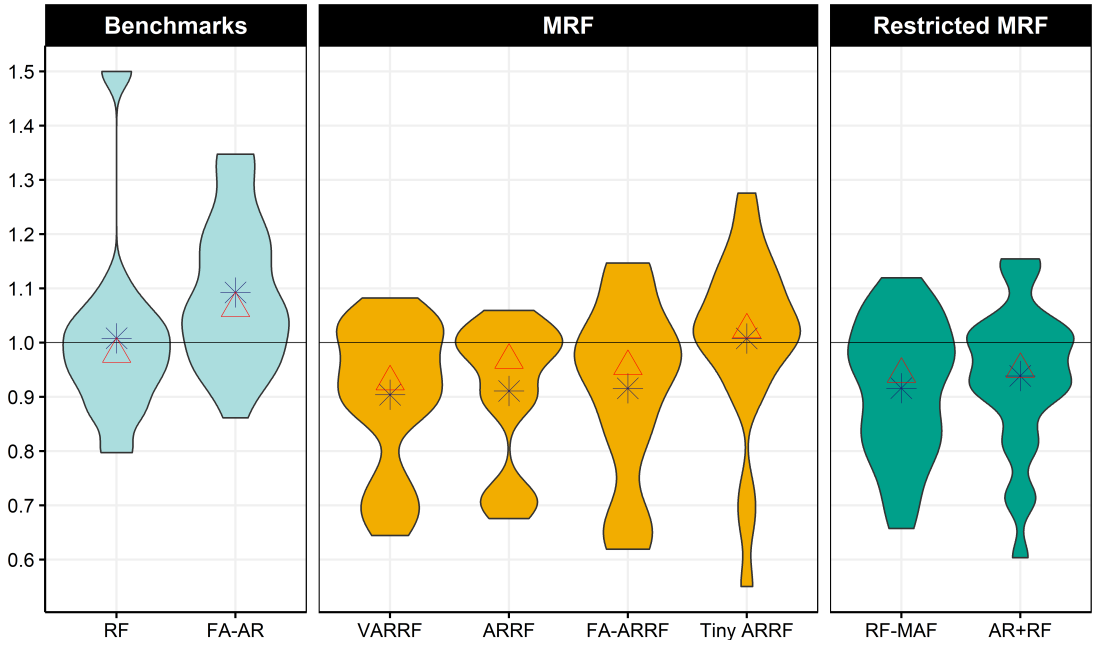


Figure 23: The distribution of $RMSE_{v,h,m}/RMSE_{v,h,AR}$ for monthly data. The star is the mean and the triangle is the median.

respectively. It is clear from this evidence, and that of the quarterly section, that forcing time-invariant inflation dynamics is costly in terms of RMSPE. GTVPs will confirm that, in accord with classic evidence on the matter (Cogley and Sargent, 2001).

Gains for INF are miles ahead from the usual competition. Table 19 includes forecasts inspired by the contribution of Atkeson et al. (2001): 1, h and 12 months moving averages are considered (where h is the targeted horizon). As in the original paper, the "AO-12" forecasts prove remarkably resilient, but are bested with sizable margins at each horizons by ARRF, Tiny ARRF, and FA-ARRF. For instance, at $h = 24$, the next best non-MRF forecast delivers 16% gains over the benchmark AR, whereas the worst MRF provides a gain of 27%. Tiny ARRF supremacy at longer horizons is sensible given that restricting S_t emphasizes long-run exogenous change, a usual suspect for INF.

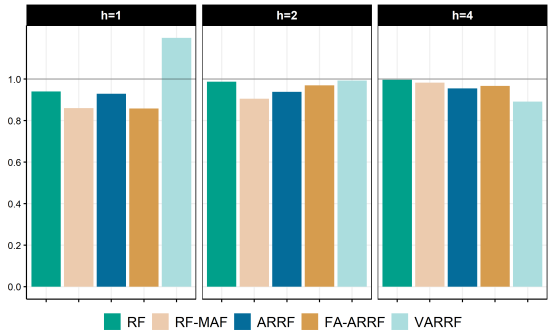
Another interesting observation emerges from MRFs successes with monthly inflation. FA-ARRF is often close to the best model, and that, at all horizons. Naturally, this is intriguing as FA-ARRF can be thought of as a Phillips' curve forecast, which recurrent failures are well documented (Atkeson et al., 2001; Stock and Watson, 2007). Moreover, it is reported that FA-AR, in contrast, does really bad. To sort this out, FA-ARRF's GTVPs are studied in section 2.5.3.

Non-US Data

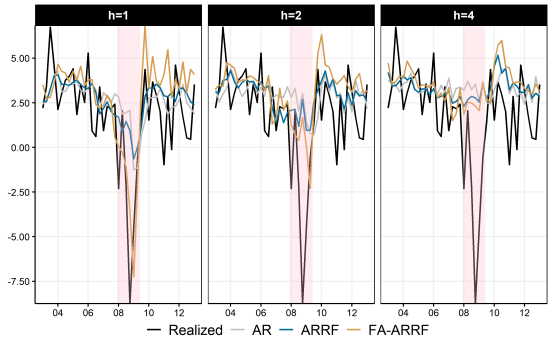
Much attention has been paid to the prediction of US economic aggregates. An even greater challenge is that of forecasting the future state of a small open economy. Such an application is beyond the scope of this chapter but is considered in Goulet Coulombe et al. (2020b). The study considers the prediction of more than a dozen key economic variables for Canada and Québec using the large Canadian data base of Fortin-Gagnon et al. (2018). Forecasts from about 50 models and different averages of them are compared, with ARRF and FA-ARRF among them. MRFs generate substantial improvements especially at the one-quarter horizon for numerous real activity variables (Canadian GDP, Québec GDP, industrial production, real investment). In such cases, ARRF or FA-ARRF provide reductions (with respect to autoregressive benchmark) that are sizable and statistically significant, going up to 32% in RMSE. That performance is sometimes miles ahead from the next best option (among Complete Subset Regression, Factor models, Neural Networks, Ridge, Lasso, plain RF and different model averagaging schemes). Goulet Coulombe et al. (2020b)'s results suggest that MRFs forecasting abilities generalize beyond the traditional exercise of predicting US aggregates.

More recently, Goulet Coulombe et al. (2021) uses MRF (along with a plethora of ML models) with a newly-built large UK macro data base, and finds that it can provide substantial gains during the Pandemic Recession. One of the reasons for that is the capacity of MRF to be nonlinear *and* extrapolate, which off-the-self tree-based methods (like RF) lack.

2.7.7. Additional Figures and Tables

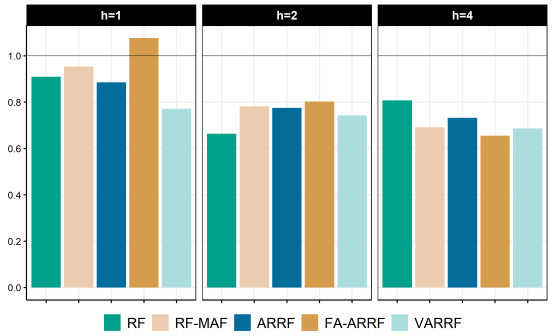


(a) $RMSE_{GDP,h,m}/RMSE_{GDP,h,AR}$

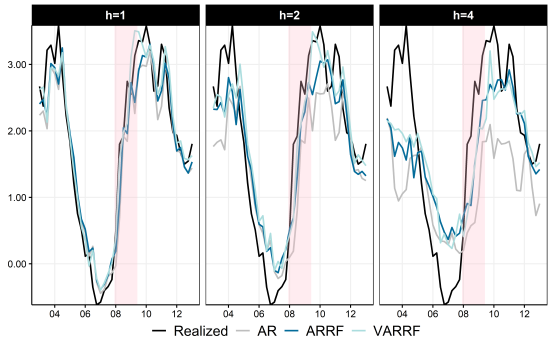


(b) A look at forecasts

Figure 24: GDP results in detail

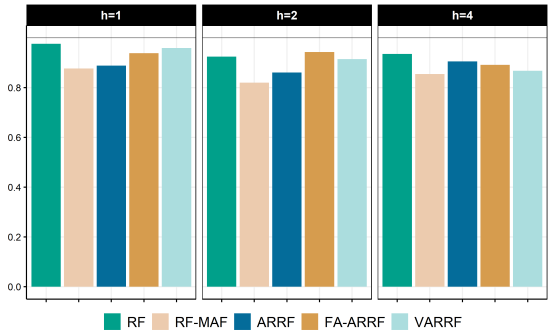


(a) $RMSE_{SPREAD,h,m}/RMSE_{SPREAD,h,AR}$

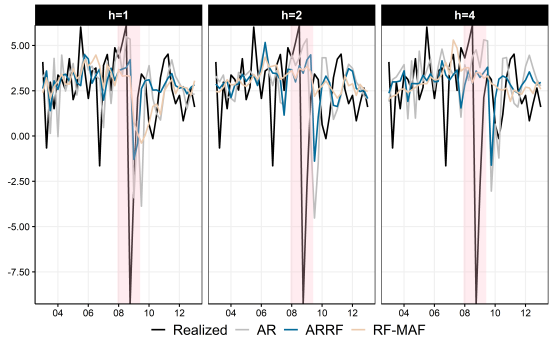


(b) A look at forecasts

Figure 25: SPREAD results in detail



(a) $RMSE_{INF,h,m}/RMSE_{INF,h,AR}$



(b) A look at forecasts

Figure 26: INF results in detail

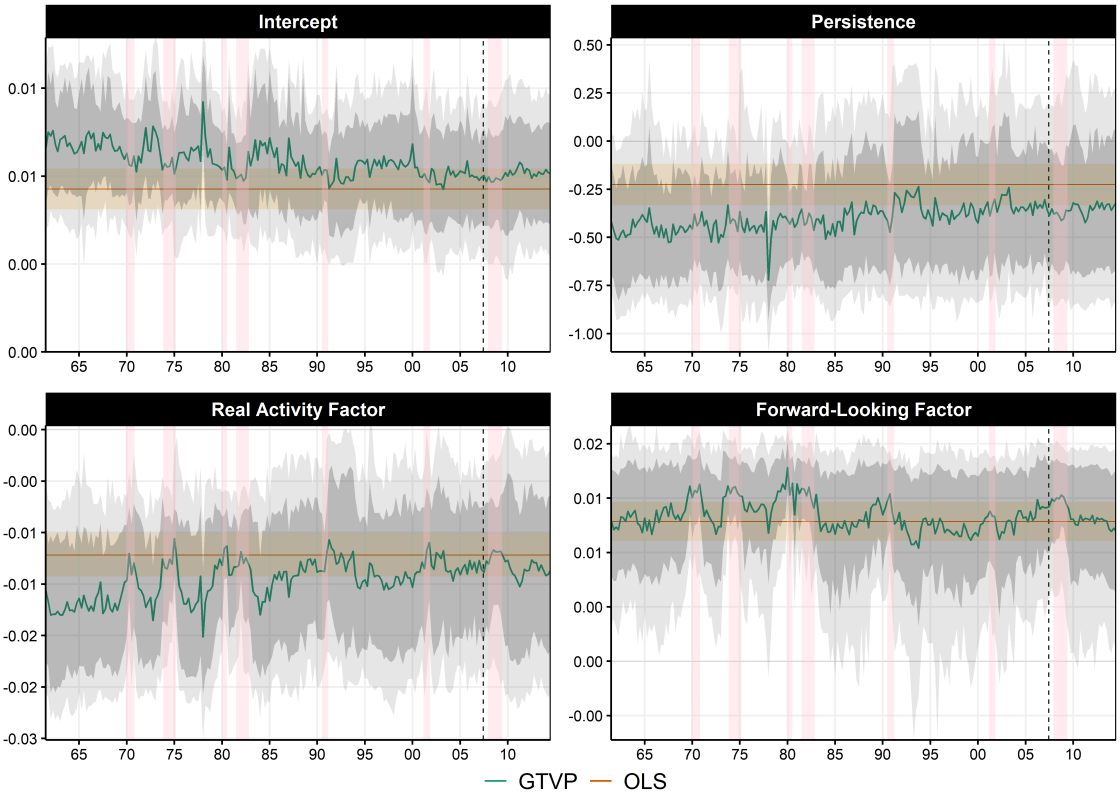
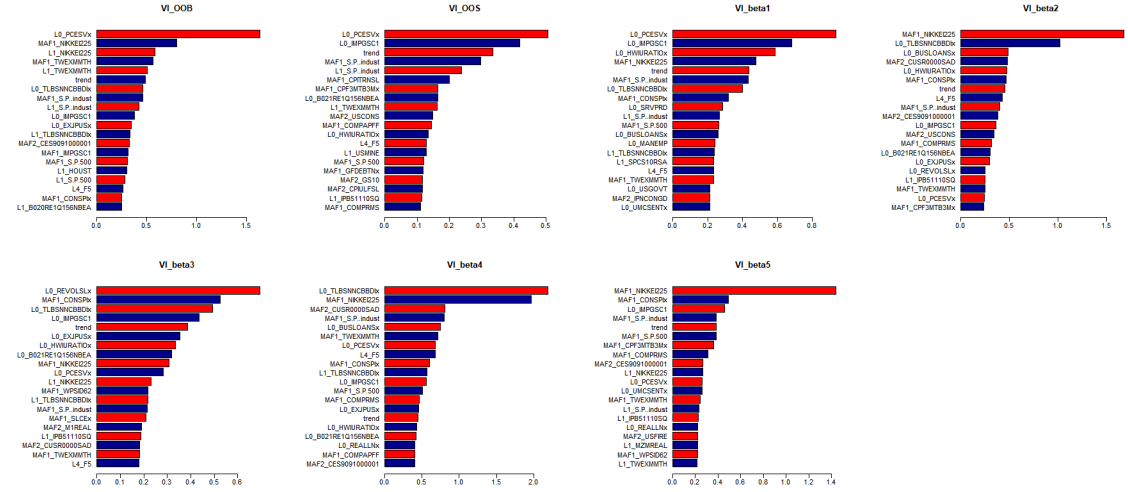
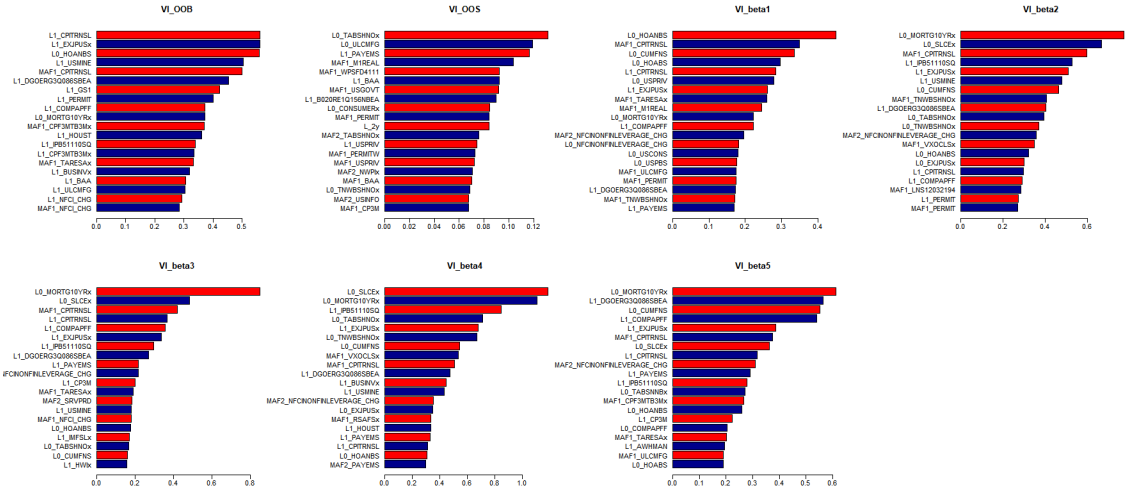


Figure 27: GTVPs of the one-quarter ahead GDP forecast. Persistence is defined as $\phi_{1,t} + \phi_{2,t}$. The grey bands are the 68% and 90% credible region. The pale orange region is the OLS coefficient $\pm$ one standard error. The vertical dotted blue line is the end of the training sample. Pink shading corresponds to NBER recessions.

Figure 28: GDP equation β_t's obtained with different techniques. Persistence is defined as $\phi_{1,t} + \phi_{2,t}$. TVPs estimated with a ridge regression as in Chapter 1 and the parameter volatility is tuned with k-fold cross-validation. Ex Post TVP means using the full sample for estimation and tuning as opposed to only using pre-2002 data as for GTVPs. The pale orange region is the OLS coefficient $\pm$ one standard error. Pink shading corresponds to NBER recessions.

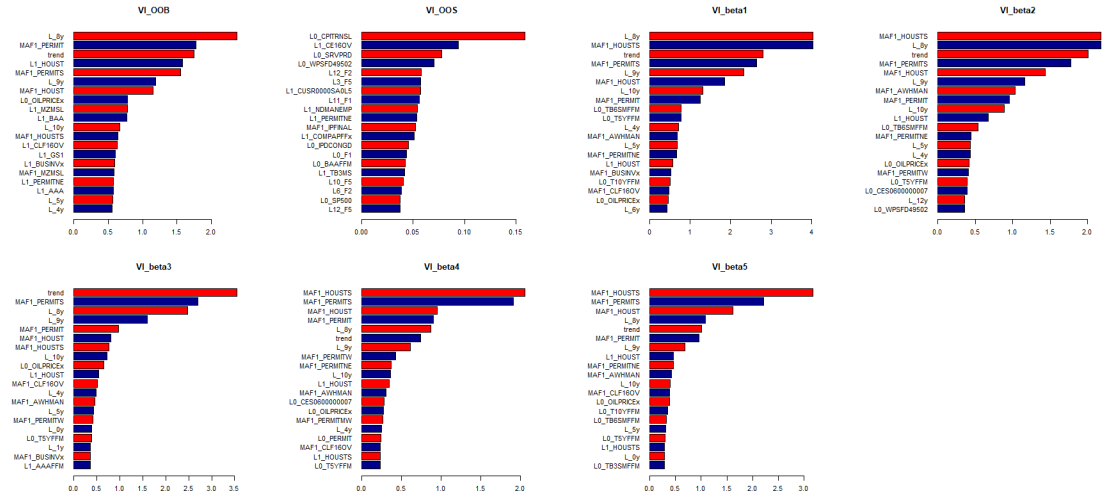


(a) GDP horizon 1

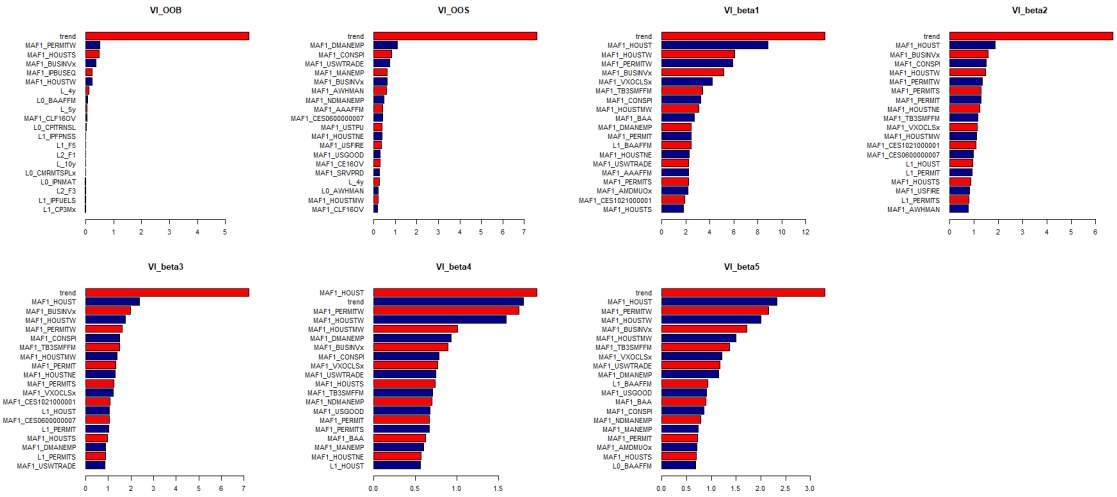


(b) UR horizon 1

Figure 29: 20 most important series according to the various variable importance (VI) criteria. Units are relative RMSE gains (in percentage) from including the specific predictor in the forest part. VI_{OOB} means VI for the out-of-bag criterion. VI_{OOS} is using the hold-out sample. VI_{β} is an out-of-bag measure of how much $\beta_{t,k}$ varies by withdrawing a certain predictor.

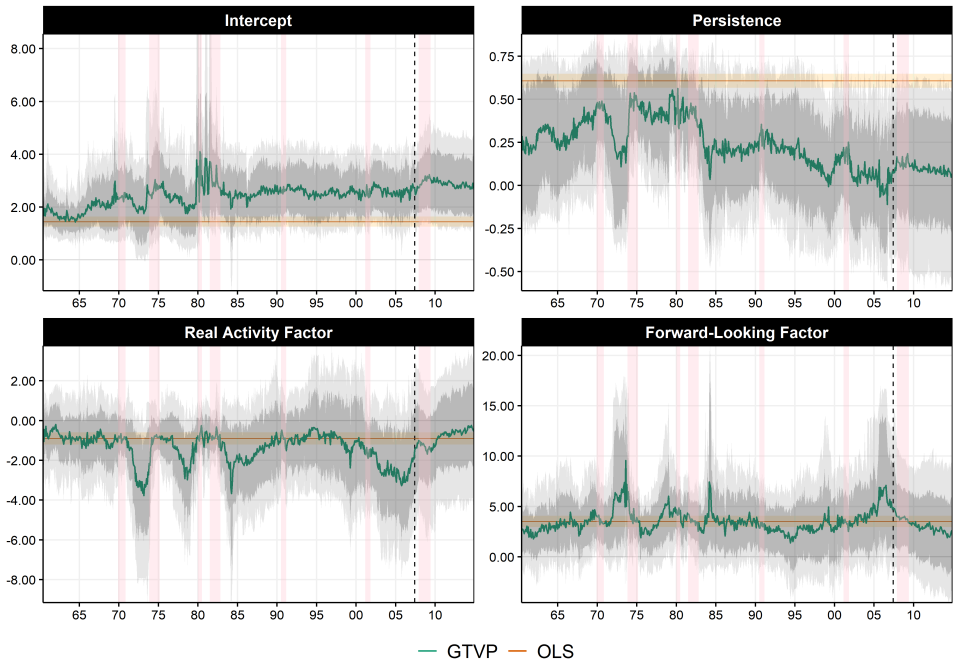


(a) One month ahead inflation forecast

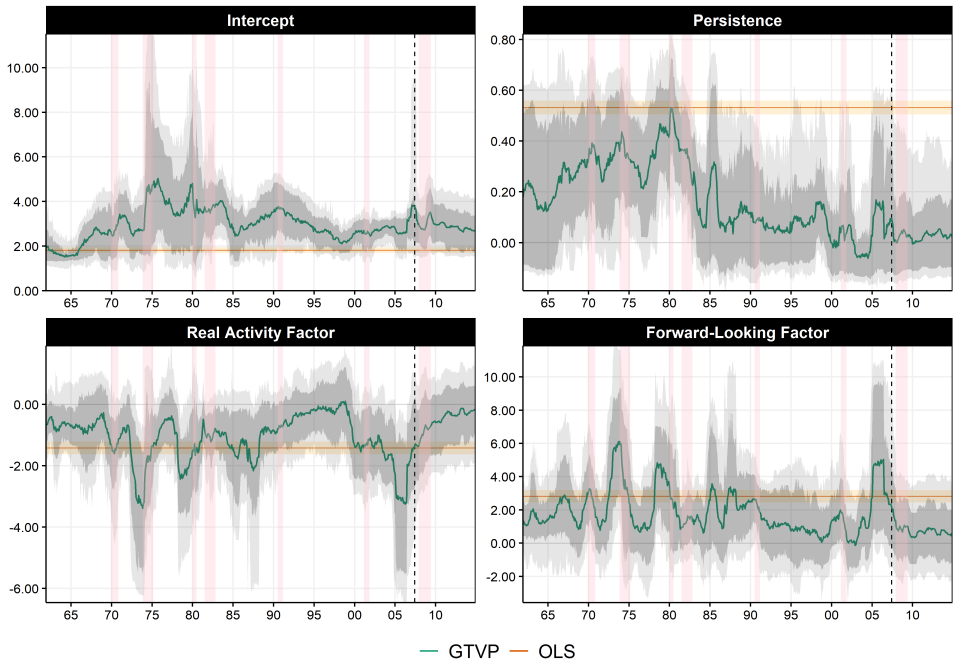


(b) Average inflation over the next 12 months

Figure 30: 20 most important series according to the various variable importance (VI) criteria. Units are relative RMSE gains (in percentage) from including the specific predictor in the forest part. VI_{OOB} means VI for the out-of-bag criterion. VI_{OOS} is using the hold-out sample. VI_{β} is an out-of-bag measure of how much $\beta_{t,k}$ varies by withdrawing a certain predictor.

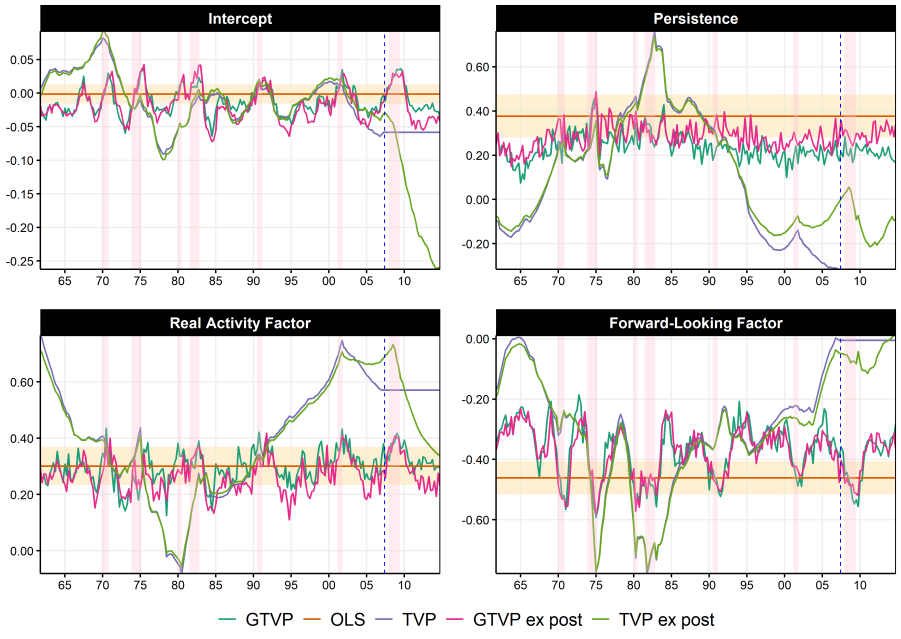


(a) One-month ahead

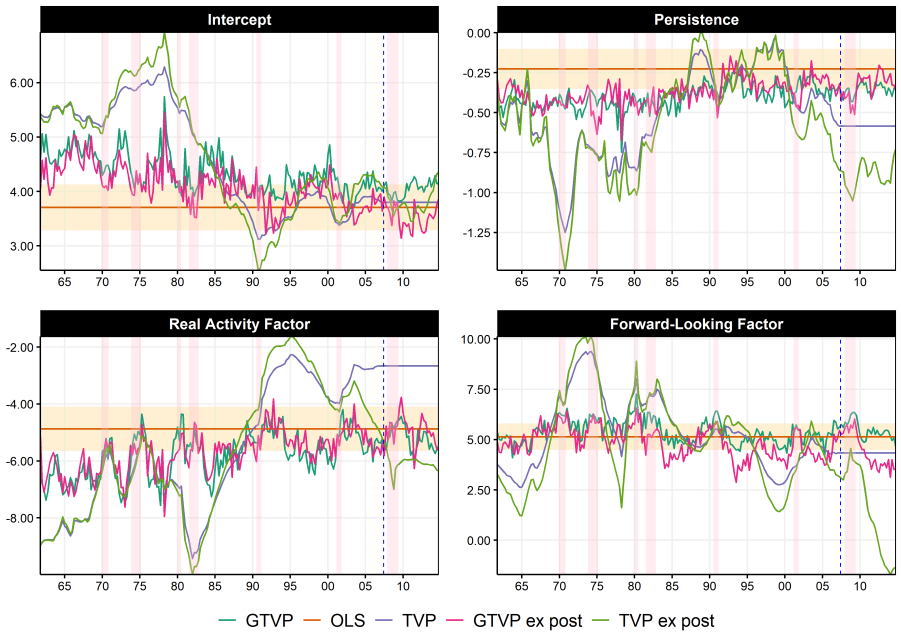


(b) 12-months ahead

Figure 31: GTVPs of monthly inflation forecast. The grey bands are the 68% and 90% credible regions. The pale orange region is the OLS coefficient $\pm$ one standard error. The vertical dotted line is the end of the training sample. Pink shading corresponds to NBER recessions.



(a) UR equation



(b) GDP equation

Figure 32: β_t's obtained with different techniques. TVPs estimated with a ridge regression as in Chapter 1 and the parameter volatility λ is tuned with k-fold cross-validation, **then divided by 100**. This means the standard deviation of parameters shocks is allowed to be about 10 times higher than what CV recommends. Ex Post TVP means using the full sample for estimation and tuning as opposed to only using pre-2002 data as for GTVPs. The pale orange region is the OLS coefficient $\pm$ one standard error.

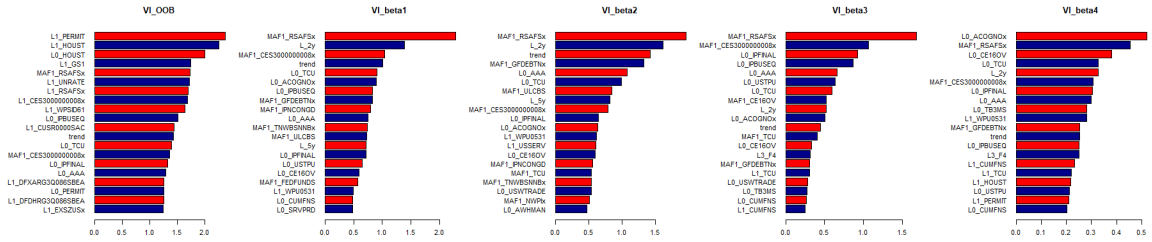


Figure 33: 20 most important series according to the various variable importance (VI) criteria. Units are relative RMSE gains (in percentage) from including the specific predictor in the forest part. VI_{OOB} means VI for the out-of-bag criterion. VI_{OOS} is using the hold-out sample. VI_{β} is an out-of-bag measure of how much $\beta_{t,k}$ varies by withdrawing a certain predictor.

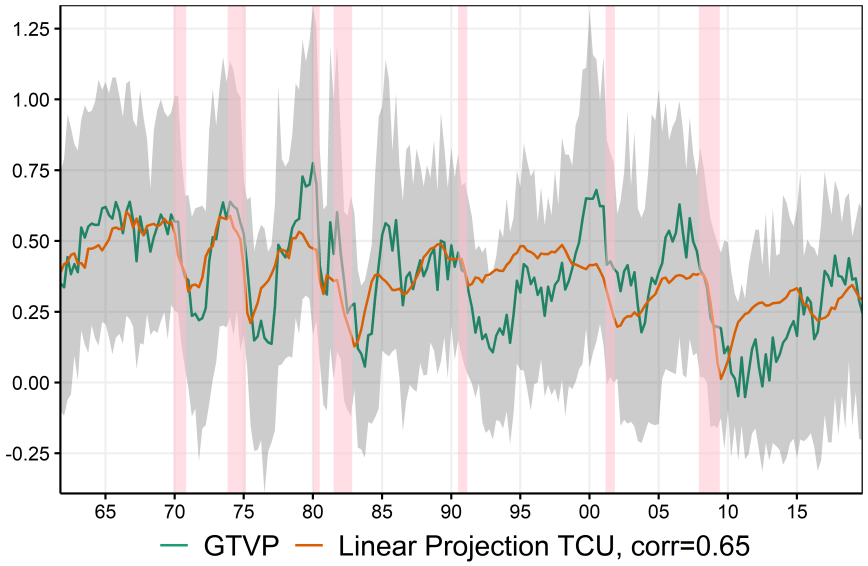


Figure 34: $\beta_{3,t}$ in (2.5) with additional controls for supply and monetary policy shocks. Capacity Utilization is still substantially correlated with the inflation-unemployment trade-off. The grey band is the 68% credible region. Pink shading corresponds to NBER recessions.

Table 19: Monthly Results

	AR4	AO-12	AO-h	FAAR	RF	RF-MAF	AR+RF	ARRF	FA-ARRF	Tiny ARRF	VARRF
IP											
h=1	1.00	1.11*	1.14	0.96	1.03	**0.94***	0.97	0.99	0.96	1.02	1.02
h=3	1.02	1.17*	1.02	0.99	1.12	0.98	**0.96**	1.03	1.01	1.02	1.08
h=9	1.01	1.04	1.03	1.06	1.02	1.06	1.02	1.04	1.10	1.09	1.03
h=12	1.01	1.00	1.00	1.05	0.99	0.97	**0.91**	0.97	1.05	1.13	0.96
h=24	1.00	**0.84**	0.84	1.17	0.92	0.86	0.86	0.88	0.95	1.11	0.89
UR											
h=1	1.01	1.03	1.09	0.95	0.97	**0.87*****	0.95	0.91***	0.90**	0.98	0.94**
h=3	1.00	1.10	1.05	0.86	1.05	**0.81*****	0.92	0.89**	0.82*	1.03	0.89***
h=9	0.99	1.11	1.10	0.92	1.02	0.96	**0.91**	0.97	0.98	1.16*	0.97
h=12	0.99	1.07	1.07	0.96	0.97	0.96	**0.91**	0.99	0.94	1.17	0.96
h=24	1.02**	1.02	1.03	1.06	0.91*	0.84	**0.81**	0.91	0.97	1.28	0.87
SPREAD											
h=1	0.99	2.88***	1.23***	1.21**	3.52***	1.07	**0.91*****	0.99	0.98	0.96	0.93**
h=3	1.01	1.68***	1.07	1.25	1.69***	0.82**	**0.81*****	1.06	0.85**	1.00	0.88**
h=9	1.01	1.36	1.27	1.06	0.94	0.73**	0.72**	0.70***	**0.62*****	1.07	0.67***
h=12	1.02	1.28	1.28	1.05	0.80***	0.66***	**0.60*****	0.68***	0.65***	1.07	0.64***
h=24	1.03	1.34*	1.34*	0.96	0.80*	0.70*	0.71*	0.69**	**0.63*****	0.90	0.70**
INF											
h=1	1.02	1.11*	1.18*	0.99	1.07	1.06*	1.01	0.95	0.96	0.95	**0.93****
h=3	1.04	1.02	1.24*	1.04	0.93	**0.88**	1.05	0.90	0.88	0.90	0.88
h=9	1.07	0.92	1.01	1.16	0.86	0.78	1.15*	**0.72**	0.82	0.73	0.76
h=12	1.09*	0.91	0.91	1.21	0.88	0.79	1.15*	0.73	0.67	**0.67***	0.70
h=24	1.04	0.90**	0.86**	1.35	1.00	1.12	1.12	0.71	0.69	**0.55****	0.73
HOUST											
h=1	**1.00**	1.10**	1.35***	1.07	1.08**	1.02	1.00	1.01	1.02	1.02	1.01
h=3	**0.96****	1.06	1.34***	1.15	1.03	1.07	1.03	1.04	1.03	1.01	1.04
h=9	**0.98**	1.05	1.12	1.35	0.98	1.02	1.01	1.02	1.14	1.03	1.03
h=12	0.98	1.05	1.05	1.32	**0.95**	1.00	1.01	1.00	1.12	1.11	1.03
h=24	0.95	1.09	1.07	1.17	**0.87**	0.94	0.95	1.00	1.15	1.23	1.06

Notes: This table report the root MSPE of the model m with respect to the root MSPE the AR(4). Best forecast of the row is in bold. Diebold-Mariano test is for each model against the AR(4). "*", "**" and "***" means p-values of below 10%, 5% and 1%. "AO-i" means i-months moving average forecasts à la Atkeson et al. (2001).

Table 20: Main Quarterly Results

	FA-AR	LASSO-MAF	Ridge-MAF	RF	RF-MAF	AR+RF	Tiny RF	FA-ARRF	ARRF	Tiny ARRF	VARRF	SETAR	STAR	TV-AR
GDP														
h=1	1.02	0.96	0.89**	0.94	0.86	0.89	1.03	**0.86**	0.93	1.04	1.20	1.01	1.03	0.99
h=2	0.96	0.98	0.98	0.99	**0.91**	0.93	1.01	0.97	0.94**	1.03	0.99	0.97	0.98	1.03
h=4	1.03	0.98	0.99***	1.00	0.98	0.99	1.03	0.97	0.95	0.98	**0.89**	0.97***	0.96***	0.96
h=6	1.36	0.98	0.98	0.98	1.00	1.00	1.08	1.01	0.97	0.98	1.00	0.98	**0.95**	0.98
h=8	1.37	1.00	0.99	0.99	0.99	**0.96**	1.15	1.06	1.00	1.01	1.04***	1.00	0.97	1.00
UR														
h=1	0.83	0.99	0.99	1.00	0.85*	0.84	1.24**	**0.72**	0.90***	1.00	1.24	1.18	1.10	1.00
h=2	0.80	0.98	0.92*	0.98	0.85	0.84	1.15*	**0.76**	0.90	0.96	0.89	1.03	0.97	0.99
h=4	0.88	0.96***	0.94**	0.96*	0.87*	0.84*	1.37	**0.79**	0.87	0.92	0.91	1.02	1.01	1.34
h=6	1.18*	0.98	0.98	1.01	0.94	0.90	1.60*	**0.89**	0.95	0.97	0.95	1.07	1.04	1.14
h=8	1.25	0.98	1.01	1.01	**0.95**	0.95	1.57	1.01	0.98	0.98	1.04	1.09	1.06	1.11**
SPREAD														
h=1	1.28	2.16***	0.93	0.91	0.95	0.79**	0.96	1.08	0.89**	1.06	**0.77****	1.51***	1.53***	0.98
h=2	1.13	1.20	0.77	**0.66****	0.78	0.72***	0.93	0.80	0.78**	1.11	0.74**	1.19	1.20	1.04
h=4	0.86	0.95	1.01	0.81	0.69**	**0.61****	1.48*	0.66**	0.73**	1.07	0.69**	1.04	1.06	1.30
h=6	1.51	0.80*	1.13	0.98	0.80	0.80	1.43	**0.72****	0.82	1.05	0.74*	1.03	1.06	1.19
h=8	1.28	**0.76****	0.96	0.92	0.83	0.89	1.36	0.82	0.88	0.99	0.85	1.11	1.14	0.99
INF														
h=1	1.01	0.93	0.95	0.98	0.88	1.23	0.90	0.94	0.89	**0.87***	0.96	1.05	1.00	0.93
h=2	1.01	0.96	0.92	0.92	**0.82**	1.00	0.88	0.94	0.86	0.87	0.91	0.86*	0.86	0.89
h=4	1.08	0.92	0.87	0.94	**0.85****	0.96	0.86	0.89	0.91*	0.95*	0.87*	0.90*	0.87*	0.91
h=6	1.32	0.96	0.90	1.01	0.88	1.00	0.86	0.91	**0.85**	0.92**	0.87	0.94	0.89	0.98
h=8	1.21	0.98	1.27	1.44	**0.88***	0.94	0.88	0.91*	0.92	0.94	0.91*	0.96	0.92	0.98
HOUST														
h=1	1.13	1.04	0.94*	**0.92***	1.00	1.01	1.24***	1.08	0.94**	0.95	1.09	1.01	0.99	1.00
h=2	1.13	0.99	0.94**	0.95*	1.01	1.02	1.10*	1.06	1.00	1.02	0.99	**0.94**	0.97	1.01
h=4	1.11	0.98**	0.97*	0.97	1.01	1.03	1.12	1.02	1.00	1.02	1.02	**0.95**	0.96	1.08
h=6	1.40	0.96	0.96	0.96	0.96***	1.01	1.16	0.97***	0.99	1.00	0.98	**0.95**	0.96	0.99
h=8	1.04	0.95	0.95	0.95	0.99	1.02	1.44	0.96	0.99	1.01	1.00	**0.95**	0.95	1.03
IR														
h=1	1.85	1.02	1.55	1.17	1.11	0.97	0.99	1.29	0.94	**0.92**	1.43	1.39	1.20	0.97
h=2	1.49	0.96	1.01	1.00	0.93	0.98	1.29***	1.22	0.93	**0.92**	1.10	1.15	1.11	1.04
h=4	**0.96**	1.00	1.03	1.03	1.04	0.99	1.39*	0.99	0.97	1.12	0.97	1.08	1.07	1.09
h=6	1.87	0.95	0.99	1.00	**0.93**	0.93	1.23*	0.98	0.95*	1.07	1.12	1.19	1.14	1.06**
h=8	1.58	0.98	1.02	1.03	**0.96**	0.96	1.20	1.04	0.96	1.10	0.98	1.25**	1.20**	1.06

Notes: This table report the root MSPE of the model m with respect to the root MSPE the AR(4). Best forecast of the row is in bold. Diebold-Mariano test is conducted for each model against the AR(4). "*", "**" and "***" means p-values of below 10%, 5% and 1%.

CHAPTER 3 : TO BAG IS TO PRUNE

3.1. Introduction

Random Forests (RF) is at the forefront of Machine Learning (ML) applications to economics. It can successfully predict asset prices (Gu et al., 2020), house prices (Mullainathan and Spiess, 2017), and macroeconomic aggregates (Medeiros et al., 2019; Chen et al., 2019; Goulet Coulombe et al., 2019). It can infer treatment effect heterogeneity (Athey et al., 2019), and estimate generalized time-varying parameters (Goulet Coulombe, 2020b). The list goes on. But what makes it so infallible? To answer that question, and eventually understand the reasons behind RF's growing list of successful econometric applications, it is better to start with an apparent paradox.

Common statistical wisdom suggests that a non-overfitting supervised learning algorithm should have approximately the same mean squared error in the training sample as in the test sample. LASSO, Splines, Boosting, Neural Networks (NN) and MARS abide by that principle. But not Random Forest. It is the norm rather than the exception that RF has an exceptionally high in-sample R^2 with a much lower, yet competitive, out-of-sample one. This means not only do the individual trees overfit the training set, but that the ensemble does, too.[1] In contrast, the algorithms mentioned above usually perform poorly in such conditions. When optimally tuned, they are expected to deliver neighboring R^2_{test} and R^2_{train}. This chapter is about understanding why RF is excused from obeying this rule — and showing how to leverage this property for other algorithms.

In hope of rationalizing Breiman (2001)'s algorithm's overwhelming success and ever-increasing popularity, myriad recent academic work has investigated RF's theoretical properties. As one would expect, the first matter on the agenda was consistency, with the most authoritative contribution to date being Scornet et al. (2015).[2] Consistency is a crucial property that any useful supervised algorithm should possess, but it is merely a necessary condition for it to be included in the toolbox of the data scientist (with a statistical sensibility). For a generic learning task, finite sample properties (usually unknown) determine which algorithms are preferred in practice. That is, it is still theoretically unclear why RF works so well, on so many data sets. It is acknowledged that much of that resilience is attributable to RF providing a very flexible non-linear function approximator that does not overfit. Most importantly, unlike many models of the non-parametric family, the latter characteristic is (almost) guaranteed even without resorting to carefully tuning

[1] This motivates the use of out-of-bag measures rather than in-sample fit to interpret the model.

[2] For a detailed review of the consistency journey and other relevant theoretical aspects, see Louppe (2014).

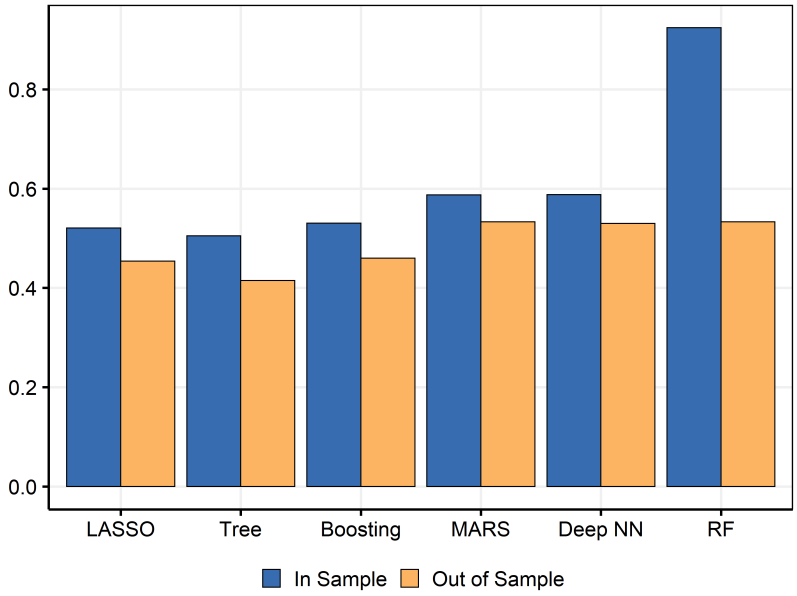


Figure 35: *Abalone* data set: comparing R^2_{train} and R^2_{test} for classic models. First four models hyperparameters are tuned by 5-fold cross-validation. RF uses default parameters. Deep NN details are in Appendix 3.6.4.

hyperparameters. Yet, it is still not clear what mechanism is behind this phenomenon.

Providing a theoretical reason to believe RF will not overfit, Breiman (2001) himself shows that as the number of trees B grows large, the generalization error is bounded. That bound goes down as the individual learners' strength increases and goes up as correlation between them increases. Taking a perpendicular direction, Taddy et al. (2015) view the forest prediction as a posterior mean over an empirical distribution of trees obtained by an approximation to Rubin (1981)'s Bayesian Bootstrap. In this paradigm, a tree is merely a single draw from a posterior distribution, which provides a "hammer" argument basis for never using only one tree.

Yet a question remains unanswered: if RF – made of fully grown completely overfitting trees – does not overfit out-of-sample, where does regularization come from? Clearly, increasing λ brings regularization in a ridge regression by shrinking coefficients toward zero, lowering the individual importance of each predictor. When it comes to RF, what contortions on the intrinsic model does its regularization entail? An appealing answer is that bagging smooths hard-thresholding rules (Bühlmann et al. (2002)), like increasing the smoothness parameter of smoothing splines. If that were the whole story, RF, as does smoothing splines, would yield comparable R^2_{test} and R^2_{train}. Model averaging arguments

would also have a similar implication.[3] As clearly displayed in Figure 35, it is not the case — so something else must be at work.[4] The newly proposed answer is: to bag (and perturb) is to prune. But not a tree. Rather, RF implicitly prunes a latent true underlying tree.

More generally, I argue that randomized greedy optimization performs early stopping. This is interesting since greedy optimization is often introduced in statistical learning books as an inevitable (but suboptimal) practical approach in the face of computational adversity (Friedman et al., 2001). It turns out the necessary evil has unsuspected benefits. A greedy algorithm treats what has already happened as given and what comes next as if it will never happen. While this depiction usually means "trouble", it is the key to this chapter's argument. By recursively fitting a model and *not* re-evaluating what came before as the algorithm progresses, the work of early stages will be immune to subsequent overfitting steps, provided the latter averages out efficiently. Mechanically, in a greedily fitted tree, the estimated structure at the top cannot be weakened by the bottom's doings – the bottom's existence is not even considered when estimating the top.[5] Moreover, when faced with only noise left to fit in a terminal node, it is shown that a *Perfectly Random Forest*'s out-of-sample prediction is the sample mean, which is unbiased and has – mostly importantly – minimum variance. In short, it performs *pruning*.

Fortunately, not only trees are eligible for the enviable property, but also other greedily fitted additive models like Boosting and Multivariate Adaptive Regression Splines (MARS). Based on this observation, I propose *Booging* and *MARSquake* which – like RF – are ensembles (of bagged and perturbed base learners) that completely overfit the training sample and yet perform nicely on the test set. Those are later shown to be promising alternatives to Boosting and MARS (both with a tuned stopping point) on real and simulated data sets. An R package implements both.

Finally, it is worth contrasting this chapter's explanation with recent "interpolating regime" and "double descent" ideas proposed to explain the success of deep learning (Belkin et al., 2019a,b; Hastie et al., 2019; Bartlett et al., 2020; Kobak et al., 2020). First, some terminology. The interpolating regime is entered whenever one fits an algorithm on the training data past the point where $R^2_{\text{train}} = 1$.[6] The double descent is the astonishing observation

[3] This renders incomplete (at best) arguments linking RF regularization to that of penalized regression (originally discussed in Friedman et al. (2001), and more recently Mentch and Zhou (2019)) using results developed for *linear* models (Elliott et al., 2013; LeJeune et al., 2020).

[4] Mullainathan and Spiess (2017)'s Table 1 – reporting results from off-the-shelf ML algorithms applied to house price prediction – is another convenient example where all aspects of the phenomenon are visible.

[5] In sharp contrast, all parameters are estimated simultaneously in a linear regression.

[6] Interpolation means training data points are effectively interpolated by the fitted function $\hat{f}$ when $R^2_{\text{train}} = 1$.

that for large-scale deep neural networks (DNN), the out-of-sample performance starts to increase past the point where $R^2_{\text{train}} = 1$. Preceded by the typical U-shaped empirical risk curve implied the classical bias–variance trade-off before $R^2_{\text{train}} = 1$, this makes it for a "double descent" — the first starting from $R^2_{\text{train}} = 0$ and the second from $R^2_{\text{train}} = 1$. Belkin et al. (2019a) evoke that the phenomenon is also present in RF. However, they mistakenly associate the number of trees to be increasing complexity (as in Boosting, whereas it is really increasing averaging/regularization in RF). Thus, there is no double descent, but rather a single descent that never ascent as complexity increases – in line with this chapter's argument. Wyner et al. (2017) also argue that interpolation may be the key for Boosted Trees and RF success because local fitting of dissident data points prevents harming the overall prediction function $\hat{f}$. But it is unclear as to why RF is so proficient at it, why "locality" emerges in the first place, and why estimation variance does not spread. The current chapter makes exactly clear how the (greedy) construction of RF guarantees that overfitting washes away out-of-sample.

This chapter is organized as follows. In section 3.2, I present the main insights and discuss their implications for RF and other greedy algorithms. In section 3.3, I demonstrate by means of simulations the implicit optimal early stopping property of RF, Booging and MARSquake. Section 3.4 applies the chapter's main ideas to classic regression data sets. Section 3.5 concludes.

3.2. Randomized Greedy Optimization Performs Early Stopping

The key ingredients for an ensemble to completely overfit in-sample while maintaining a stellar generalization error are (i) the base learner prediction function is obtained by greedy/recursive optimization and (ii) enough randomization in the fitting process. Section 3.2.1 explains why their combination generates the observed phenomenon. Clearly, RF satisfies both above requirements and is used as the leading example.

Multivariate Adaptive Regression Splines (MARS, Friedman (1991)) and Gradient Boosting (Friedman (2001), Friedman (2002)) are also eligible for implicit early stopping. In fact, for any forward stagewise regression procedure which can generate enough randomness in the recursive model building pass *and* does not re-evaluate previously estimated coefficients as the model complexity increases, we can barter the choice of a stopping point for an ensemble of bootstrapped and perturbed overfitting base models. In subsection 3.2.2, I detail when we can expect both criteria to be fully met and when not. No base learner perfectly satisfies both requirements, but some get closer than others.

In simulations (section 3.3), it is found that bagged and perturbed fully overfitting MARS/-Boosting performance is quantitatively equivalent to that of stopping the respective base learner at the ex-post optimal stopping point. Classification and Regression Tree (CART,

Breiman et al. (1984)) also inherits the property that letting trees grow to their full extent will not cause any harm. However, that performance is much higher than that of an optimally pruned plain CART. Consequently, it is clear that, unlike MARS and Boosting, RF is pruning something else than the base learner itself. In subsection 3.2.3, I use an analogy to time series econometrics to argue that bagging trees (Breiman, 1996) works so well because it is recovering a true latent tree. In combination with other ideas to be presented in this section, this implies that RF is not merely pruning CART. Rather, it is a self-pruning latent tree.

3.2.1. The Miracle of Randomized Recursive Fitting

It is common to see that RF will have R^2_{train} magnitudes higher than R^2_{test}, a symptom which would suggest overfitting for many standard algorithms. That is, the traditionally defined in-sample fitted values

$$\hat{y}_i^{\text{RF}} = \frac{1}{B}\sum_{b=1}^{B} \hat{y}_{i,b}^{\text{tree}}$$

and corresponding residuals have nothing to do with what one gets when applying the estimated model to new data, unless the "true" R^2 is really high.[7] While this R^2_{train} curiosity is usually of limited interest *per se*, it creates some intriguing headaches from a more traditional statistical perspective. For instance, any attempt to interpret the intrinsic RF *model* relies on measurements obtained on pseudo hold-out samples (called *out-of-bag*). In contrast, one would not refrain from exploring the structure of MARS' fitted values or that of a single tree. Indeed, most algorithms, when properly tuned, will produce comparable R^2_{test} and R^2_{train}. This implies that using the in-sample conditional mean $\hat{y}_i$ for any subsequent analysis is perfectly fine. In that way, they behave similarly to any classical nonparametric estimators where a bandwidth parameter must be chosen to balance estimation flexibility and the threat of overfitting. Once it is chosen according to CV or some information criteria, in-sample values provide reliable estimates of the *true* conditional mean and error term.

I argue that RF's notably different behavior can be explained by the combination of two elements: greedy optimization and randomization of the recursive model fitting sequence. By construction, the instability of trees makes the latter an easy task: simply bootstrapping the original data can generate substantially different predictors.[8] The former, greedy optimization, is usually seen as the suboptimal yet inevitable approach when solving for a global solution is computationally unthinkable. In this section, I argue that greedy op-

[7] B is the total number of base learners.

[8] As argued in Breiman (1996), not every algorithm responds as vividly as trees to small perturbations of the original sample. Section 3.2.2 expands on that and compare trees' ability to generate inner randomness (within the fitting procedure) to that of Boosting-like model building procedures.

timization, when combined with randomization of the model building pass, has an additional benefit. When combined in a properly randomized ensemble, no harm will come in letting each greedily optimized base learner completely overfit the training sample. In the case of RF, this translates to the heuristic recommendation of considering fully grown trees where each terminal node contains either a single observation or very few.

What Happens in the Overfitting Zone Stays in the Overfitting Zone

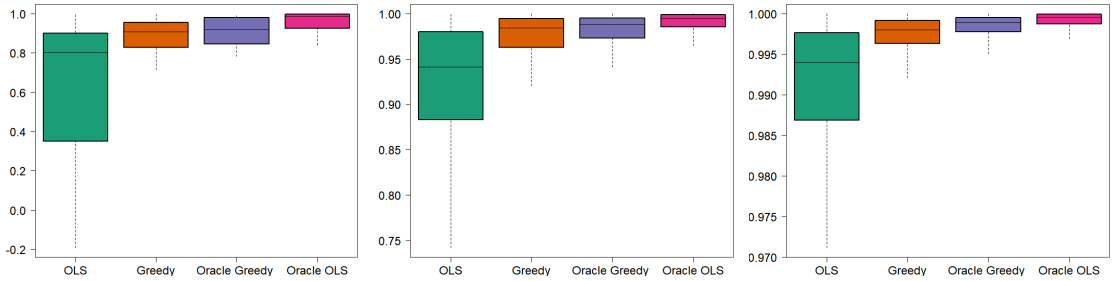
In a global estimation procedure, overfitting will weaken the whole prediction function. More concretely, estimating many useless coefficients in a linear regression will inflate the generalization error by increasing the variance of *both* the few useful coefficients and the useless ones. Bagging such a predictor will still be largely suboptimal: the ensemble will still rely on an average of coefficients which are largely inferior to those that would be obtained from regression excluding the useless regressors. Hence, we are still in the standard case where $R^2_{\text{test}} < R^2_{\text{train}}$ reveals that the predictor performance is inferior to that of an optimally pruned counterpart.

A greedily optimized model works differently. At each step of the forward pass, everything that came before is treated as given and what comes next as if it will never happen. The essential insight of this section resides in the first attribute. That is, as the algorithm progresses past a certain step s, the function estimated before s is treated as given. Inevitably, the algorithm will eventually reach s^* where the only thing left to fit is the unshrinkable "true" error $\epsilon_i = \hat{\epsilon}_{i,s^*} = y_i - \hat{f}_{s^*}(x_i)$. The key is that entering deep in the overfitting zone will not alter $\hat{f}_{s-1}$ since it is not re-evaluated. As a result, early non-overfitting steps can be immune to the weakening effect of subsequent ones, as long as the latter efficiently averages out to 0 in the hold-out sample. An immediate implication of this separability property (which is argued to be particularly strong for trees in section 3.2.2) is that there is no need to stop the forward pass at the unknown s^* to obtain optimal predictions.

These abstract principles can be readily applied to think about fitting trees where a step s is splitting the subsample obtained from step $s-1$. As more formally put in section 3.2.3, a tree does not distinguish whether the current sample to split is the original data set of the result of an already busy sequence of splits. Moreover, like any splits along the tree path, those optimized before venturing past s^* cannot be subsequently revoked. This implies that the predictive structure attached to them cannot be altered nor weaken by ulterior decision the greedy algorithm makes.

Alternatively, we can think of fitting a linear regression with orthogonal features. A step s is adding a regressor by fitting it to the residual of the previous step. In the ($\sim$boosting) linear regression case, we can hope that important predictors will go in very soon in the

process and will be followed by a sequence of useless predictors until those are exhausted. Unlike the coefficients from the all-at-once-OLS, the early fitted coefficients in the forward pass of the stagewise algorithm were estimated as part of a model that only included a handful of predictors. In effect, those are precluded from the eventual weakening effect that comes with the inversion of a potentially near-singular $X'X$.[9]



(a) $B = 10$ (b) $B = 100$ (c) $B = 1000$

Figure 36: R^2_{test} between the a toy algorithm's prediction on the optimal prediction (x_1) for varying number of replication samples B. **Notice the varying y-axis scales.**

A toy simulation can help visualize the phenomenon. Suppose the DGP is $y = x_1 + \epsilon$ and there are 2 regressors available x_1 and x_2. To make this a high-dimensional problem, *there are only 3 observations*. Thus, by construction, there is a sizable benefit to stopping after including a single regressor. I compare the performance of four approaches. First, the *Oracle OLS* which is an ideal regression including only x_1. Second, *Oracle Greedy* which is a recursive (two-step here) scheme that knows the first variable to go in is x_1. Third, *Greedy* is the same as the previous one, but the first variable to go in is the one with the highest correlation with y. Fourth, *OLS* is simply a regression including both regressors. The performance statistic reported in Figure 36 is the distribution of R^2_{test} between a toy algorithm hold-out sample prediction and the true conditional mean (x_1), obtained over 100 simulations.[10] Finally, B is the number of independent samples on which base learners are fitted and then averaged. $B = 1000$ represent the ideal case where it is somehow very easy to randomize the 3 observations regression and $B = 10$ is closer to a rough/imperfect randomization scenario. Undeniably, even when averaged over B independent samples, OLS is largely suboptimal to the Greedy approaches. Both perform rather similarly, with R^2_{test}'s that are themselves very close to the optimally pruned Oracle OLS.

[9] Adding a ridge penalty will alleviate the singularity problems, but will also (potentially heavily) shrink the real coefficients of interest, compromising their predictive power.

[10] Each test set comprise 1000 observations.

Bagging and Perturbing the Model as an Approximation to Population Sampling

At s^*, which corresponds to the *true* terminal node in the case of a tree, the DGP is simply

$$y_i = \mu + \epsilon_i. \tag{3.1}$$

Clearly, the best possible prediction is the mean of all observations contained in the node. I argue that perfect randomization will also procure this optimal prediction out-of-sample, even if the ensemble itself is completely overfitting in-sample. This *Perfectly* Random Forest is, of course, merely a theoretical device and how close RF gets to this hypothetical version is an empirical question. Nevertheless, it is widely believed (and further confirmed in section 3.3) that bagging (`B`) and perturbing (`P`) trees can get very close to what would obtain from population sampling.[11] By the latter, I mean that each tree is grown on non-overlapping samples from a population. Essentially, this is what bootstrapping any statistic is meant to approximate. It is nothing new to display that a predictor's performance improves when averaging it over many close-to independent samples.[12] The more subtle point being made here is that a good approximation to population sampling (via `B` & `P`) can generate a predictor whose structure will be close to the optimally pruned one, and that, without attempting any form of early stopping whatsoever. In other words, (3.1) more generally represents the truth from the hypothetical point s^* where a recursive fitting algorithm should optimally stop or otherwise enter the overfitting zone.[13] Proper inner randomization assure that a prediction close to $\bar{y}$ is returned.

It is known that perfectly uncorrelated trees will have a bounded generalized error (Breiman (2001)). It is a trivial (but nevertheless interesting) byproduct to show that given a large number of samples B, the prediction of a `B` & `P` ensemble of completely overfitting trees that achieve perfect randomization is the (optimal) sample average. This enlightening result is possible by taking the recursive view and assuming perfect randomization. First, the out-of-sample prediction of a RF for observation j is

$$\hat{\mu}_j^{\text{RF}} = \frac{1}{B} \sum_{b=1}^{B} \hat{\mu}_{j,b}$$

where $\hat{\mu}_{j,b}$ is the prediction of the b tree for observation j. Importantly, observation j is not included when fitting the trees, so we are looking at the prediction for a new data point

[11] By perturbing trees, I refer to randomly selecting `mtry` features to be considered for a given split – as implemented by RF. Of course, there exists other perturbation schemes, like injecting noise. However, `mtry` is by far the most widely used, at least, for trees.

[12] In Breiman (2001)'s well-known equation, this corresponds to the case where $\rho = 0$, i.e., all the trees are perfectly uncorrelated.

[13] In the case of MARS and Boosting, $\mu = 0$.

using a function trained on observations $i \neq j$. For simplicity, assume fully grown trees, which means terminal nodes include a single observation.[14] The model is applied to the terminal node DGP in (3.1). Since the tree is fitting noise, perfect randomization implies that each out-of-sample tree prediction is a randomly chosen y_i for each b. The prediction is thus

$$\hat{\mu}_j^{\text{RF}} = \frac{1}{B} \sum_{b=1}^{B} y_{i(b)}.$$

Define $r = {}^{B}/_{N}$ where N is the number of training observations and r will eventually stand for "replicas". Since the $y_{i(b)}$'s amount to random draws of $y_{1:N}$, for a large enough B, we know with certainty that the vector to be averaged ($y_{i(1:B)}$) will contain r times the same observation y_i. Hence, the prediction equivalently is

$$\hat{\mu}_j^{\text{RF}} = \frac{1}{B} \sum_{i=1}^{N} \sum_{r'=1}^{r} y_{i,r'} = \frac{1}{B} \sum_{i=1}^{N} \sum_{r'=1}^{r} y_i = \frac{r}{B} \sum_{i=1}^{N} y_i.$$

Finally, remembering that $r = {}^{B}/_{N}$ yields

$$\hat{\mu}_j^{\text{RF}} = \frac{1}{N} \sum_{i=1}^{N} y_i.$$

In words, when a Perfectly Random Forest is starting to fit pure noise, its out-of-sample prediction averages out to the simple mean, which is *optimal* under (3.1) and a squared loss function. Intuitively, at s^*, the test set behavior of the prediction function (from fully-grown trees) is identical to that of doing (random) subsampling with subsamples containing one observation. Averaging the results of the latter (over a large B) is just a complicated way to compute a mean. Hence, the out-of-sample prediction as provided by the perfectly random forest is one where implicit/automatic pruning was performed.[15] It is equivalent to that of an algorithm which knows the "true" s^*. gives A direct implication is that we need not to worry about finding s^* through cross-validation, since the optimally stopped prediction is what being reported out-of-sample. Of course, this result's usefulness relies on perfect randomization being somehow empirically attainable. Section 3.3 will ask "How close to population sampling are we when fitting `B` & `P` trees?" and the answer will be "surprisingly close".

One may wonder what is the connection at a deeper level between this form of implicit reg-

[14] This also directly implies that each base learners' R^2_{train} is one and that of the ensemble is bounded below by the subsampling rate, which will inevitably be much higher than $R^2_{\text{test}} = 0$.

[15] This provides a justification for Duroux and Scornet (2016)'s finding that pruning the base learners while shutting down `B` can deliver a performance similar to that of plain RF (provided a wise choice of tuning parameters).

ularization and cross-validation itself. Both aim at improving the capacity of the model to generalize by relying on a bootstrap approximation to population sampling. By construction, the true conditional mean should be systemically present in every data set whereas the noise vectors differ. Hence, CV chooses the optimal s^* by assuming that anything that happens beyond that point is not systematic and will not help in predicting new instances, as revealed by the k-fold MSE for different values of s^*. The implicit pruning detailed above operates on a similar heuristic. Each forward model building pass is done on a pseudo-new data set and averaging those should make the signal (conditional mean) stand out and the overfitted noise average out.[16] Intuitively, if step s's addition to the model is meaningful and systematic, it should show up similarly in many bootstrap replicas. If it is not, it won't. Recursive fitting allows this averaging to happen piece by piece rather than for the whole model at once, which deliver the desirable outcome that letting the base learners (as well as their ensemble) completely overfit is practically costless.

The above also helps in understanding $R^2_{\text{test}} < R^2_{\text{train}}$ in RF. The gap's existence is a direct implication of implicit pruning via `B` & `P` being only active *out-of-sample*. Population sampling itself does not generate $R^2_{\text{test}} < R^2_{\text{train}}$[17], only its approximation by `B` & `P` does. A central role in this is that of `mtry` – the number of randomly selected features to be considered for a split. Overfitting situations can be thought of as an overabundance of parameters vs observations. The attached predictors are either directly available in the data or created via some form of basis expansions which trees is one (successful) possibility out of many. In such high-dimensional situations, it is clear that the model itself – the predictive structure – is barely identified: many different tree structure can rationalize a training sample with $R^2_{\text{train}} = 1$. Yet, these predictive structures will generate substantially different predictions when applied to new data.[18] This property of overfitting predictors (combined with the recursive fitting procedure) is the channel through which `mtry` strongly regularize the hold-sample prediction. However, the heterogeneity in structures that `mtry` generates cannot deflate R^2_{train} since *different* overfitting base learners, when trained on the same data, provide the *same* fitted values.[19] Ergo, $R^2_{\text{test}} < R^2_{\text{train}}$.

Not Your Average Model Averaging

At first sight, this may seem new like nothing new: RF successfully controls overfitting by approximating more resampling by model averaging. The latter is known to provide

[16] Of course, this is not perfect as those data sets overlaps, which leaves an important role for `mtry`.

[17] As the subsampling rate mechanically decrease – a luxury obtained from a growing sample size and fixed model complexity – the R^2_{train} itself will look much like a true R^2_{test} since the contribution of observation i to its out-of-bag prediction shrinks with subsampling rate.

[18] A simpler example is that of ridge regression: if there are more regressors than observations, then $\lambda \to 0$ leads to a multitude of solutions for β and the non-identification of the predictive structure.

[19] Namely, they all report y itself or something close to it.

a sort of regularization that can in some special case be equivalent to more traditional shrinkage estimator (Elliott et al., 2013; LeJeune et al., 2020). The success of simple forecast combinations schemes have been detailed extensively over the last decades (Clemen, 1989). A classic application of the forecast combinations literature is that of combining forecasts from professional economic forecasters. In this setup, it is plausible that forecasts may differ because of different models *and* different information sets (Timmermann, 2006). Following along the idea, one can think of RF as a well-designed one-model and one-training set way of mimicking an ensemble of forecasters that puts heterogeneous weights on observations (`B`) and assign different priorities to different regressors (`P`). What is new is that unlike averaging a kitchen-sink OLS regression (for instance), a greedy algorithm makes the structure estimated before s^* immune to what happens in the overfitting zone. In contrast, schemes like that of Eliasz et al. (2004) or those discussed in Rapach and Zhou (2013) imply directly or indirectly tuning the number of regressors in the base learner linear models. This means that including too many of them could damage the overall predictor's performance.[20] This is analogous to that of tuning a ridge regression and, as one would expect from this link, the resulting R^2_{test} is usually in the neighborhood of R^2_{train}.[21] Hence, `B` & `P` are not the source of the "paradox" in themselves: they need to be paired with a greedy algorithm which has the potential to generate sufficient inner randomization.

Conversely, the ideas presented above could help at understanding why forecast combinations work so well, which unlike confirming their multiple successes, is still an ongoing venture (Timmermann, 2006). It is plausible that individual forecasters construct their predictive rule in an inductive recursive fashion. That is, human-based economic forecasting has likely more to do with a decision tree (based on looking at multiple time series plots) or a stepwise regression, than with the solution of a global problem (like OLS, LASSO and others). Indeed, it is arguably much easier to learn in a greedy fashion (both for a human and a computer) than to solve a complex multivariate problem directly for its global solution. Thus, assuming underlying forecasts are constructed as such, the average will behave in a very distinctive way if those are overfitting. As argued in Hellwig (2018) for the survey of IMF forecasters, the latter assertion is very likely true. As a result, the discussion above provides yet another explanation for the success of forecast combinations (especially the simple average scheme): significant inner randomization combined with recursively constructed overfitting forecasts provides implicit (and necessary) pruning. This is not a replacement but rather a complement to traditional explanations (usually for linear models)

[20] Also, see Figure 36.

[21] This is why using a (global) linear model to think about `mtry`'s effect – while it may yield interesting insights (Mentch and Zhou, 2019) – provides an incomplete answer that fails to capture one of RF's most salient regularities: $R^2_{\text{train}} > R^2_{\text{test}}$.

that link the effects of model averaging to traditional shrinkage estimators.[22]

3.2.2. Leveraging the Insight for Other Models

Not only trees require pruning. Many additive schemes must be optimally stopped at s^* to obtain the best test set performance. It has been discussed that mixing `B` & `P` with a greedy recursive algorithm can lead the algorithm to perform implicit early stopping. It is natural to wonder if certain well-known greedy model building algorithms could also benefit for this property. In sections 3.3 and 3.4, I consider `B` & `P` versions of MARS and Gradient Boosted Trees. Before jumping to do so, I discuss why they can plausibly benefit from it, but perhaps not as much as trees. As a guiding light through this discussion, here are three conjectured commandments for `B` & `P` to implicitly perform early stopping on an algorithm:

C1. It is greedy;

C2. It generates enough randomization so the ensemble consists of diversified predictors;

C3. The incremental improvement steps (in the model space) must not be too big.

It is clear that RF satisfies them all. Point 1 and Point 2 suggests that some form of *fully overfitting* Boosting and MARS could equivalently benefit from `B` & `P`. However, the success of such an enterprise is bounded by the ability of the algorithm to satisfy Point 2. Indeed, if only an insufficient amount of randomization is generated by the model building scheme, there will be benefits from stopping base learners earlier. Point 3 excludes the option of fitting neural networks greedily, layer by layer, as popularized in Bengio et al. (2007). Indeed, there is little hope that an overfitting ensemble will yield a decently behaved average since adding a layer can hardly be described as an incremental improvement in the model space.[23]

Why Does it Work Best for Trees?

The usual answer is that trees are unstable and bagging works best for unstable prediction rules. Of course, this is framed in terms of increasing predictive accuracy, while this chapter rather focuses on the implicit pruning property. I revisit the "instability" argument in a slightly different light. I ask: which prediction rule, when bootstrapped/perturbed generates enough inner randomness for an ensemble of completely overfitting base learners not

[22] For instance, Friedman et al. (2001) discuss the link for RF itself, Rapach and Zhou (2013) discuss it for the case of forecasting stock returns with averages of linear models.

[23] Often, a single layer has enough parameters to totally overfit a data set. It is not excluded, though, that such a deep neural net with an astonishing number of heavily regularized layers could work. Given the often overwhelming complexity of optimizing hyperparameters in such models (and good results depending on very specific values of them), this is an important avenue to explore in future work – and could be contrasted with deep "interpolating" NNs which avoid overfitting for reasons of their own (Olson et al., 2018; Belkin et al., 2019a). Furthermore, the general idea of mixing elements of deep learning (usually multiple layers) with the robustness of forests has gain some traction in recent years (Zhou and Feng, 2017; Feng et al., 2018).

to overfit in the hold-out sample? To provide an answer, I compare trees and a boosting-like procedure by looking at both as additive models. In yet another episode of "what does not kill you makes you stronger", what makes trees weak is what provides them with this enviable pruning property. MARS and Boosting can be "weakened" in a similar fashion, but not as much.

The reason why trees generate so much randomization is the *irreversibility* of the model building pass. For parsimony, I only consider plain Boosting as the counterexample, the principle clearly applying to MARS and similar greedily optimized additive models. Consider building a small tree: after one split, the prediction function is

$$\hat{y}_i = \beta_1 I(x_i > 0) + \beta_2 I(x_i \leq 0).$$

Further splitting within each of the newly created subsamples, the additive model becomes

$$\hat{y}_i = \beta_1 I(x_i > 0) \left[\alpha_1 I(z_i > 0) + \alpha_2 I(z_i \leq 0)\right] + \beta_2 I(x_i \leq 0) \left[(\gamma_1 I(w_i > 0) + \gamma_2 I(w_i \leq 0)\right].$$

Finally, define $d^+_{x,i} = I(x_i > 0)$ as a regressor and the rest accordingly. The polynomial-looking model is

$$\begin{aligned}\hat{y}_i &= \beta_1 d^+_{x,i} \left[\alpha_1 d^+_{z,i} + \alpha_2 d^-_{z,i}\right] + \beta_2 d^-_{x,i} \left[(\gamma_1 d^+_{w,i} + \gamma_2 d^-_{w,i}\right] \\ &= \theta_1 d^+_{x,i} d^+_{z,i} + \theta_2 d^+_{x,i} d^-_{z,i} + \theta_3 d^-_{x,i} d^+_{w,i} + \theta_4 d^-_{x,i} d^-_{w,i}.\end{aligned}$$

This representation is helpful to allow a look at trees in the same way one could look at boosting. The goal is to understand why randomization is more easily obtained with a tree than with an additive scheme like boosting. Trees are indeed very singular additive models. In fact, they are partly multiplicative. It is clear from the above that $d_{x,i}$ better be a good choice, because it is not going away: any term in the model building pass will be multiplied by it. By construction, no term added later in the expansion has the power to entirely undo the damage of a potentially harmful first split. In other words, splitting the sample (here according to the dummies $d^+_{x,i}$ and $d^-_{x,i}$) is an *irreversible* action. Of course, observations misclassified by $d_{x,i}$ will eventually get some relief by additional splits within that assigned subsample. Nonetheless, nothing comes close to bring them back to the other side of $d_{x,i}$ if it is realized ex post that this would have been optimal.

To contrast with the above, let us look at a toy boosting model where the base learners are single-split trees. The first step is

$$\hat{y}_i = \nu \left[\beta_1 d^+_{x,i} + \beta_2 d^-_{x,i}\right]$$

and ν is the usual shrinkage parameter. The fitting recursion consists of taking the in-sample prediction error at step s and consider it as the new target in step $s+1$. An important difference with respect to the plain tree fitting process is that step s leading to

$$\hat{y}_i = \nu \left[\beta_1 d^+_{x,i} + \beta_2 d^-_{x,i}\right] + \cdots - \nu \left[\beta_1 d^+_{x,i} + \beta_2 d^-_{x,i}\right] \tag{3.2}$$

is absolutely possible. In words, by additivity, it is possible to correct any step that eventually turned out to be suboptimal in the search for a close-to global optima. With the randomization induced in Stochastic Gradient Boosting and other practical aspects, this is unlikely to happen exactly in those terms.[24] However, a small ν and a large number of steps/trees in the additive model will mechanically increase the algorithm's potential for "reversibility". Indeed, Friedman et al. (2001) detail an equivalence between a procedure similar to the above and LASSO. If $\nu \to 0$, # of steps $\to \infty$ and regressors are uncorrelated, they obtain the LASSO solution – a *global* solution. Thus, there is an evident tension between how close to a global optimization a boosting-like algorithm can get and its capacity to generate inner randomization sufficiently to be dispensed from tuning the stopping point. This is intuitive. There is only one truth to be learned, and learning it slowly will safely get you there. Alternatively, by learning fast and imprecisely, you get it right on average over many tryouts.

On that spectrum, MARS is positioned between trees and Boosting. It is close to the example above, with $\nu = 1$ and using different base learners. While it is not common to refer to a reduced number of steps in Boosting as "pruning", it is the traditional language used to describe dropping off terms after MARS' forward pass. Since MARS does not use ν to slow the learning process, it can quickly overfit with much fewer additive terms than an algorithm using a small ν. Hence, the phenomena described in (3.2) is believed to be an oddity unlikely to happen in MARS, making the model a middle ground of sorts when it comes to satisfying Point 2.

Given the above arguments, can pruning the underlying base learners still help? The answer is yes. As Uhlig (2017) puts it (for a very different scientific question): if you know it, impose it; if you do not: do not impose it. If both the number of observations and regressors is small, respectively limiting the randomizing power of `B` & `P`), there likely will be gains from regularizing base learners directly. As mentioned earlier, perfect regularization is obtained under perfect randomization. If it is obvious that in a specific application, we are very far from that idyllic case, then pruning (gently) the base learners is reasonable to consider.

[24] Stochastic Gradient Boosting (Friedman, 2002) randomly selects a subset of observations at each step to train the weak learner.

An interesting question is whether the properties detailed here apply to LASSO, which would free the world from ever tuning λ again. Indeed, when implemented via Least Angle Regression (Efron et al., 2004), the algorithm very much looks like a forward stage-wise regression. In the spirit of the above, one would hope to let a randomized version of the regularization path roll until $\lambda = 0$, average those solutions and obtain the same R^2_{test} as if λ had been carefully tuned. Unfortunately, LASSO violates two of the requirements listed before. First, parameters are re-evaluated along the regularization path. For λ's that lay in the overfitting territory, the estimated coefficients will be weakened since they are re-estimated in an overcrowded model. Second, letting the model overfit (when $p < N$) implies setting $\lambda = 0$ which returns the OLS solution for any iteration, making the desired level of randomization likely unattainable.[25]

3.2.3. Why RF is Not Equivalent to Pruning a Single Tree

Bagging and perturbing the model as implemented by RF leads to two enviable outcomes. The first is that the randomization procedure implicitly prunes an overfitting ensemble when applied to new data. This was the subject of previous subsections. The second, more standard, is that as a result of randomization, RF performs orders of magnitude better than a single pruned true (Breiman, 1996). This is also observed in the simulations from section 3.3: `B` & `P` CART does much better than the ex-post optimally pruned base learner. In contrast,`B` & `P` MARS and Boosting will provide similar performance to that of their respective base learner stopped at s^*. Thus, RF must be pruning something else. I complete the argument of previous sections by arguing that its "pruning via inner randomization" is applied on the true *latent* tree $\mathcal{T}$ in

$$y_i = \mathcal{T}(X_i) + \epsilon_i \tag{3.3}$$

which itself can only be constructed from randomization. In short, it is the recursive fitting procedure itself that generates the need for Bagging.[26]

The inspiration for the following argument comes from forecasting with non-linear time series models, in particular with the so-called Self-Exciting Threshold Autoregression (SETAR). A simple illustrative SETAR DGP is

$$y_{t+1} = \eta_t \phi_1 y_t + (1 - \eta_t)\phi_2 y_t + \epsilon_t, \quad \eta_t = I(y_t > 0) \tag{3.4}$$

[25]This last point could be alleviated, when in the $p > N$ case, the LASSO solution can include at most N predictors. In that scenario, the included set of variables would depend on the order within the regularization path (rather than its termination) which would increase randomization. Nevertheless, we cannot expect LASSO to benefit from automatic tuning because linear regression coefficients are re-evaluated along the estimation path.

[26]In Appendix 3.6.3, I review a more standard case for Bagging based on presumed heteroscedascity.

where ϵ_t is normally distributed. The forecasting problem consists in predicting y_{t+h} for $h = 1, ..., H$ given information at time t. As it is clear from (3.4), y_{t+1} is needed to obtain the *predictive function* for y_{t+2} which is either ϕ_1 or ϕ_2. Alas, only an estimate $\hat{y}_{t+1} = E(y_{t+1}|y_t)$ is available. By construction, $E(\hat{y}_{t+1}) = y_{t+1}$. However, by properties of expectations, $E(f(\hat{y}_{t+1})) \neq f(y_{t+1})$ if f is non-linear. Hence, proceeding to iterate forward using $\hat{y}_{t+h}$'s as substitutes for y_{t+h} at every step leads to a bias problem that only gets worse with the forecast horizon. If such an analogy were to be true for trees, this would mean that as the tree increase in depth, the more certain we can be that we are far from $\mathcal{T}(X_i)$, the optimal prediction function. I argue that it is the case.

Following the time series analogy, the prediction for a particular i can be obtained by a series of recursions. Define the cutting operator

$$\mathcal{C}(\boldsymbol{S}; y, X, i) \equiv \mathcal{S}_i \left(\underset{k \in \mathcal{K}, c \in \mathbb{R}}{\arg\min} \left[\min_{\mu_1} \sum_{i \in \{\boldsymbol{S} | X_k \leq c\}} (y_i - \mu_1)^2 + \min_{\mu_2} \sum_{i \in \{\boldsymbol{S} | X_k > c\}} (y_i - \mu_2)^2 \right] \right) \quad (3.5)$$

where $\mathcal{S}_i$ extract the subset that includes i out of the two produced by the splitting step. Inside the $\mathcal{S}_i$ operator is the traditional one-step tree problem. $\mathcal{K}$ is the set of potential features to operate the split at an optimized value c. $\boldsymbol{S}$ is the sample to split and is itself the result of previous cutting operations from steps $s-1$, $s-2$ and so on. To get the next finer subset that includes i, the operator is applied to the latest available subset: $\boldsymbol{S}' = \mathcal{C}(\boldsymbol{S}; y, X, i)$. The prediction for i can be obtained by using $\mathcal{C}$ recursively starting from $\boldsymbol{S}_0$ (the full data set) and taking the mean in the final $\boldsymbol{S}$ chosen by some stopping rule. In other words, the true tree prediction in (3.3) is $\mathcal{T}(X_i) = E\left(y_{i'} | i' \in \mathcal{C}^D(\boldsymbol{S}_0; y, X, i)\right)$ where D is the number of times the cutting operator must be applied to obtain the final subset in which i resides. To obtain the true tree prediction – the mean of observations in i's "true" terminal node – the sequence of $\mathcal{C}$'s must be perfect. Hence, consistency remains on safe ground: as the sample size grows large, estimation error vanishes and $\hat{\boldsymbol{S}} \to \boldsymbol{S}$ at each step. The finite sample story is, however, quite different.

Using $\hat{y}_{t+1}$ *in situ* of y_{t+1} in SETAR and $\hat{\boldsymbol{S}}$ *in situ* of $\boldsymbol{S}$ in a tree generate problems of the same nature. At each step, the expected composition of $\hat{\boldsymbol{S}}$ is indeed $\boldsymbol{S}$.[27] However, just like the recursive forecasting problem, the expected *terminal* subset is defined as an expectation over a recursion of nonlinear operators. Using $\hat{\boldsymbol{S}}$ rather than the unobserved $\boldsymbol{S}$ at each step does *not* deliver the desired expectation. Intuitively, getting the right k and c out of many possible combinations is unlikely. These small errors are reflected in $\hat{\boldsymbol{S}} \neq \boldsymbol{S}$ which

[27] This notion can be formalized by defining the expectation in terms of indicator functions for each candidate observation. Each observation at each cutting step is expected to be classified in the right one of two groups.

is taken *as given* by the next step. Those errors eventually trickle down with absolutely no guarantee that they average out. In short, the direct CART procedure produces an unreliable estimate of a greedily constructed predictor $\mathcal{T}(X_i)$ because it takes as given at each step something that is not given, but estimated. Since $\mathcal{C}$ is a non-linear operator, this implies that the mean itself is not exempted from bias.

If the direct procedure cannot procure the right expected subset on which to take the average and predict, what will? The intuition for the answer, again, stems from the SETAR example. The proposed solution in the literature is – with a distinctively familiar sound – using bootstrap to simulate the intractable expectation (Clements and Smith (1997)).[28] $\hat{y}_{t+1}$ is augmented with a randomly drawn shock (from a parametric distribution or from those in the sample) and a forecast of y_{t+2} is computed conditional on it. Then, the procedure is repeated for B different shocks and the final forecast is the average of all predictions, which, by the non-linearity of $f()$, can make it a very different quantity from $f(\hat{y}_{t+1})$. A forecaster will naturally be interested in more than y_{t+2}. This procedure can be adapted by replacing the draw of a single shock by a series of them that will be used as the model is simulated forward. The prediction at step H is an average of forecasts at the end of each b randomly generated sequence.

Analogously, a natural approach is to simulate the distribution of $\boldsymbol{S}$ entering a next splitting step is to bootstrap the sample of the previous step, run $\mathcal{C}$ a total of B times, apply $\mathcal{C}$ in the next step and finally take the average of these B bootstrapped trees predictions. For a deeper tree, the growing process continues on the bootstrapped sample and the average is taken once the terminal condition is reached.

Coming to the original question: if RF is pruning something, what is it? I conjecture it is pruning $\mathcal{T}$ in (3.3). Unlike the implicit early-stopping property explained in section 3.2.1, this statement cannot be supported or refuted by the simulations presented in section 3.3. However, in Goulet Coulombe (2021), it is shown that under a "true tree" DGP, the performance of RF and a version of CART with a low learning rate coincides. The latter can be linked to fitting the true tree optimally via an (extremely) high-dimensional LASSO problem.

3.3. Simulations

Simulations are carried to display quantitatively the insights presented in the previous section. Namely, I want to display that (i) ideal population sampling of greedy algorithms performs pruning/early stopping, (ii) RF very closely approximates it for trees and (iii) the property also extends to altered versions of Boosting and MARS.

[28] For a discussion of the SETAR case and other non-linear time series models, see section 2.7 in Khan (2015).

3.3.1. Setup

I consider 3 versions of 3 algorithms on 5 DGPs. The 3 models are a single regression tree (CART), Stochastic Gradient Boosting (with tree base learners) and MARS. The five DGPs are a Tree[29], Friedman 1, 2 and 3 (Friedman (1991)) as well as a linear model[30]. The first two versions of each model are rather obvious. First, I include the plain model and second, a bootstrapped and perturbed ensemble of it, as described earlier. Additionally, `B` & `P` versions of MARS and Boosting have the so-called data augmentation (`DA`) option activated. It consists in enlarging the feature matrix to additionally incorporate $\tilde{X} = X + \mathcal{E}$ where $\mathcal{E}$ is a matrix of Gaussian noise.[31] Overall, `DA` can improve perturbation's potential when regressors are scarce.[32] The Boosting and MARS `B` & `P` + `DA` versions will be referred to by the less gloomy-sounding sobriquets *Booging* and *MARSquake*. The `R` package *bagofprunes* implements both. Execution details are available in Appendix 3.6.2.

The third version of each model, "Population Sampling" aims at displaying what results look like under the ideal case of perfect randomization. In this third version, subsampling is replaced by sampling B non-overlapping subsets of N observation from a population of $B \times N$ observations. This third version has the benefit of making clear which algorithm generates enough inner randomization to get close to that desirable upper bound. For all simulations, $N = 400$ and the test set also has 400 observations.

In terms of standard hyperparameters, Boosting has the shrinkage parameter $\nu = 0.1$, the fraction of randomly selected observations to build trees at each step is 0.5, and the interaction depth of those trees is 3. Of course, while those are fixed for all simulations, we will want to tune them once we get to real data. However, here, the point is rather to study the hold-out sample performance of each model as its depth increase, and compare that across the 3 versions. MARS has the polynomial degree set to 3.[33] RF is used with a rather high `mtry` of $9/10$ so to be better visually in sync with plain CART at a given depth.[34] The subsampling rate is $2/3$ for all bagged models.

3.3.2. Results

Figures 37 and 39 report the median R^2 between hold-out sample predictions and the true conditional mean for 30 simulations. Columns are DGPs and rows are models. The x-axis is an increasing index of complexity/depth for each greedy model. Overfitting should

[29] The true tree DGP is generated using a CART algorithm's prediction function as a "new" conditional mean function from which to simulate. The "true" minimal node size being used is 40 (10% of the training set).

[30] The linear DGP is the sum of five mutually orthogonal and normally distributed regressors.

[31] For categorical variables, $\tilde{X}$ is obtained by duplicating X and shuffling a fraction of its rows.

[32] This is the case for the considered Friedman's DGPs which have 5 useful regressors and 5 useless ones.

[33] For those unfamiliar with this machinery, see Friedman (2002) for Boosting and Friedman (1991) or Milborrow (2018) for MARS.

[34] For completeness, results when using `mtry`$= 1/2$ for both plain CART and RF are reported in Figure 40. It is clear that in the high signal-to-noise ratio environment, the milder perturbation of `mtry`$=9/10$ is preferable.

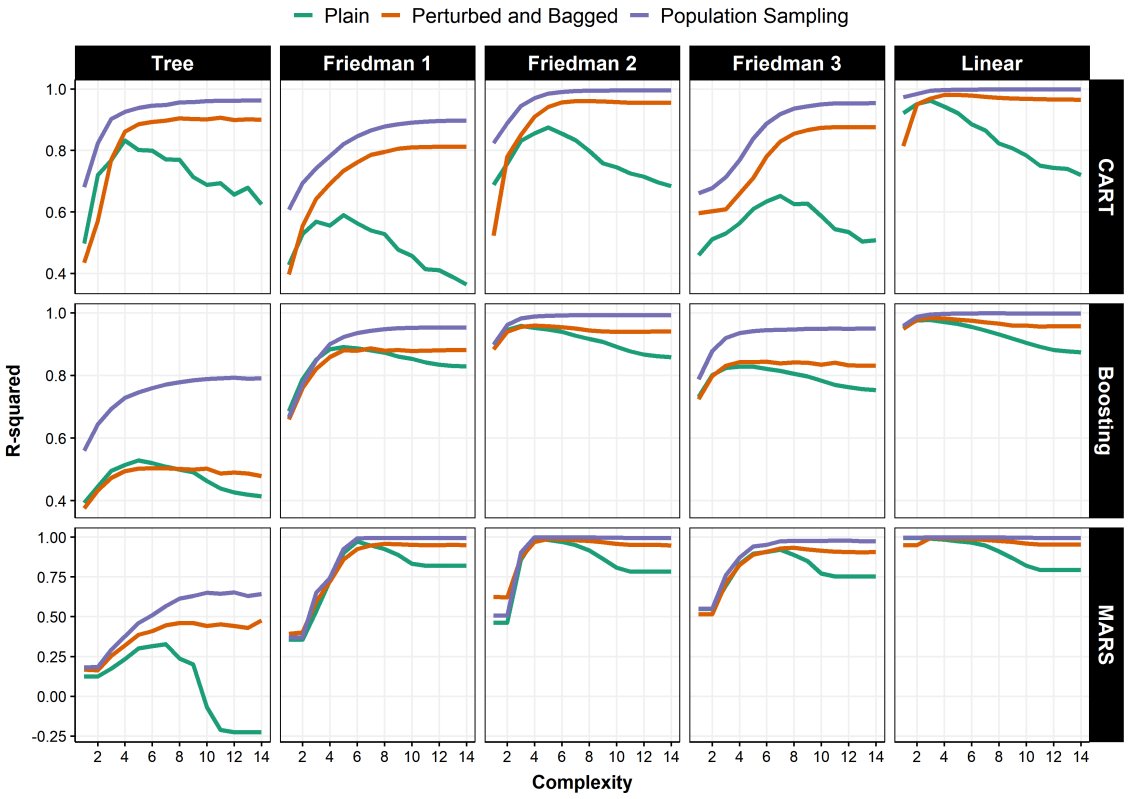


Figure 37: This plots the hold-out sample R^2 between the prediction **and the true conditional mean**. The level of noise is calibrated so the signal-to-noise ratio is 4. Column facets are DGPs and row facets are base learners. The x-axis is an index of depth of the greedy model. For CART, it is a decreasing minimal size node $\in 1.4^{\{16,..,2\}}$, for Boosting, an increasing number of steps $\in 1.5^{\{4,..,18\}}$ and for MARS, it is an increasing number of included terms $\in 1.4^{\{2,..,16\}}$.

manifest itself by a decreasing R^2 past a certain depth. I consider two levels of noise, one that corresponds to a signal-to-noise ratio of 4 (Figure 37) and one of 1 (Figure 39).

What does section 3.2 imply for the curves in Figure 37? First, the population sampling versions (purple line) should be weakly increasing since they perform implicit "perfect" early stopping. Second, the `B` & `P` versions (orange), should be parallel to those provided the underlying greedy model is generating enough inner randomization. Third, the value of the orange line at the point of maximal depth should be as high or higher than the maximal value of the green curve (i.e, the plain version's ex-post optimal stopping point).

When it comes to CART, those three properties are verified exactly. For any DGP, and both the population sampling and the `B` & `P` versions, increasing the complexity of the model by shrinking the minimal node size does not lead to a performance metric that eventually decrease. The striking parallelism of the purple and orange lines is due to trees generating enough inner randomization with `B` & `P` so it performs self-pruning at a level

comparable to that of the ideal experiment.[35] Depending on the DGP, the plain version generally follows the `B & P` one for some time before detaching itself from it past its ex-post optimal point of early stopping — in like with the idea that `B & P` CART (aka RF) performs implicit pruning.

Looking at Boosting and MARS, we again see that the population sampling line is weakly increasing in the respective depth of both models. If the `B & P` version fails to match this ideal shape, it is because the current specification cannot generate enough inner randomization. Figure 37 shows unequivocal encouraging results for both Boosting and MARS. For all DGPs, a clear pattern is observed: the `B & P` version's performance increases until it approximately reaches the optimal point (as can be ex-post determined by the hump in the green line) and then *remains at that level*, even if the base learners (one example being the 'Plain' version) are clearly suffering from overfitting. Under those conditions, it is fair to say that the enviable RF property is transferable to Boosting, and in a more pronounced fashion, MARS. When the noise level increase as depicted in Figure 39, we observed the same – albeit marginally less successful – phenomenon. Indeed, in those harder conditions, there is a small gap between ideal randomization and the one generated by MARSquake. However, the decrease in performance following the optimal depth is orders of magnitude smaller than what is observed for the plain version.

As discussed earlier, while it is eligible for the self-pruning property, Boosting is expected to be overall more recalcitrant than MARS, especially if a small ν is used. This is observed in a mild form for the higher level of noise in Figure 39. Nevertheless, the decline in R^2 between the hold-out sample prediction and the true conditional mean is, again, much smaller than what the plain version yields. Additionally, for all DGPs under a signal-to-noise ratio of 4, we see Booging obeying a pattern close to the ideal population sampling version. As depth increase, it yields a R^2 that remains at the optimal non-overfitting level, all the while its respective plain versions is slowly (but clearly) suffering from overfitting. Finally, it is observed for all DGPs and base learners that a higher signal-to-noise ratio will help in reducing the gap between post-s^* *slopes* of the orange and purple lines. In sum, with DGPs ranging from the rather complex true tree[36] to a simple linear models, these simulations demonstrate the main insights advanced in the previous sections. They provide reasons to believe that Booging and MARSquake could (at least) seldomly yield performance improvements on standard data sets – without CV.

[35] The purple line is mechanically expected to be at least above the orange one for a fixed depth: the former uses more data points which also helps at reducing estimation error.

[36] Overall performance is lower for this harder DGP except, obviously, for RF.

3.4. Empirics

Applying `B` & `P` to Boosting and MARS is nothing new in itself. For instance, Rasmussen (1997) report that Bagged MARS supplant MARS on many data sets. This section has a slightly more subtle aim than crowning the winner of a models' horse race. Rather than focusing on improving the tuned/pruned model which is already believed to be optimal, *Booging* and *MARSquake* bag and perturb completely overfitting based learners, which, as we will see, perform very poorly by themselves. Their performance will be compared to versions of Boosting and MARS where the optimal stopping point has been tuned by CV. The goal is to verify that in many instances, Booging and MARSquake provide a similar predictive power to that of tuned models. Since CV's circumstantial imperfections are vastly documented (Krstajic et al., 2014; Bergmeir et al., 2018), it is not unrealistic to expect the `B` & `P` versions to sometimes outperform their tuned counterparts. In sum, this small application shows that the equivalence championed in previous chapters holds with real data and thus provides data scientists with a fruitful alternative to consider when building models.

3.4.1. Setup

Most data sets are standard in the literature (mostly from UCI repository) with a few additions which are thought to be of particular interest here. For instance, many of the standard regression data sets have a limited number of features with respect to the number of observations. A less standard inclusion like *NBA Salary* has 483 observations and 26 features. *Crime Florida* pushes it much further with a total of 98 features and 90 observations. Those data sets are interesting because avoiding CV could generate larger payoffs in higher-dimensional setups. Further information on data sets is gathered in Table 21.

Still in the high-dimensional realm, but with the additional complication of non-*iid* data, are the 6 US macroeconomic data sets based on McCracken and Ng (2020).[37] There are two obvious potential benefits from self-tuning models in a macroeconomic forecasting environment. First, traditional CV is known to be overoptimistic in a time series context and avoiding it could generate forecasting gains (Bergmeir et al., 2018). Second, forecasting "horse races" are usually conducted in a recursive fashion which mimics the reality of economic forecasting in quasi-real-time. That is, as new observations are available, the model must be constantly re-estimated (or at least, often) along with the optimization of its hyperparameters. Avoiding the latter implies substantial decrease in computational burden,

[37] Bagging has received attention of its own in the macroeconomic forecasting literature (Inoue and Kilian, 2008; Hillebrand and Medeiros, 2010; Hillebrand et al., 2020). However, nearly all studies consider the more common problem of variable selection via hard-thresholding rules – like t-tests (Lee et al., 2020). Those strategies are akin to what discussed in section 3.2.1, and cannot (and do not) strive for automatic pruning. Nevertheless, the motivation for using Bagging in their context is very close to what described for trees in section 3.2.3.

which can sometimes be a matter of days. The 3 macroeconomic variables are quarterly GDP growth, unemployment change and inflation. I consider predicting those variables at an horizon of 1 quarter ($h = 1$) and 2 quarters ($h = 2$). The X matrix is based on Goulet Coulombe (2020b)'s recommendations for ML algorithms when applied to macro data, which is itself a twist (for statistical efficiency and lessen computational demand) on well-accepted time series transformations (to achieve stationarity) as detailed in McCracken and Ng (2020).[38] Each data set has 212 observations and around 600 predictors.[39]

Beyond Boosting, MARS, and their different variants under scrutiny for this exercise, I include a few benchmark models. Those include LASSO, RF with default tuning parameters (`mtry=1/3`), a cost-complexity pruned regression tree, and two different neural networks. The first NNs is shallow (2 layers of 32 and 16 neurons) and is inspired from Gu et al. (2020). Such an architecture has provided reasonable performance on Canadian (Goulet Coulombe et al., 2020b) and UK macroeconomic data Goulet Coulombe et al. (2021). The second is a deep NN (DNN, with 10 layers of 100 neurons) following the recommendations of Olson and Wyner (2018) for small data sets. Additional NNs details (like their tuning) are in Appendix 3.6.4. For macro data sets, the benchmarks additionally include an autoregressive model of order 2 (AR) and a factor-augmented regression with 2 lags (FA-AR) which are widely known to be hard to beat (Kotchoni et al., 2019).

For all data sets, I keep 70% of observations for training (and optimizing hyperparameters if needed) and the remaining 30% to evaluate performance. For cross-sectional data sets, those observations are chosen randomly. For time series applications, I keep the observations that consist of the first 70% in the sample as the training set. The test set starts before the 2001 recession and ends in 2014, which conveniently includes two recessions. Lastly, a seldomly binding outlier filter is implemented. Every prediction that is larger than twice the maximal absolute difference (in the training sample) with respect to the mean is replaced by the RF prediction (which is immune to outliers since it cannot extrapolate). This last addition is particularly helpful to prevent wildly negative R^2_{test} for non-tuned plain MARS and (less frequently) Boosting.

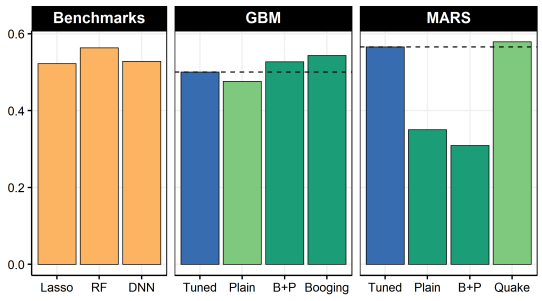
3.4.2. Results

All prediction results are reported in Table 22 and an illustrative subset of those is included in Figure 38.[40] Moreover, to empirically document the R^2_{test} and R^2_{train} gap, Table 23 reports R^2_{train}'s. On the *Abalone* data set, non-tuned MARS is overfitting, which leads to subpar

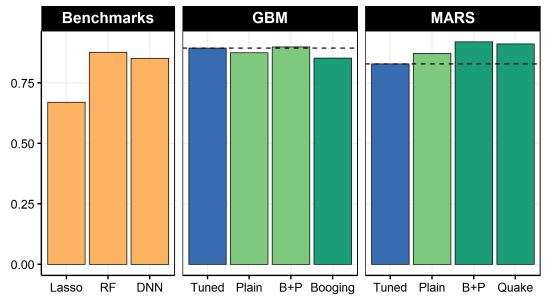
[38] Goulet Coulombe et al. (2019) further study optimal data transformations for machine macroeconomic forecasting for many series and algorithms.

[39] The number of features varies across macro data sets because a mild screening rule was implemented ex-ante, the latter helping to decrease computing time.

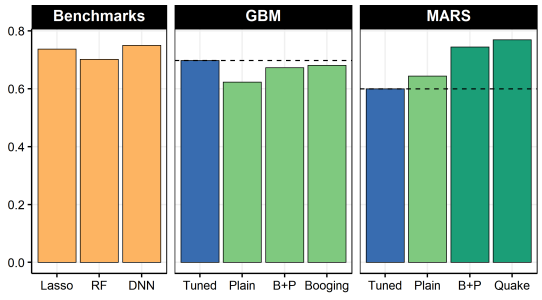
[40] NN and Tree are left out of Figure 38 to enhance readability since neither of them were ever the best model for a given data set, execpt for NN beating `B` & `P` MARS by 1% on *Crime Florida*.



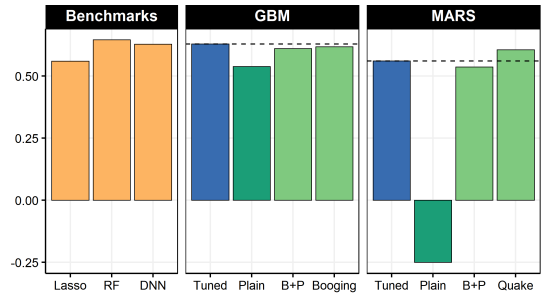
(a) *Abalone*



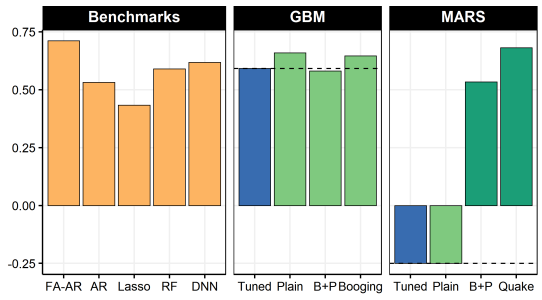
(b) *Boston Housing*



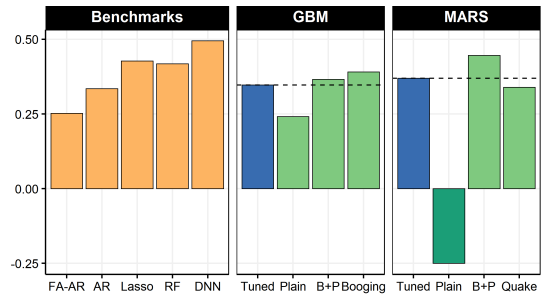
(c) *Crime Florida*



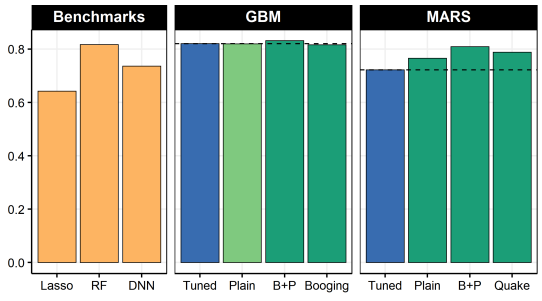
(d) *Fish Toxicity*



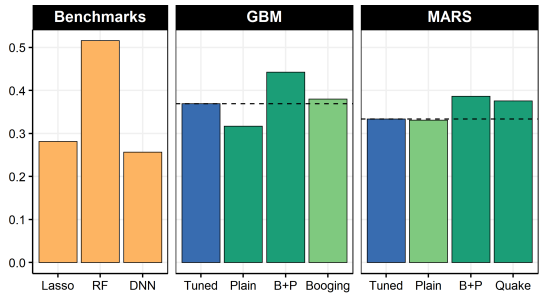
(e) *US Unemployment Rate* $(h = 1)$



(f) *US Inflation* $(h = 1)$



(g) *California Housing*



(h) *White Wine*

Figure 38: A Subset of Empirical Prediction Results. Performance metric R^2_{test}. Darker green bars means the performance differential between the tuned version and the three others is statistically significant at the 5% level using t-tests (and Diebold and Mariano (2002) tests for time series data). Light green means the difference is not significant at the prescribed level. To enhance visibility in certain cases, R^2_{test}'s below -0.25 are constrained to 0.25.

performance. In line with simulation results, Booging and MARSquake perform almost indistinguishably as well as using a single base learner and tuning it. Even better, the newly proposed overfitting Boosting ensembles deliver statistically significant *gains* at the 5% confidence level. As RF, those two ensembles have a very high R^2_{train} (see Table 23) and yet, stellar performance is reported on the test set. However, it is noted that `B` & `P` MARS does not generate enough inner randomization to match the tuned model. This wedge between `B` & `P` MARS and MARSquake suggests an important role for data augmentation when features are scarce (7 in the case of *Abalone*).

For *Boston Housing*, all models behave similarly (including the plain versions) which is attributable to the true R^2 (very) likely being quite high for that particular data set. Thus, there is little room for overfitting in the first place. Yet, the ensembles R^2_{train}'s are all in the vicinity of 1, which is markedly higher than 0.85 as obtained for most models on the test set. Clearly, implicit pruning via randomization acted as a potent regularizer.

Crime Florida – the very high-dimensional case which is not time series – brings new insights to the table. While `B` & `P` Boosting and Booging are doing marginally better than the tuned version, the two ensembles of completely overfitting MARS (their R^2_{train} are respectively 0.97 and 0.98) are doing much better than the tuned version. They both deliver a surprisingly high R^2_{test} of almost 0.8 in the case of MARSquake, an improvement which is statistically significant for the `B` & `P` version. The latter is also the overall second best model (being 1% less than NN) for this data set.

Fish Toxicity, *Red Wine*, *White Wine* are data sets with a more common ratio of predictors to observations. In both cases, the plain overfitting versions are significantly worse than the tuned versions. The ensembles deliver a performance (with respect to tuned counterparts) that is either significantly better or not statistically distinguishable. *California Housing* is an example of data set with an enviable number of observations (more than 20,000). It is observed that all ensembles do significantly better than the tuned versions for MARS. All performances are nearly identical for Boosting. Furthermore, for both *California Housing* and *Bike Sharing*, which are the sole data sets with more than 15,000 observations, the tuned version of MARS is worse than RF or any version of Boosting. In contrast, the overfitting MARS ensembles perform similarly well.

As one should expect, results are more disparate when looking at the macroeconomic data sets. The economic forecasting problem is (i) high-dimensional (ii) incorporates a strong unpredictable component (economic shocks) and (iii) the target variable surely contains a sizable measurement error. To make things even more gruesome, the true model – if there is any – may be constantly evolving. Thus, those data sets are difficult laboratories which

differ in many aspects to those considered up to now. Consequently, it is recomforting that a (now) very familiar pattern is also visible for both unemployment and (to some extent) GDP at $h = 1$. Booging does as well as the tuned Boosting. Moreover, the former provides the best outcome among all models, with a 11% R^2_{test} increase with respect to both economic forecasting workhorses (AR, FA-AR). When it comes to plain and tuned MARS, all models are somewhat worse than the benchmarks with the tuned model itself delivering a terrible R^2_{test}. MARSquake is partially exempted from this failure for GDP, and completely is for unemployment. In the latter case, MARSquake is as good as FA-AR which incredible resilience is vastly documented (Kotchoni et al., 2019; Goulet Coulombe et al., 2019, 2020a). For inflation ($h = 1$), the best models are clearly `B` & `P` MARS and DNN. Generally, it is expected that the `DA` option has less bite for those data sets since the raw data is already extremely "wide" and has a strong factor structure (McCracken and Ng, 2020).

In terms of overall performance, a quick look at Table 22 reveals how well ensembles of overfitting base learners do. One version or another (including RF) is the best model for 11 out of 20 data sets, and the 9 other cases are often 1% (or less) away from being ties. The leading models in that regard are RF and MARS-based ensembles, with 5 wins and 4 wins, respectively. Zooming on macroeconomic data sets, Booging dominates its tuned counterpart for all 6 data sets. For MARS, when R^2_{test}'s are positive, MARSquake subdues tuned MARS 3 times out of 4. Thus, overfitting ensembles of any kind work well for macro forecasting where the use of CV is seldomly hazardous.

For the vast majority of cases, we observe that $R^2_{\text{test}} < R^2_{\text{train}}$ by a wide margin because the latter is excessively high. For instance, R^2_{train} are almost all above 0.95 for Booging and RF, and MARSquake as well when X is large. In Table 23, plain MARS' R^2_{train} can sometimes be "deceivingly" far from 1, something that never happens for Boosting and RF. This is due to MARS being occasionally recalcitrant to continue adding redundant and/or useless terms (see Appendix 3.6.2 for details). Nonetheless $R^2_{\text{test}} < R^2_{\text{train}}$ is clearly maintained. This rarity can be thought of as "earlier" stopping, a reasonable form of very mild regularization applied on base learners.

Lastly, a comment on overall NN and Deep NN performances. DNN's performance tends to be more stable than that of NN, especially for macroeconomic targets — against the traditional wisdom that tighter architectures are more appropriate for the noisy macro data environment (Goulet Coulombe et al., 2019). The computationally demanding DNN is usually dominated by RF and other ensembles, with the noticeable exception of inflation where it narrowly beats `B` & `P` MARS by 4% at $h = 1$ with a R^2_{test} of 0.49, and distance the competition even further at $h = 2$ with a R^2_{test} of 0.51. In line with Olson et al. (2018)'s

results, this suggests econometricians should not refrain from using deep architecture in future research, even when faced with small sample sizes. In the meantime, RF, Booging and MARSquake remain a trio that is very hard to beat.

Giving the ongoing discussion on the non-overfitting properties of DNN's in the ML literature, it is interesting to investigate through Tables 22 and 23 how their R^2_{test} and R^2_{train} compares, and check if DNN inherit similar properties to RF, as put forward in Belkin et al. (2019a) and others. The short answer is "much less". RF's R^2_{train} is *almost always* above 0.9, whereas that of DNN fluctuates highly depending on the target, being roughly evenly distributed between 0.5 to 0.9. Even though DNN's gap between R^2_{train} and R^2_{test} is high for macro data sets (and yet DNN delivers solid performances sporadically), we remain far from what could be referred to as the "interpolating" regime.

All in all, empirical results confirm the insights developed in section 3.2. In almost every instance, the overfitting ensembles do at least as well as the tuned version while completely overfitting the training sample, the same way RF would. Sometimes they do much better. Thus, they are alternatives to their cross-validated counterparts. In sum, it seems that, mixed with a proper amount of randomization, *greed is good*.

3.5. Conclusion

A fundamental problem is to detect at which point a learner stops learning and starts imitating. In ML, the common tool to prevent an algorithm from damaging its hold-out sample performance by overfitting is the intuitive solution of cross-validation. It is widespread knowledge that performing CV on Random Forests rarely yields dramatic improvements. Concurrently, it is often observed that $R^2_{\text{test}} < R^2_{\text{train}}$ without R^2_{test} being any less competitive. In this chapter, I argued that proper inner randomization as generated by Bagging and perturbing the model, when combined with a greedy fitting procedure, will implicitly prune the learner once it starts fitting noise. By the virtues of recursive model building, the earlier fitting steps are immune to the instability brought upon by ulterior (and potentially harmful) steps. Once upon a time, the author heard a very senior data scientist and researcher say in a seminar, 'If you put a gun to my head and say "predict", I use Random Forest.' This chapter rationalizes this feeling of security by noting that unlike any standard ML algorithms out there, RF performs its own pruning without the perils of cross-validation.

3.6. Appendix

3.6.1. *Additional Graphs and Tables*

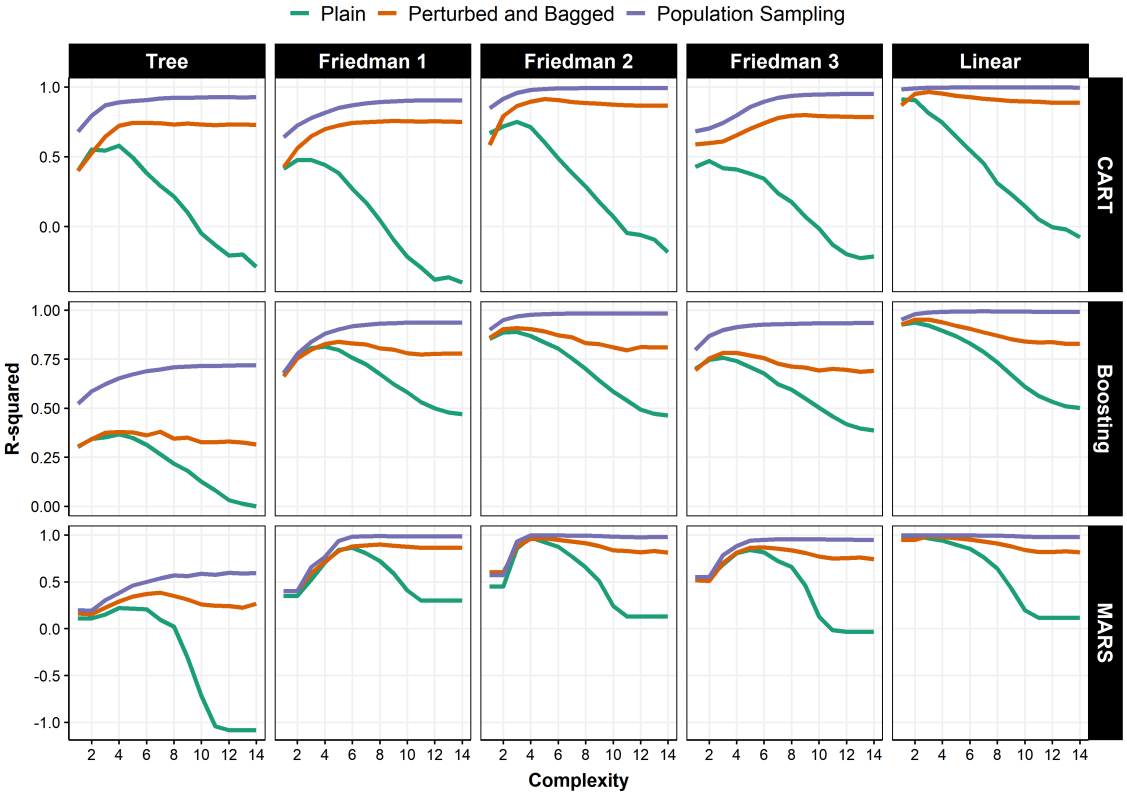


Figure 39: This plots the hold-out sample R^2 between the prediction and the true conditional mean. The level of noise is calibrated so the signal-to-noise ratio is 1. Column facets are DGPs and row facets are base learners. The x-axis is an index of depth of the greedy model. For CART, it is a decreasing minimal size node $\in 1.4^{\{16,..,2\}}$, for Boosting, an increasing number of steps $\in 1.5^{\{4,..,18\}}$ and for MARS, it is an increasing number of included terms $\in 1.4^{\{2,..,16\}}$.

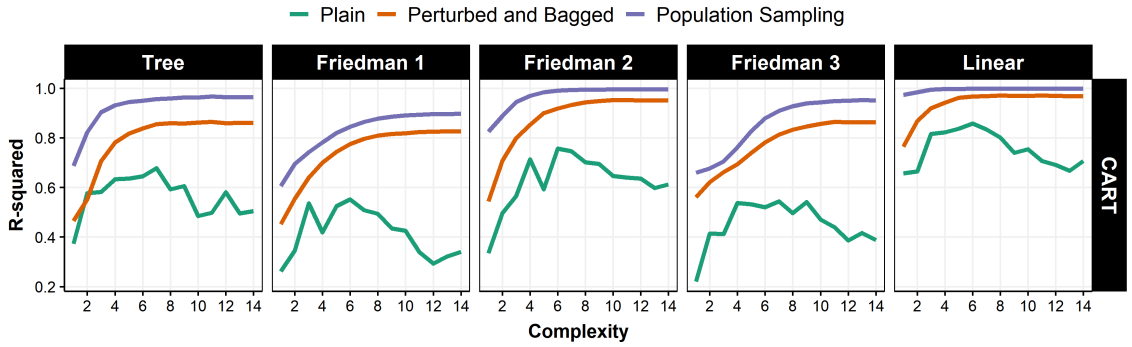


Figure 40: This is Figure 37's first row with `mtry`= 0.5.

Table 21: 20 Data Sets

Abbreviation	**Observations**	**Features**	**Data Source**
Abalone	4,177	7	archive.ics.uci.edu
Boston Housing	506	13	lib.stat.cmu.edu
Auto	392	7	archive.ics.uci.edu
Bike Sharing	17,379	13	archive.ics.uci.edu
White Wine	4,898	10	archive.ics.uci.edu
Red Wine	1,599	10	archive.ics.uci.edu
Concrete	1,030	8	archive.ics.uci.edu
Fish Toxicity	908	6	archive.ics.uci.edu
Forest Fire	517	12	archive.ics.uci.edu
NBA Salary	483	25	kaggle.com
CA Housing	20,428	9	kaggle.com
Crime Florida	90	97	census.gov
Friedman 1 R^2=.7	1,000	10	cran.r-project.org
Friedman 1 R^2=.4	1,000	10	cran.r-project.org
GDP h=1	212	599	Google Drive
GDP h=2	212	563	Google Drive
UNRATE h=1	212	619	Google Drive
UNRATE h=2	212	627	Google Drive
INF h=1	212	619	Google Drive
INF h=2	212	611	Google Drive

Notes: The number of features includes categorical variables expanded as multiple dummies and will thus be sometimes higher than what reported at data source website. Data source URLs are visibly abbreviated but lead directly to the exact data set or package being used. The number of features varies for each macro data set because a mild screening rule was implemented ex-ante, the latter helping to decrease computing time.

Table 22: R^2_{test} for all data sets and models

	Benchmarks							GBM				MARS			
	FA-AR	AR	LASSO	RF	Tree	NN	DNN	Tuned	Plain	B & P	Booging	Tuned	Plain	B & P	Quake
Abalone			0.52	0.56	0.45	0.54	0.53	0.50	0.48	0.53*	0.54**	0.57	0.35*	0.31*	**0.58*** **
Boston Housing			0.67	0.88	0.79	0.86	0.85	0.89	0.88	0.90	0.85*	0.83	0.87	**0.92**	0.91
Auto			0.66	**0.71**	0.61	0.13	0.64	0.64	0.59**	0.65	0.64*	0.71	−0.54 *	0.53	0.63
Bike Sharing			0.38	0.91	0.73	0.88	0.94	**0.95**	0.93***	0.91***	0.91***	0.71	0.89***	0.87***	0.90***
White Wine			0.28	**0.52**	0.28	0.37	0.26	0.37	0.32*	0.44***	0.38	0.33	0.33***	0.39**	0.38***
Red Wine			0.34	**0.47**	0.35	0.33	0.37	0.37	0.23**	0.37	0.38	0.38	0.29*	0.33	0.35
Concrete			0.59	0.90	0.71	0.89	0.88	**0.92**	0.92	0.90*	0.90***	0.83	0.87	0.30***	0.89
Fish Toxicity			0.56	**0.65**	0.57	0.60	0.63	0.63	0.54***	0.61	0.62	0.56	−0.25 ***	0.54*	0.61
Forest Fire			0.00	−0.11	0.00	−0.02	0.01	−0.03	−0.68 ***	−0.32 ***	−0.08	**0.01**	−1.55 *	−0.68	−0.36
NBA Salary			0.52	**0.60**	0.34	0.22	0.21	0.50	0.29***	0.49	0.50	0.36	0.11*	0.59*	0.53
CA Housing			0.64	0.82	0.59	0.75	0.74	0.82	0.82	**0.83*** **	0.82**	0.72	0.77***	0.81***	0.79***
Crime Florida			0.66	0.79	0.60	**0.82**	0.75	0.75	0.77	0.81*	0.79	0.70	0.44*	**0.81**	0.80
F1 $R^2 = 0.7$			0.53	0.62	0.50	0.43	0.51	0.65	0.54***	0.60***	0.67**	0.68	0.55	0.62	**0.69*** **
F1 $R^2 = 0.4$			0.32	0.40	0.36	0.19	0.28	0.40	0.16***	0.34*	0.41	**0.41**	0.14*	0.35	0.40*
GDP h=1	0.27	0.27	0.24	0.35	0.18	0.06	0.26	0.36	0.17	0.37	**0.38**	0.00	−9.08 ***	−0.45 **	−0.12 **
GDP h=2	−0.03	0.17	−0.01	0.16	0.00	−0.06	−0.52	0.15	−0.56 **	**0.20**	0.18	−0.40	−4.37 **	−0.41 *	−0.37 ***
UNRATE h=1	**0.71**	0.53	0.43	0.59	0.22	−0.69	0.62	0.59	0.66	0.58	0.65	−0.65	−0.72 ***	0.53	0.68
UNRATE h=2	**0.52**	0.29	0.26	0.37	0.16	0.14	0.41	0.43	0.35	0.42	0.48	0.16	−0.80 **	−0.28	0.26
INF h=1	0.25	0.33	0.43	0.42	0.25	0.41	**0.49**	0.35	0.24	0.37	0.39	0.37	−0.57 **	0.45	0.34
INF h=2	0.05	0.22	0.09	0.28	0.45	0.19	**0.51**	0.15	−0.26 ***	0.16	0.27*	0.39	−2.50 **	0.24	0.42

Notes: This table reports R^2_{test} for 20 data sets and different models, either standard or introduced in the text. For macroeconomic targets (the last 6 data sets), the set of benchmark models additionally includes an autoregressive model of order 2 (AR) and a factor-augmented regression with 2 lags (FA-AR). Numbers in bold identify the best predictive performance of the row. For GBM and MARS, t-test (and Diebold and Mariano (2002) tests for time series data) are performed to evaluate whether the difference in predictive performance between the tuned version and the remaining three models of each block is statistically significant. '*', '**' and '***' respectively refer to p-values below 5%, 1% and 0.1%. F1 means "Friedman 1" DGP of Friedman (1991).

Table 23: R^2_{train} for all data sets and models

	Benchmarks							GBM				MARS			
	FA-AR	AR	LASSO	RF	Tree	NN	DNN	Tuned	Plain	B & P	Booging	Tuned	Plain	B & P	Quake
Abalone			0.50	0.92	0.50	0.60	0.59	0.53	0.85	0.86	0.91	0.57	0.65	0.78	0.61
Boston Housing			0.72	0.98	0.87	0.90	0.89	1.00	1.00	0.99	0.99	0.90	0.97	0.97	0.98
Auto			0.68	0.96	0.77	0.13	0.81	0.86	1.00	0.98	0.98	0.77	0.98	0.93	0.96
Bike Sharing			0.38	0.98	0.89	0.95	0.96	0.95	0.94	0.95	0.71	0.89	0.88	0.90	
White Wine			0.26	0.92	0.27	0.47	0.75	0.44	0.82	0.85	0.88	0.37	0.46	0.52	0.51
Red Wine			0.29	0.91	0.41	0.40	0.42	0.41	0.96	0.94	0.95	0.44	0.56	0.69	0.67
Concrete			0.61	0.98	0.75	0.91	0.93	0.98	0.99	0.98	0.99	0.88	0.98	0.74	0.95
Fish Toxicity			0.54	0.93	0.60	0.64	0.61	0.92	0.97	0.95	0.97	0.63	0.96	0.82	0.88
Forest Fire			0.00	0.81	0.00	0.00	0.07	0.40	0.97	0.88	0.91	0.04	0.62	0.73	0.76
NBA Salary			0.47	0.93	0.72	0.65	0.71	0.99	1.00	0.97	0.97	0.64	0.92	0.84	0.93
CA Housing			0.63	0.97	0.61	0.78	0.85	0.86	0.89	0.91	0.90	0.72	0.80	0.83	0.81
Crime Florida			0.65	0.96	0.84	0.88	0.94	1.00	1.00	0.98	0.98	0.75	1.00	0.97	0.98
F1 $R^2 = 0.7$			0.45	0.93	0.45	0.62	0.71	0.95	1.00	0.97	0.97	0.65	0.81	0.84	0.86
F1 $R^2 = 0.4$			0.23	0.89	0.30	0.34	0.35	0.48	1.00	0.94	0.94	0.38	0.64	0.75	0.76
GDP h=1	0.41	0.11	0.23	0.91	0.51	0.26	0.44	0.81	1.00	0.96	0.96	0.47	1.00	0.94	0.94
GDP h=2	0.26	0.06	0.07	0.89	0.00	0.26	0.55	0.76	1.00	0.95	0.95	0.29	1.00	0.94	0.95
UNRATE h=1	0.57	0.40	0.48	0.93	0.81	-0.07	0.82	0.83	1.00	0.97	0.97	0.76	0.99	0.97	0.96
UNRATE h=2	0.41	0.13	0.35	0.92	0.38	0.42	0.25	0.99	1.00	0.96	0.96	0.75	1.00	0.96	0.96
INF h=1	0.76	0.73	0.90	0.97	0.81	0.64	0.94	1.00	1.00	0.99	0.99	0.73	1.00	0.99	0.99
INF h=2	0.69	0.63	0.72	0.96	0.72	0.67	0.92	1.00	1.00	0.99	0.98	0.81	1.00	0.99	0.98

Notes: This table reports R^2_{train} for 20 data sets and different models, either standard or introduced in the text. For macroeconomic targets (the last 6 data sets), the set of benchmark models additionally includes an autoregressive model of order 2 (AR) and a factor-augmented regression with 2 lags (FA-AR). F1 means "Friedman 1" DGP of Friedman (1991).

3.6.2. Implementation Details for Booging and MARSquake

Booging and MARSquake are the `B` & `P` +`DA` versions of Boosted Trees and MARS, respectively. The data-augmentation option will likely be redundant in high-dimensional situations where the available regressors already have a factor structure (like macroeconomic data).

ABOUT B. For both algorithms, `B` is made operational by subsampling. As usual, reasonable candidates for the sampling rate are $2/3$ and $3/4$. All ensembles use $B = 100$ subsamples.

ABOUT P. The primary source of perturbation in Booging is straightforward. Using subsamples to construct trees at each step is already integrated within Stochastic Gradient Boosting. By construction, it perturbs the Boosting fitting path and achieve a similar goal as that of the original `mtry` in RF. Note that, for fairness, this standard feature is also activated for any reported results on "plain" Boosting.

The implementation of `P` in MARSquake is more akin to that of RF. At each step of the forward pass, MASS evaluate all variables as potential candidates to enter a hinge function, and select the one which (greedily) maximize fit at this step. In the spirit of RF's `mtry`, `P` is applied by stochastically restricting the set of available features at each step. I set the fraction of randomly considered X's to $1/2$.

To further enhance perturbation in both algorithms, we can randomly drop a fraction of features from base learners' respective information sets. Since `DA` creates replicas of the data and keep some of its correlation structure, features are unlikely to be entirely dropped from a boosting run, provided the dropping rate is not too high. I suggest 20%. This can is analogous to `mtry`-like randomly select features, but for a whole tree (in RF) rather than at each split.

ABOUT DA. Perturbation work better if there is a lot to perturb. In many data sets, X is rich in observations but contains few regressors. To assure `P` meets its full randomization potential, a cheap data augmentation procedure can be carried. `DA` is simply adding fake regressors that are correlated with the original X and maintain in part their cross-correlation structure. Say X contains K regressors. I take the $N \times K$ matrix X and create two duplicates $\tilde{X} = X + \mathcal{E}$ where $\mathcal{E}$ is a matrix of Gaussian noise. SD is set to $1/3$ that of the variable. For X_k's that are either categorical or ordinal, I create the corresponding $\tilde{X}_k$ by taking X_k and shuffling 20% of its observations.

LAST WORD ON MARS. It is known that standard MARS has a forward and a backward pass. The latter's role is to prevent overfitting by (traditional) pruning. Obviously, there is no backward pass in MARSquake. Certain implementations of MARS (like *earth*, Milborrow (2018)) may contain foolproof features rendering the forward pass recalcitrant to blatantly overfit in certain situations (usually when regressor are not numerous). To partially circumvent this rare occurrence, one can run MARS again on residuals obtained from a first MARS run which failed to attain a high enough R^2_{train}.

3.6.3. Bagging and Heteroscedasticity

Grandvalet (2004) expands on Breiman (1996) and discuss in greater detail why bagging can boost trees' performance but not so much for OLS or splines. His argument basically boils down that trees are non-linear functionals of the data while splines or OLS are just linear combinations of the data. In the case of OLS, perturbing the data weights B times gives a similar $\hat{\beta}$ as computing OLS with all weights being equal to 1. However, $\hat{E}^{\text{tree}}_{\boldsymbol{\omega}}(y|X)$ can be very far from just computing the same expectation at the mean $\omega_i = 1 \ \forall i$. Hence, if ω_i in

$$y_i = \mathcal{T}(X_i) + \omega_i \epsilon_i, \qquad \epsilon_i \sim N(0,1) \tag{3.6}$$

follows a certain non-degenerate distribution, it is argued that bagging will yield significant improvements. Of course, under these conditions (and a linear DGP), OLS would still be consistent, so that as the sample gets large, heteroscedascity does not compromise prediction.[41] That is, $\hat{E}^{\text{OLS}}(y|X;\boldsymbol{\omega}) \to \hat{E}^{\text{OLS}}(y|X;\hat{\boldsymbol{\omega}} = \mathbf{1})$ as the sample size grows. No such guarantees are available for complicated non-linear recursive estimators, such as trees.

Such reasoning can be extended to finite samples and in a straightforward application of a basic property of expectations: $E(f(\omega)) \neq f(E(\omega))$ unless f is linear in ω. If f is only mildly non-linear – like for the OLS or ridge functional, the shortcut $f(E(\omega))$ will be a reliable approximation to the real expectation of interest (Breiman (1996) refers to those as "stable" predictors). If $f = \mathcal{T}$, the shortcut likely provides an abysmal approximation. An alternative is to resort to "pairs" bootstrap (or subsampling) to implicitly simulate from a plausible distribution of ω_i and then use the mean over many bootstrapped trees to obtain $\hat{E}^{\text{tree}}_{\boldsymbol{\omega}}(y|X)$. Coming back to the main point of this chapter, it is clear that pruning CART is an imperfect enterprise because the model it is pruning will not coincide to the true conditional expectation *if* ω_i's are heterogeneous.

Nevertheless, relying on presumed "badness" in the data to justify RF's usual supremacy

[41] A different story occurs in small samples where down-weighting noisy observations can provide substantial improvements. One example out of many is the use of stochastic volatility to improve (even) point forecasts in a macroeconomic context.

over a single tree seems thin. There are many examples where heteroscedascity is visibly absent from the test set errors and yet, RF will do much better than (pruned) CART.

3.6.4. Additional NN details

For both neural networks, the batch size is 32 and the optimizer is Adam (with Keras default values). Continuous X's are normalized so that all values are within the 0-1 range.

NN in Table 22 is a standard feed-forward networks with an architecture in the vein of Gu et al. (2020). There are two hidden layers, the first with 32 neurons and the second with 16 neurons. The number of epochs is fixed at 100. The activation function is *ReLu* and that of the output layer is linear. The learning rate $\in \{0.001, 0.01\}$ and the LASSO λ parameter $\in \{0.001, 0.0001\}$ are chosen by 5-fold cross-validation. A batch normalization layer follows each *ReLu* layers. Early stopping is applied by stopping training whenever 20 epochs pass without any improvement of the cross-validation MSE.

DNN in Table 22 is a standard feed-forward networks with an architecture closely following that of Olson and Wyner (2018) for small data sets. There are 10 hidden layers, each featuring 100 neurons. The number of epochs is fixed at 200. The activation function is *eLu* and that of the output layer is linear. The learning rate $\in \{0.001, 0.01, 0.1\}$ and the LASSO λ parameter $\in \{0.001, 0.00001\}$ are chosen by 5-fold cross-validation. No early stopping is applied.

Bibliography

K. A. Aastveit, A. S. Jore, and F. Ravazzolo. Identification and real-time forecasting of norwegian business cycles. *International Journal of Forecasting*, 32(2):283–292, 2016.

A. Abbate, S. Eickmeier, W. Lemke, and M. Marcellino. The changing international transmission of financial shocks: evidence from a classical time-varying favar. *Journal of Money, Credit and Banking*, 48(4):573–601, 2016.

T. Adrian, N. Boyarchenko, and D. Giannone. Vulnerable growth. *American Economic Review*, 109(4):1263–89, 2019.

W. P. Alexander and S. D. Grimshaw. Treed regression. *Journal of Computational and Graphical Statistics*, 5(2):156–175, 1996.

S. Almon. The distributed lag between capital appropriations and expenditures. *Econometrica: Journal of the Econometric Society*, pages 178–196, 1965.

P. Amir-Ahmadi, C. Matthes, and M.-C. Wang. Choosing prior hyperparameters: with applications to time-varying parameter models. *Journal of Business & Economic Statistics*, pages 1–13, 2018.

T. W. Anderson. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *The Annals of Mathematical Statistics*, 22(3):327–351, 1951.

S. B. Aruoba, L. Bocola, and F. Schorfheide. Assessing dsge model nonlinearities. *Journal of Economic Dynamics and Control*, 83:34–54, 2017.

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

A. Atkeson, L. E. Ohanian, et al. Are phillips curves useful for forecasting inflation? *Federal Reserve bank of Minneapolis quarterly review*, 25(1):2–11, 2001.

A. J. Auerbach and Y. Gorodnichenko. Fiscal multipliers in recession and expansion. In *Fiscal policy after the financial crisis*, pages 63–98. University of Chicago Press, 2012a.

A. J. Auerbach and Y. Gorodnichenko. Measuring the output responses to fiscal policy. *American Economic Journal: Economic Policy*, 4(2):1–27, 2012b.

J. Bai and S. Ng. Forecasting economic time series using targeted predictors. *Journal of Econometrics*, 146(2):304–317, 2008.

J. Bai and S. Ng. Principal components and regularized estimation of factor models. *arXiv preprint arXiv:1708.08137*, 2017.

M. Bańbura, D. Giannone, and L. Reichlin. Large bayesian vector auto regressions. *Journal of Applied Econometrics*, 25(1):71–92, 2010.

S. M. Barakchian and C. Crowe. Monetary policy matters: Evidence from new shocks data. *Journal of Monetary Economics*, 60(8):950–966, 2013.

P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 2020.

N. Batini, G. Callegari, and G. Melina. Successful austerity in the united states, europe and japan. 2012.

C. Baumeister and L. Kilian. What central bankers need to know about forecasting oil prices. *International Economic Review*, 55:869–889, 2014. URL https://EconPapers.repec.org/RePEc:wly:iecrev:v:55:y:2014:i::p:869-889.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019a.

M. Belkin, A. Rakhlin, and A. B. Tsybakov. Does data interpolation contradict statistical optimality? In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1611–1619. PMLR, 2019b.

M. A. Belmonte, G. Koop, and D. Korobilis. Hierarchical shrinkage in time-varying parameter models. *Journal of Forecasting*, 33(1):80–94, 2014.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

C. Bergmeir, R. J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, 2018.

B. S. Bernanke, J. Boivin, and P. Eliasz. Measuring the effects of monetary policy: a factor-augmented vector autoregressive (favar) approach. *The Quarterly journal of economics*, 120(1):387–422, 2005.

A. Bitto and S. Frühwirth-Schnatter. Achieving shrinkage in a time-varying parameter model framework. *Journal of Econometrics*, 2018.

O. Blanchard. The phillips curve: Back to the'60s? *American Economic Review*, 106(5):31–34, 2016.

O. Blanchard, E. Cerutti, and L. Summers. Inflation and activity–two explorations and their monetary policy implications. Technical report, National Bureau of Economic Research, 2015.

J. Boivin. Has us monetary policy changed? evidence from drifting coefficients and real-time data. Technical report, National Bureau of Economic Research, 2005.

D. Borup, B. J. Christensen, N. N. Mühlbach, M. S. Nielsen, et al. Targeting predictors in random forest regression. Technical report, Department of Economics and Business Economics, Aarhus University, 2020a.

D. Borup, D. Rapach, and E. C. M. Schütte. Now-and backcasting initial claims with high-dimensional daily internet search-volume data. *Available at SSRN 3690832*, 2020b.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

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

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

P. Bühlmann, B. Yu, et al. Analyzing bagging. *The Annals of Statistics*, 30(4):927–961, 2002.

M. Camacho and G. P. Quiros. Jump-and-rest effect of us business cycles. *Studies in Nonlinear Dynamics & Econometrics*, 11(4), 2007.

A. Carriero, G. Kapetanios, and M. Marcellino. Forecasting large datasets with bayesian reduced rank multivariate models. *Journal of Applied Econometrics*, 26(5):735–761, 2011.

K. Carstensen, M. Heinrich, M. Reif, and M. H. Wolters. Predicting ordinary and severe recessions with a three-state markov-switching dynamic factor model: An application to the german business cycle. *International Journal of Forecasting*, 36(3):829–850, 2020.

J. L. Castle, J. A. Doornik, D. F. Hendry, and F. Pretis. Detecting location shifts during model selection by step-indicator saturation. *Econometrics*, 3(2):240–264, 2015.

G. Chamberlain and G. W. Imbens. Nonparametric applications of bayesian inference. *Journal of Business & Economic Statistics*, 21(1):12–18, 2003.

J. Champagne and R. Sekkel. Changes in monetary regimes and the identification of monetary policy shocks: Narrative evidence from canada. *Journal of Monetary Economics*, 99:72–87, 2018.

J. C. Chan and E. Eisenstat. Comparing hybrid time-varying parameter vars. *Economics Letters*, 171:1–5, 2018.

J. C. Chan, E. Eisenstat, and R. W. Strachan. Reducing dimensions in a large TVP-VAR. CAMA Working Papers 2018-49, Centre for Applied Macroeconomic Analysis, Crawford School of Public Policy, The Australian National University, Oct. 2018. URL https://ideas.repec.org/p/een/camaaa/2018-49.html.

B. Chen and Y. Hong. Testing for smooth structural changes in time series models via nonparametric regression. *Econometrica*, 80(3):1157–1183, 2012.

J. C. Chen, A. Dunn, K. K. Hood, A. Driessen, and A. Batch. Off to the races: A comparison of machine learning and alternative data for predicting economic indicators. In *Big Data for 21st Century Economic Statistics*. University of Chicago Press, 2019.

G. Chevillon. Direct multi-step estimation and forecasting. *Journal of Economic Surveys*, 21 (4):746–785, 2007.

P. Cirillo and P. Muliere. An urn-based bayesian block bootstrap. *Metrika*, 76(1):93–106, 2013.

R. Clarida, J. Gali, and M. Gertler. Monetary policy rules and macroeconomic stability: evidence and some theory. *The Quarterly journal of economics*, 115(1):147–180, 2000.

R. T. Clemen. Combining forecasts: A review and annotated bibliography. *International journal of forecasting*, 5(4):559–583, 1989.

M. P. Clements and J. Smith. The performance of alternative forecasting methods for setar models. *International Journal of Forecasting*, 13(4):463–475, 1997.

J. Cloyne and P. Hürtgen. The macroeconomic effects of monetary policy: a new measure for the united kingdom. *American Economic Journal: Macroeconomics*, 8(4):75–102, 2016.

M. Clyde and H. Lee. Bagging and the bayesian bootstrap. In *AISTATS*, 2001.

T. Cogley and T. J. Sargent. Evolving post-world war ii us inflation dynamics. *NBER macroeconomics annual*, 16:331–373, 2001.

T. Cogley and T. J. Sargent. Drifts and volatilities: monetary policies and outcomes in the post wwii us. *Review of Economic dynamics*, 8(2):262–302, 2005.

T. Cogley, G. E. Primiceri, and T. J. Sargent. Inflation-gap persistence in the us. *American Economic Journal: Macroeconomics*, 2(1):43–69, 2010.

O. Coibion and Y. Gorodnichenko. Is the phillips curve alive and well after all? inflation expectations and the missing disinflation. *American Economic Journal: Macroeconomics*, 7(1):197–232, 2015.

A. D'Agostino, L. Gambetti, and D. Giannone. Macroeconomic forecasting and structural change. *Journal of Applied Econometrics*, 28(1):82–101, 2013.

E. B. Dagum and S. Bianconcini. Equivalent reproducing kernels for smoothing spline predictors. In *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, 2009a.

E. B. Dagum and S. Bianconcini. Equivalent reproducing kernels for smoothing spline predictors. In *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, 2009b.

J. de Wind and L. Gambetti. Reduced-rank time-varying vector autoregressions. CPB Discussion Paper 270, CPB Netherlands Bureau for Economic Policy Analysis, Mar. 2014. URL https://ideas.repec.org/p/cpb/discus/270.html.

M. Del Negro and G. E. Primiceri. Time varying structural vector autoregressions and monetary policy: a corrigendum. *The review of economic studies*, 82(4):1342–1345, 2015.

M. Del Negro, M. P. Giannoni, and F. Schorfheide. Inflation in the great recession and new keynesian models. *American Economic Journal: Macroeconomics*, 7(1):168–96, 2015.

M. Del Negro, M. Lenza, G. E. Primiceri, and A. Tambalotti. What's up with the phillips curve? Technical report, National Bureau of Economic Research, 2020.

D. Delle Monache, A. De Polis, and I. Petrella. Modeling and forecasting macroeconomic downside risk. 2020.

F. X. Diebold and R. S. Mariano. Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144, 2002.

F. X. Diebold and G. D. Rudebusch. Measuring business cycles: A modern perspective. Technical report, National Bureau of Economic Research, 1994.

T. Doan, R. Litterman, and C. Sims. Forecasting and conditional projection using realistic prior distributions. *Econometric reviews*, 3(1):1–100, 1984.

J. J. Dolado, R. Maria-Dolores, and M. Naveira. Are monetary-policy reaction functions asymmetric?: The role of nonlinearity in the phillips curve. *European Economic Review*, 49(2):485–503, 2005.

A. Doser, R. C. Nunes, N. Rao, and V. Sheremirov. Inflation expectations and nonlinearities in the phillips curve. 2017.

R. Duroux and E. Scornet. Impact of subsampling and pruning on random forests. *arXiv preprint arXiv:1603.04261*, 2016.

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

P. Eliasz, J. H. Stock, and M. W. Watson. Optimal tests for reduced rank time variation in regression coefficients and for level variation in the multivariate local level model. *manuscript, Harvard University*, 2004.

G. Elliott, A. Gargano, and A. Timmermann. Complete subset regressions. *Journal of Econometrics*, 177(2):357–373, 2013.

A. Estrella and F. S. Mishkin. Predicting us recessions: Financial variables as leading indicators. *Review of Economics and Statistics*, 80(1):45–61, 1998.

J. Faust and J. H. Wright. Forecasting inflation. In *Handbook of economic forecasting*, volume 2, pages 2–56. Elsevier, 2013.

J. Feng, Y. Yu, and Z.-H. Zhou. Multi-layered gradient boosting decision trees. In *Advances in neural information processing systems*, pages 3551–3561, 2018.

T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

O. Fortin-Gagnon, M. Leroux, D. Stevanovic, and S. Surprenant. A large canadian database for macroeconomic analysis. Technical report, Department of Economics, UQAM, 2018.

D. A. Freedman et al. Bootstrapping regression models. *The Annals of Statistics*, 9(6):1218–1228, 1981.

R. Friedberg, J. Tibshirani, S. Athey, and S. Wager. Local linear forests. *arXiv preprint arXiv:1807.11408*, 2018.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

J. H. Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

J. H. Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4): 367–378, 2002.

F. Frommlet and G. Nuel. An adaptive ridge procedure for l0 regularization. *PloS one*, 11 (2):e0148620, 2016.

J. Galí and L. Gambetti. Has the us wage phillips curve flattened? a semi-structural exploration. Technical report, National Bureau of Economic Research, 2019.

E. Ghysels, A. Sinko, and R. Valkanov. Midas regressions: Further results and new directions. *Econometric Reviews*, 26(1):53–90, 2007.

L. Giraitis, G. Kapetanios, and T. Yates. Inference on stochastic time-varying coefficient models. *Journal of Econometrics*, 179(1):46–65, 2014.

L. Giraitis, G. Kapetanios, and T. Yates. Inference on multivariate heteroscedastic time varying random coefficient models. *Journal of Time Series Analysis*, 39(2):129–149, 2018.

G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

T. Götz and K. Hauzenberger. Large mixed-frequency vars with a parsimonious time-varying parameter structure. 2018.

P. Goulet Coulombe. Time-varying parameters as ridge regressions. *arXiv preprint arXiv:2009.00401*, 2020a.

P. Goulet Coulombe. The macroeconomy as a random forest. *arXiv preprint arXiv:2006.12724*, 2020b.

P. Goulet Coulombe. To bag is to prune. *arXiv preprint arXiv:2008.07063*, 2020c.

P. Goulet Coulombe. Slow-growing trees. Technical report, 2021.

P. Goulet Coulombe, M. Leroux, D. Stevanovic, S. Surprenant, et al. How is machine learning useful for macroeconomic forecasting? Technical report, CIRANO, 2019.

P. Goulet Coulombe, M. Leroux, D. Stevanovic, S. Surprenant, et al. Macroeconomic data transformations matter. Technical report, CIRANO, 2020a.

P. Goulet Coulombe, M. Leroux, D. Stevanovic, S. Surprenant, et al. Prévision de l'activité économique au québec et au canada à l'aide des méthodes "machine learning". Technical report, Technical report, CIRANO, 2020b.

P. Goulet Coulombe, M. Marcellino, and D. Stevanovic. Can machine learning catch the covid-19 recession? *CEPR Discussion Paper No. DP15867*, 2021.

Y. Grandvalet. Least absolute shrinkage is equivalent to quadratic penalization. In *ICANN 98*, pages 201–206. Springer, 1998.

Y. Grandvalet. Bagging equalizes influence. *Machine Learning*, 55(3):251–270, 2004.

C. W. Granger. Non-linear models: Where do we go next-time varying parameter models? *Studies in Nonlinear Dynamics & Econometrics*, 12(3), 2008.

A. L. Grant and J. C. Chan. A bayesian model comparison for trend-cycle decompositions of output. *Journal of Money, Credit and Banking*, 49(2-3):525–552, 2017.

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

P. R. Hahn, C. M. Carvalho, and S. Mukherjee. Partial factor modeling: predictor-dependent shrinkage for linear regression. *Journal of the American Statistical Association*, 108(503):999–1008, 2013.

J. D. Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, NJ, 1994.

B. E. Hansen. Threshold autoregression in economics. *Statistics and its Interface*, 4(2):123–127, 2011.

C. Hansen and Y. Liao. The factor-lasso and k-step bootstrap approach for inference in high-dimensional economic applications. *Econometric Theory*, 35(3):465–509, 2019.

H. Hassani, S. Heravi, and A. Zhigljavsky. Forecasting european industrial production with singular spectrum analysis. *International journal of forecasting*, 25(1):103–118, 2009.

H. Hassani, A. S. Soofi, and A. Zhigljavsky. Predicting inflation dynamics with singular spectrum analysis. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 176(3):743–760, 2013.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

N. Hauzenberger, F. Huber, and G. Koop. Dynamic shrinkage priors for large time-varying parameter regressions using scalable markov chain monte carlo methods. *arXiv preprint arXiv:2005.03906*, 2020.

K.-P. Hellwig. *Overfitting in Judgment-based Economic Forecasts: The Case of IMF Growth Projections*. International Monetary Fund, 2018.

E. Hillebrand and M. C. Medeiros. The benefits of bagging for forecast models of realized volatility. *Econometric Reviews*, 29(5-6):571–593, 2010.

E. Hillebrand, M. Lukas, W. Wei, et al. Bagging weak predictors. Technical report, Monash University, Department of Econometrics and Business Statistics, 2020.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

D. R. Hoover, J. A. Rice, C. O. Wu, and L.-P. Yang. Nonparametric smoothing estimates of time-varying coefficient models with longitudinal data. *Biometrika*, 85(4):809–822, 1998.

F. Huber, G. Koop, and M. Pfarrhofer. Bayesian inference in high-dimensional time-varying parameter models using integrated rotated gaussian approximations. *arXiv preprint arXiv:2002.10274*, 2020.

A. Inoue and L. Kilian. How useful is bagging in forecasting economic time series? a case study of us consumer price inflation. *Journal of the American Statistical Association*, 103 (482):511–522, 2008.

M. Ito, A. Noda, and T. Wada. International stock market efficiency: a non-bayesian time-varying model approach. *Applied Economics*, 46(23):2744–2754, 2014.

M. Ito, A. Noda, and T. Wada. An alternative estimation method of a time-varying parameter model. Technical report, Working Paper, Faculty of Economics, Keio University, Japan, 2017.

Ò. Jordà. Estimation and inference of impulse responses by local projections. *American economic review*, 95(1):161–182, 2005.

K. R. Kadiyala and S. Karlsson. Numerical methods for estimation and inference in bayesian var-models. *Journal of Applied Econometrics*, 12(2):99–132, 1997.

G. Kapetanios, M. Marcellino, and F. Venditti. Large time-varying parameter vars: A non-parametric approach. *Journal of Applied Econometrics*, 34(7):1027–1049, 2019.

G. Karabatsos. A dirichlet process functional approach to heteroscedastic-consistent covariance estimation. *International Journal of Approximate Reasoning*, 78:210–222, 2016.

B. T. Kelly, S. Pruitt, and Y. Su. Instrumented principal component analysis. 2017.

M. Y. Khan. *Advances in applied nonlinear time series modeling*. PhD thesis, lmu, 2015.

L. Kilian and H. Lütkepohl. *Structural vector autoregressive analysis*. Cambridge University Press, 2017.

G. S. Kimeldorf and G. Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2): 495–502, 1970.

D. Kobak, J. Lomond, and B. Sanchez. The optimal ridge penalty for real-world high-dimensional data can be zero or negative due to the implicit ridge regularization. *Journal of Machine Learning Research*, 21(169):1–16, 2020.

G. Koop and D. Korobilis. Large time-varying parameter vars. *Journal of Econometrics*, 177 (2):185–198, 2013.

G. M. Koop. *Bayesian econometrics*. John Wiley & Sons Inc., 2003.

G. M. Koop. Forecasting with medium and large bayesian vars. *Journal of Applied Econometrics*, 28(2):177–203, 2013.

D. Korobilis. Data-based priors for vector autoregressions with drifting coefficients. 2014.

D. Korobilis. High-dimensional macroeconomic forecasting using message passing algorithms. *Journal of Business & Economic Statistics*, pages 1–12, 2019.

R. Kotchoni, M. Leroux, and D. Stevanovic. Macroeconomic forecast accuracy in a data-rich environment. *Journal of Applied Econometrics*, 34(7):1050–1072, 2019.

D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):1–15, 2014.

T. Lancaster. A note on bootstraps and robustness. *Available at SSRN 896764*, 2003.

E. E. Leamer. Housing is the business cycle. Technical report, National Bureau of Economic Research, 2007.

T.-H. Lee, A. Ullah, and R. Wang. Bootstrap aggregating and random forest. In *Macroeconomic Forecasting in the Era of Big Data*, pages 389–429. Springer, 2020.

D. LeJeune, H. Javadi, and R. Baraniuk. The implicit regularization of ordinary least squares ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 3525–3535, 2020.

Y. Lin, H. H. Zhang, et al. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

J. Lindé and M. Trabandt. Resolving the missing deflation puzzle. 2019.

Z. Liu and G. Li. Efficient regularized regression for variable selection with l0 penalty. *arXiv preprint arXiv:1407.7508*, 2014.

G. Louppe. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*, 2014.

R. E. Lucas. Econometric policy evaluation: A critique. In *Carnegie-Rochester conference series on public policy*, volume 1, pages 19–46, 1976.

A. Lusompa. Local projections, autocorrelation, and efficiency. *Autocorrelation, and Efficiency (March 29, 2020)*, 2020.

J. G. MacKinnon. Bootstrap methods in econometrics. *Economic Record*, 82:S2–S18, 2006.

M. McCracken and S. Ng. Fred-qd: A quarterly database for macroeconomic research. Technical report, National Bureau of Economic Research, 2020.

M. W. McCracken and S. Ng. Fred-md: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589, 2016.

M. C. Medeiros, G. F. Vasconcelos, Á. Veiga, and E. Zilberman. Forecasting inflation in a data-rich environment: the benefits of machine learning methods. *Journal of Business & Economic Statistics*, (just-accepted):1–45, 2019.

C. Meek, D. M. Chickering, and D. Heckerman. Autoregressive tree models for time-series analysis. In *Proceedings of the 2002 SIAM International Conference on Data Mining*, pages 229–244. SIAM, 2002.

N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun): 983–999, 2006.

L. Mentch and S. Zhou. Randomization as regularization: A degrees of freedom explanation for random forest success. *arXiv preprint arXiv:1911.00190*, 2019.

S. Milborrow. *earth: Multivariate Adaptive Regression Splines*, 2018. URL https://CRAN.R-project.org/package=earth. R package.

T. Mineyama. Downward nominal wage rigidity and inflation dynamics during and after the great recession. *Available at SSRN 3157995*, 2020.

C. Molnar. *Interpretable machine learning*. Lulu.com, 2019.

A. Mukherjee and J. Zhu. Reduced rank ridge regression and its kernel extensions. *Statistical analysis and data mining: the ASA data science journal*, 4(6):612–622, 2011.

S. Mullainathan and J. Spiess. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

M. Olson, A. J. Wyner, and R. Berk. Modern neural networks generalize on small data sets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3623–3632, 2018.

M. A. Olson and A. J. Wyner. Making sense of random forest probabilities: a kernel perspective. *arXiv preprint arXiv:1812.05792*, 2018.

P. Perron et al. Dealing with structural breaks. *Palgrave handbook of econometrics*, 1(2):278–352, 2006.

K. Petrova. A quasi-bayesian local likelihood approach to time varying parameter var models. *Journal of Econometrics*, 212(1):286–306, 2019.

D. Pettenuzzo and A. Timmermann. Forecasting macroeconomic variables under model instability. *Journal of Business & Economic Statistics*, 35(2):183–201, 2017.

D. J. Poirier. Bayesian interpretations of heteroskedastic consistent covariance estimators using the informed bayesian bootstrap. *Econometric Reviews*, 30(4):457–468, 2011.

G. E. Primiceri. Time varying structural vector autoregressions and monetary policy. *The Review of Economic Studies*, 72(3):821–852, 2005.

V. A. Ramey. Macroeconomic shocks and their propagation. In *Handbook of macroeconomics*, volume 2, pages 71–162. Elsevier, 2016.

V. A. Ramey and S. Zubairy. Government spending multipliers in good times and in bad: evidence from us historical data. *Journal of Political Economy*, 126(2):850–901, 2018a.

V. A. Ramey and S. Zubairy. Government spending multipliers in good times and in bad: evidence from us historical data. *Journal of Political Economy*, 126(2):850–901, 2018b.

D. Rapach and G. Zhou. Forecasting stock returns. In *Handbook of economic forecasting*, volume 2, pages 328–383. Elsevier, 2013.

C. E. Rasmussen. *Evaluation of Gaussian processes and other methods for non-linear regression*. PhD thesis, University of Toronto Toronto, Canada, 1997.

M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

C. D. Romer and D. H. Romer. A new measure of monetary shocks: Derivation and implications. *American Economic Review*, 94(4):1055–1084, 2004.

D. B. Rubin. The bayesian bootstrap. *The annals of statistics*, pages 130–134, 1981.

G. Ruisi. Time-varying local projections. Technical report, Working Paper, 2019.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

E. Scornet, G. Biau, J.-P. Vert, et al. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015.

R. J. Shiller. A distributed lag estimator derived from smoothness priors. *Econometrica (pre-1986)*, 41(4):775, 1973.

C. A. Sims. A nine-variable probabilistic macroeconomic forecasting model. In *Business cycles, indicators and forecasting*, pages 179–212. University of Chicago press, 1993.

C. A. Sims and T. Zha. Were there regime switches in us monetary policy? *American Economic Review*, 96(1):54–81, 2006.

D. Stevanovic. Common time variation of parameters in reduced-form macroeconomic models. *Studies in Nonlinear Dynamics & Econometrics*, 20(2):159–183, 2016.

J. H. Stock. Unit roots, structural breaks and trends. *Handbook of econometrics*, 4:2739–2841, 1994.

J. H. Stock and M. W. Watson. New indexes of coincident and leading economic indicators. *NBER macroeconomics annual*, 4:351–394, 1989.

J. H. Stock and M. W. Watson. Evidence on structural instability in macroeconomic time series relations. *Journal of Business & Economic Statistics*, 14(1):11–30, 1996.

J. H. Stock and M. W. Watson. A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series. Technical report, National Bureau of Economic Research, 1998a.

J. H. Stock and M. W. Watson. Business cycle fluctuations in us macroeconomic time series. Technical report, National Bureau of Economic Research, 1998b.

J. H. Stock and M. W. Watson. Median unbiased estimation of coefficient variance in a time-varying parameter model. *Journal of the American Statistical Association*, 93(441): 349–358, 1998c.

J. H. Stock and M. W. Watson. Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, 20(2):147–162, 2002.

J. H. Stock and M. W. Watson. Why has us inflation become harder to forecast? *Journal of Money, Credit and banking*, 39:3–33, 2007.

J. H. Stock and M. W. Watson. Phillips curve inflation forecasts. Technical report, National Bureau of Economic Research, 2008.

J. H. Stock and M. W. Watson. Slack and cyclically sensitive inflation. Technical report, National Bureau of Economic Research, 2019.

M. Taddy, C.-S. Chen, J. Yu, and M. Wyle. Bayesian and empirical bayesian forests. *arXiv preprint arXiv:1502.02312*, 2015.

M. Taddy, M. Gardner, L. Chen, and D. Draper. A nonparametric bayesian analysis of heterogenous treatment effects in digital experimentation. *Journal of Business & Economic Statistics*, 34(4):661–672, 2016.

T. Teräsvirta. Specification, estimation, and evaluation of smooth transition autoregressive models. *Journal of the american Statistical association*, 89(425):208–218, 1994.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

R. Tibshirani, M. Wainwright, and T. Hastie. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.

A. Timmermann. Forecast combinations. *Handbook of economic forecasting*, 1:135–196, 2006.

H. Uhlig. Shocks, sign restrictions, and identification. *Advances in Economics and Econometrics*, 2:95, 2017.

G. Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

Y. Wang and I. H. Witten. Induction of model trees for predicting continuous classes. 1996.

P. Wei, Z. Lu, and J. Song. Variable importance analysis: a comprehensive review. *Reliability Engineering & System Safety*, 142:399–432, 2015.

D. Wochner. Dynamic factor trees and forests–a theory-led machine learning framework for non-linear and state-dependent short-term us gdp growth predictions. 2020.

N. Woloszko. Adaptive trees: a new approach to economic forecasting. 2020.

A. J. Wyner, M. Olson, J. Bleich, and D. Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18 (1):1558–1590, 2017.

Z.-H. Zhou and J. Feng. Deep forest. *arXiv preprint arXiv:1702.08835*, 2017.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.