Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Economics School of Economics

5-2020

Econometric methods and data Science techniques: A review of two strands of literature and an introduction to hybrid methods

Tian XIE

Jun YU
Singapore Management University, yujun@smu.edu.sg

Tao ZENG

Follow this and additional works at: https://ink.library.smu.edu.sg/soe_research

Part of the Econometrics Commons

Citation

XIE, Tian; Jun YU; and ZENG, Tao. Econometric methods and data Science techniques: A review of two strands of literature and an introduction to hybrid methods. (2020). 1-132. Research Collection School Of Economics.
Available at: https://ink.library.smu.edu.sg/soe_research/2392

This Working Paper is brought to you for free and open access by the School of Economics at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Economics by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

SMU ECONOMICS & STATISTICS

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Econometric Methods and Data Science Techniques: A Review of Two Strands of Literature and an Introduction to Hybrid Methods

Tian Xie, Jun Yu, Tao Zeng

May 2020

Paper No. 16-2020

ANY OPINION EXPRESSED ARE THOSE OF THE AUTHOR(S) AND NOT NECESSARILY THOSE OF
THE SCHOOL OF ECONOMICS, SMU

Econometric Methods and Data Science Techniques: A Review of Two Strands of Literature and an Introduction to Hybrid Methods[*]

Tian Xie[†], Jun Yu[‡], Tao Zeng[§]

May 30, 2020

Abstract

The data market has been growing at an exceptional pace. Consequently, more sophisticated strategies to conduct economic forecasts have been introduced with machine learning techniques. Does machine learning pose a threat to conventional econometric methods in terms of forecasting? Moreover, does machine learning present great opportunities to cross-fertilize the field of econometric forecasting? In this report, we develop a pedagogical framework that identifies complementarity and bridges between the two strands of literature. Existing econometric methods and machine learning techniques for economic forecasting are reviewed and compared. The advantages and disadvantages of these two classes of methods are discussed. A class of hybrid methods that combine conventional econometrics and machine learning are introduced. New directions for integrating the above two are suggested. The out-of-sample performance of alternatives is compared when they are employed to forecast the Chicago Board Options Exchange Volatility Index and the harmonized index of consumer prices for the euro area. In the first exercise, econometric methods seem to work better, whereas machine learning methods generally dominate in the second empirical application.

[*] We wish to thank Ding Xuan Ng, Edward Robinson and David Hardoon for helpful discussion on this topic.

[†] College of Business, Shanghai University of Finance and Economics, China. Email: xietian001@hotmail.com.

[‡] School Economics and Lee Kong Chian School of Business, Singapore Management University. Email: yujun@smu.edu.sg.

[§] School of Economics and Academy of Financial Research, Zhejiang University, Hangzhou, China. Email: ztzt6512@gmail.com.

Contents

1 Introduction

Big data, cloud computing, mobile technologies and social media, are among the most important changes in the modern era. The high-dimensional nature and the automated feature of machine learning methods make it feasible to deal with big data. Moreover, machine learning methods emphasize stable out-of-sample performance because of their ability in the regularization of model selection and the mitigation of model overfitting. Not surprisingly, sophisticated strategies have been introduced to conduct economic forecasts using machine learning techniques. Machine learning is different from traditional econometric prediction techniques which are known to be powerful to explain the financial market and macroeconomic phenomena. Consequently, the following questions naturally arise. Does machine learning pose a threat to conventional econometric methods to forecast economic activities? Or does machine learning present great opportunities to cross-fertilize the field of econometrics?

This report answers these important questions by reviewing and comparing two strands of literature: forecasting methods via econometric models and forecasting methods via machine learning techniques. We have three goals in this report. First, when reviewing the two classes of methods, special attention will be paid to identifying the strength and weakness of alternative methods. We argue that the two classes of methods differ in their purposes, focuses, and methodologies.

Moreover, we extend the literature on the economic forecast by introducing a class of hybrid methods that combine econometrics and machine learning techniques. Some of the hybrid methods include but are not restricted to the split-sample method, its model averaging extensions, and the model averaging tree methods. Some new directions on how to combining these two approaches are suggested.

Finally, we compare the performance of alternative methods using real data. In particular, we apply various methods to forecast the volatility index (VIX). In this case, we have found evidence of superior forecasting performance of conventional econometric models.

We also compare the out-of-sample performance of alternative methods when they are used to forecast the harmonized index of consumer prices (HICP) for the euro area. In this case, we have found evidence of superior forecasting performance of machine learning methods.

It is important to point out that by no mean the review of econometric methods and machine learning techniques is exhaustive. On the contrary, the choice of methods and techniques is rather selective. Our selection reflects partly the experience we have with the two strands of the literature, and also partly the popularity in their usage of economic forecasts.

The report is organized as follows. Section 2 reviews conventional econometric methods, including methods based on reduced-form models, methods based on structural models, model averaging techniques. We also review methods for variable selection, lag length selection, dimension reduction. Section 3 reviews machine learning techniques. Section 4 introduces some hybrid methods that combine conventional econometric methods and machine learning techniques. Section 5 illustrates some of the methods reviewed in both classes to forecast VIX and HICP. Section 7 concludes.

1.1 Notations and acronyms

In this paper, we adopt the following notations:

y	scalar
$\boldsymbol{x}$	vector (bold lower-case)
$\boldsymbol{X}$	matrix (bold upper-case)
$\boldsymbol{X}^\top$	transpose of matrix $\boldsymbol{X}$
$\boldsymbol{X}^{-1}$	inverse of matrix $\boldsymbol{X}$
y_t	variable y at time t
Ly_t	lag of y_t, i.e., y_{t-1}
$\mathbb{R}$	real line
$\mathbb{R}_+$	positive part of $\mathbb{R}$
$\mathbb{R}_-$	negative part of $\mathbb{R}$
$\mathbb{R}^k$	Euclidean k space
Ω	information set
$\mathbb{E}(y)$	expectation of y
$\text{Var}(y)$	variance of y
$\text{Cov}(x,y)$	covariance of x and y
$Pr(\cdot)$	probability of
$\rightarrow$	limit
$\overset{p}{\rightarrow}$	convergence in probability
$\overset{d}{\rightarrow}$	convergence in distribution
$\equiv$	definitional equality
$\sim$	distributed as

Note that all the vectors in this article are column vectors, unless otherwise indicated.

In the following list, we summarize all the acronyms along with the full terms they represent in this paper.

AIC	Akaike information criterion
AIC$_c$	finite-sample corrected Akaike information criterion
AR	autoregression
ARCH	autoregressive conditional heteroscedasticity
ARFIMA	autoregressive fractionally integrated moving average
ARIMA	autoregressive integrated moving average
ARMA	autoregressive-moving average
BAG	bootstrap aggregation
BEKK	Baba, Engle, Kraft and Kroner
BIC	Bayesian information criterion
BM	Brownian motion
BMA	Bayesian model averaging
CART	classification and regression trees
CV	cross validation
DCC	dynamic conditional correlation
DIC	deviance information criterion
DSGE	dynamic stochastic general equilibrium
FAVAR	factor-augmented vector autoregressive
fBM	fractional Brownian motion
FEVD	forecast error variance decomposition
FMA	frequentist model averaging
fOU	fractional Ornstein-Uhlenbeck process
GARCH	generalized autoregressive conditional heteroscedasticity
HAR	heterogeneous autoregressive
HQ	Hannan-Quinn
IRF	impulse response function
LASSO	least absolute shrinkage selective operator
LOOCV	leave-one-out cross validation
LSB	least squares boosting
MA	moving average
MAB	model averaging bagging
MAFE	mean absolute forecast error
MARF	model averaging random forecast
MART	model averaging tree
MARS	multivariate adaptive regression splines
MBB	moving block bootstrap
MGARCH	multivariate GARCH
MSFE	mean square forecast error
MSM	markov switching model
MSV	multivariate stochastic volatility

NARX	network with exogenous inputs
NN	neural network
PCA	principle component analysis
PCR	principle component regression
PLS	partial least squares
PLSR	partial least squares regression
PMA	predictive model average
QLIKE	Gaussian quasi-likelihood
RT	regression tree
RWMH	random walk Metropolis-Hastings
SDR	standard deviation reduction
SDFE	standard deviation of forecast error
SETAR	self-exciting threshold autoregressive
SPLT	split sample
SPX	Standard and Poor 500 index
SSM	state space model
SSR	sum of squared residuals
SVAR	structural vector autoregressive
SV	stochastic volatility
SVM	support vector machine
SVR	support vector regression
TAR	threshold autoregressive
TVC	time varying coefficient
VAR	vector autoregressive
VIX	volatility index

1.2 A Non-technical summary

Conventional time series methods assume that there is a true data generation process (DGP). According to the well-known Wold theorem, any stationary time series[1] can be expressed as an infinite order moving average (MA) process, which is a linear combination of white noises. Then, a natural thing to do is to use a finite order MA model to approximate the infinite order MA model. Alternatively we can use a finite order autoregressive (AR) model which can be expressed as an infinite order MA model but with some restrictions on the coefficients. On the other hand, one can combine AR and MA models to make the so-called Auto-regressive and Moving Average (ARMA) model. ARMA models can not capture nonlinear dynamics or long-range dependence in data.

To cope with the feature of long-range dependence which has been widely observed in economic and financial data, the autoregressive fractionally integrated moving average (ARFIMA) model extends the ARMA model by allowing non-integer values of differencing. As an alternative method to model the long-range dependence, Corsi (2009) proposed the heterogeneous autoregressive (HAR) model which can well approximate long memory and multi-scaling properties of data and easy to implement. Another alternative for capturing long-range dependence is to use a continuous-time model based on the fractional Brownian motion as shown in Wang et al. (2019).

Various models can capture nonlinear dynamics. The threshold autoregressive model (TAR) is an extension of autoregressive model with a threshold variable q_t that describes the structure change of parameters in the AR model. It assumes that the behavior of the time series changes once q_t exceeds some threshold value. If q_t is the lagged value of the series, it becomes the self-exciting TAR model (SETAR). Markov switching model (MSM) also considers the structure change of parameters. Different from TAR, the switching mechanism in MSM is controlled by a discrete unobserved state variable that follows a first-order Markov chain. Unlike the TAR and MSM models, in which the parameters change over

[1] Here, the word “stationary” means that the first and second moments of the time series are not time-varying.

time discretely, the time-varying coefficient (TVC) model allows the coefficients to change with time continuously. The TVC model can be specified in a state-space form and the likelihood function can be obtained using the Kalman filter. The structure break model specifies different patterns over different periods (not different states as in TAR, MSM and TVC models) where a structure break corresponds to an unexpected change in the parameter value. Pesaran and Timmermann (2007), Hansen et al. (2012) and Pesaran et al. (2013) discussed the forecasting performance of the structure break model.

If one is interested in forecasting volatility but only has access to prices/returns, GARCH-type models or stochastic volatility (SV) models can be used. Proposed by Engle (1982), the ARCH model assumes that the variance of the current error term is a function of lagged squared errors. Essentially the ARCH model assumes the squared return follows an AR model, whereas the GARCH model extends the ARCH model by assuming the squared return follows an ARMA model. Unlike ARCH-type models, SV models specify volatility as a separate random process, which provides certain advantages over the ARCH-type models (Kim et al., 1998). In SV models, the variance is latent and the likelihood function does not have a closed-form expression. Estimation of SV models is more difficult than that of GARCH-type models.

In practice, the number of predictors can be close to or even greater than the sample size. Such a phenomenon is called the curse of dimensionality and can cause serious problems to traditional estimation methods (for example, inconsistency). One solution to reduce the dimensionality of predictors is to use the principal component regression (PCR) or the partial least square regression (PLSR). PCR is a regression technique based on principal component analysis (PCA). It is a statistical procedure that converts a large set of possibly correlated variables into a small set of linearly uncorrelated variables (named principal components) and finds the components which can explain the variation in predictors as much as possible. Partial least squares (PLS), on the other hand, incorporates the information from the response variable to decompose the matrix of predictors.

The univariate models can be extended to a multivariate setting. A popular class of multivariate models for forecasting macroeconomic variables is reduced-form vector autoregression (VAR) models which can be regarded as the multivariate extension of AR models. Instead of regressing one single dependent variable on its lags, we regress a vector of time series variables on lagged vectors of these variables in VAR. VAR models have been proven to be particularly useful for describing the dynamic behavior of economic and financial time series and for forecasting. However, for high dimensional data, the total number of parameters in VAR can be very large and the VAR may not perform well out-of-sample. To reduce the dimensionality and to extract the information from a large number of time series, factor analysis has been widely used in practice. Factor models decompose the behavior of a high dimensional vector of economic variables into a component driven by few unobservable factors common to all the variables and variable specific idiosyncratic components.

In the conventional VAR model, the error terms are assumed to be statistical innovations. Therefore, it is impossible to identify the effect of fundamental economic shocks on the economy, such as monetary shock, technology shock, etc. A structural vector autoregression (SVAR) model makes explicit identifying assumptions to isolate estimates of the effects of fundamental shocks on the economy. Dynamic stochastic general equilibrium (DSGE) models build on explicit micro-foundations by allowing agents to do optimization. They have become very popular in macroeconomics over the last three decades. Bayesian methods have been widely applied to estimate DSGE models.

Given that many alternative models can be used to generate forecasts, it is important to know which model has the overall best performance. One of the most widely used model selection methods is the Akaike information criterion (AIC) proposed by Akaike (1973). AIC provides an asymptotically unbiased estimator of the Kullback-Leibler (K-L) divergence between the true DGP and the predictive density of the candidate model. Mallow's Cp (Mallows, 1973) provides an asymptotically unbiased estimator of the mean squared forecast error (MSFE) for a candidate model. BIC (Bayesian Information Criterion) by Schwarz (1978) takes a similar form with AIC but has the heavier penalty term than AIC.

An information criterion based on Bayesian estimators is the deviance information criterion (DIC) proposed by Spiegelhalter et al. (2002) and justified by Li et al. (2019).

Another way to evaluate the performance of a model is via the cross-validation (CV) method. A conventional validation approach is to split the data set into two parts. One part is called the "training set", which we use to estimate the model. The other part is the "validation set", which is used to evaluate the estimated model.

Model selection methods are designed to select the best model from the candidate set. The selected model is then used to forecast future economic activities. However, it is possible that the true DGP is not included in the candidate set. As a result, all candidate models are misspecified. When this occurs, a popular method to do forecast is via the model averaging technique, which averages the predictions from a collection of candidate models. Bayesian model averaging (BMA) takes prediction as an average of the predictions from different models weighted by the posterior model probabilities. Frequentist model averaging (FMA) construct model weights using information criteria such as AIC, BIC, or Mallows' Cp.

The above-mentioned time series methods, including the model averaging techniques, assume that there is one true DGP. When the DGP does not exist and when the available data is of large dimensional, it has been found that some algorithmic methods such as machine learning methods are useful. A small but growing set of studies have reported usefulness of machine learning methods in forecast economic variables; see Biau and D'elia (2010), Jung et al. (2019), and Chuku et al. (2019) in forecasting GDP growth rates, Tiffin (2016) in nowcasting GDP growth rates, and Medeiros et al. (2019) in forecasting inflation.

When a high dimensional problem is caused by a large set of input variables, as an adaptive procedure for regression, the multivariate adaptive regression splines (MARS) method excels. It is a non-parametric regression technique and can be seen as an extension of linear models that automatically model nonlinearity and interactions between variables. The model is a weighted sum of a constant, hinge function or product of hinge functions.

The hinge function usually takes the form of $\max(0, h)$ or $\max(0, h - x)$ where x is some predictor and h is the knot. MARS automatically selects variable x and values of x for knots of the hinge functions.

The placement of knots, the number of knots, and the degree of the polynomial can be seen as tuning parameters, which are subject to manipulation by a data analyst. The tuning process can be very complicated since there are at least three of them that must be tuned simultaneously. Moreover, there is little or no formal theory to justify the tuning. On the other hand, a useful alternative is to alter the fitting process itself so that the tuning is accomplished automatically, guided by clear statistical reasoning. One popular approach is to combine a penalty with the loss function to be optimized.

Strategies that are designed to control the magnitude of the coefficients are called shrinkage or regularization. Two popular methods have been offered for how to control the complexity of the fitted values. One is ridge regression which constrains the sum of the squared regression coefficients to be less than some constant. Ridge regression can create a parsimonious model when the number of predictors exceeds the number of observations, or when the predictors are highly correlated.

The other method is called the least absolute shrinkage selection operator (LASSO) by Tibshirani (1996). LASSO constraints the sum of the absolute values of the regression coefficients to be less than some constant instead. Unlike the ridge penalty, the LASSO penalty leads to a nonlinear estimator, and a quadratic programming solution is needed. The LASSO regression is capable of shrinking coefficients to 0. Therefore, it can be used as a variable selection tool in practice. Zou and Hastie (2005) pointed out that the LASSO solution paths are unstable when predictors are highly correlated. If variables are strongly correlated, LASSO is indifferent among them. Zou and Hastie (2005) proposed elastic-net as an improved version of the LASSO to overcome such limitation. Fan and Li (2001) and Zou (2006) argued that LASSO may not satisfy the oracle property, referring to a property that a method can asymptotically identify the right subset model with probability converging to 1 and has optimal estimation rate. Zou (2006) proposed the adaptive LASSO which enjoys the oracle property.

Ridge and LASSO-type regressions are linear models which cannot deal with the nonlinearity such as interaction effects. Breiman et al. (1984) proposed the Classification and Regression Trees (CART) method, in which classification mostly deals with the categorical response of non-numeric symbols and texts and regression trees focus on quantitative responses variables. Given the numerical nature of our data set, we only consider the second part of CART, regression tree (RT). The trick in applying RT is to find the best split. Consider a sample of $\{y_t, x_t\}_{t=1}^{n}$. A simple regression will yield a sum of squared residuals, SSR_0. Suppose we split the original sample into two sub-samples such that $n = n_1 + n_2$ with one of the predictors at some cut point. The RT method finds the best split of a sample (the best split variable and its cut point) to minimize the sum of squared residuals (SSR) from the two sub-samples. That is, the SSR values computed from each sub-sample should follow: $\mathrm{SSR}_1 + \mathrm{SSR}_2 \leq \mathrm{SSR}_0$. We can continue splitting until we reach a predetermined boundary. If the data are stationary and ergodic, the RT method demonstrates better forecasting accuracy. Intuitively, for cross-sectional data, the RT method performs better because it removes heterogeneity problems by splitting the sample into clusters with heterogeneous features; for time series data, a good split should coincide with jumps and structure breaks.

Bagging trees combines the bootstrap aggregation (aka bagging) methods by Breiman (1996) with RT ensembles. Bootstrap, which was introduced to statistics by Efron (1979), is the practice of estimating properties of an estimator (such as its variance) by sampling from an approximating distribution. By bootstrapping a bunch of sub-samples, fitting a regression tree to each sub-sample, and then averaging the predictions across the bootstrapped samples, we create more robust and accurate predictions than a single tree model.

Bagging trees typically suffer from a strong correlation among trees, which reduces the overall performance of the model. It is because a well-performed predictor has a high probability to be one of the most important predictors in many RTs which leads to highly correlated trees. The random forest (RF) algorithm solves this problem by randomly choosing a subset of the predictors during the splitting process for each bootstrap sample. In this way, some of those trees do not allow the same well-performed predictor to be used in the tree, hence de-correlates the RTs.

The boosting tree method is also an ensemble learning method but is fundamentally different from RF. Boosting works with the full training sample and all of the predictors. Within each iteration, the poorly fitted observations are given more weight, which eventually forces the (poor) fitting functions to evolve in boosting. We usually denote the number of iterations as the learning cycle of the boosting process. Moreover, the final output values are a weighted average over a large set of earlier fitting results instead of a simple average as in the RF method.

All decision tree algorithms discussed above base their forecasts on a set of piecewise local constant model. In fact, algorithms have been developed to estimate regression models in the leaf nodes to not just aid in prediction, but also simplify the tree model structure. That is, it is suggested that the gains in prediction from using a piecewise linear model could allow one to grow shorter trees that are more parsimonious. The M5' algorithm (Quinlan, 1992 and Wang and Witten, 1997) builds subgroups using the same algorithm as RT but a multiple regression models is estimated in the terminal node. The model in each leaf only contains the independent variables encountered in split rules in the leaf node's sub-tree and are simplified to reduce a multiplicative factor to inflate the estimated error.

In machine learning, support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The theory behind SVM is due to Vapnik (1996). The classic SVM was designed for classification and a version of SVM for regression, later known as support vector regression (SVR), was proposed by Drucker et al. (1996). The goal of SVR is to find a function that deviates from the response variable by value no greater than a predetermined ϵ for each input observation and at the same time is as flat as possible.

Most conventional econometric methods assume the DGP exists. Typical DGPs are assumed to have analytical expressions that depend on a set of unknown parameters. A "training" set is used to estimate these parameters. Such a setup allows econometricians to do a few interesting things: (1) to develop the asymptotic theory of estimators and test statistics; (2) to do counter-factual analysis and scenario analysis; (3) to obtain the impact of structure shocks via impulse response and variance decomposition; (4) the estimated DGP, with the unknown parameters being replaced with their estimates, is used to forecast in the "testing" set. If the relationships among economic variables are too complicated for an analytical function or become increasingly complicated as new data come in, then the assumption of the existence of a DGP is unrealistic. In this case, machine learning methods that only aim to predict variables may be useful. Therefore, machine learning methods are expected to pose a challenge to conventional econometric methods when prediction is the primary concern, and when the relationships among economic variables are very complicated.

However, it is possible to combine the strengths of both methods for cross-fertilization. Most machine learning techniques neglect parameter heterogeneity as they typically rely on local constant models that assume homogeneity in outcomes within individual terminal leaves. This limitation can impact their predictive ability. The presence of heterogeneity can change how the data should be partitioned thereby influencing the forecasting results. On the other hand, conventional econometric methods have provided many effective techniques to deal with heterogeneity. This sets a motivation of the need of hybrid methods.

Hirano and Wright (2017) proposed a split-sample (SPLT) method to mitigate uncertainty about the choice of predictors. They investigate the distributional properties of SPLT in a local asymptotic framework. The core of SPLT is more in the econometric tradition, which consists of splitting the training sample set into two parts, one for model selection via AIC and the other for model estimation. Moreover, the authors show that adding a bagging step to the plain SPLT substantially improves its prediction performance.

The bagging augmented SPLT method can be viewed as a hybrid of econometric and machine learning methods. Liu and Xie (2018) further extended SPLT by replacing the

AIC model selection method by the prediction model average (PMA) method developed by Xie (2015), while keeping the bagging procedure. Liu and Xie (2018) denoted this hybrid method by SPLTPMA. In SPLTPMA, after an initial sample split, a prediction model averaging technique is applied to the first subsample to obtain a weight structure over all the candidate models, and then use the weights to calculate a weighted average model as the prediction model, where each candidate model is estimated on the second subsample.

For the tree-based method, after partitioning the dataset into various subgroups, no heterogeneity is assumed within subgroups, and a simple average is computed to represent the feature in that subgroup. From the perspective of econometrics, however, this rules out heterogeneity within recursively partitioned subgroups and may appear unsatisfying. Lehrer and Xie (2018) suggested that for each tree leaf we can construct a sequence of $m = 1, \ldots, M$ linear candidate models, in which regressors of each model m is a subset of the regressors belonging to that tree leaf. The regressors $X^m_{i \in l}$ for each candidate model within each tree leaf is constructed such that the number of regressors $k^m_l \ll n_l$ for all m. Using these candidate models, they perform model averaging estimation to obtain the averaged coefficient and denote the new method as a model averaging tree (MART). Based on MART, we can apply bagging trees and random forest, which lead to model averaging bagging (MAB) and model averaging random forest (MARF).

The out-of-sample performance of alternative methods is compared when they are used to forecast Chicago Board Options Exchange's Volatility Index and the harmonized index of consumer prices for the euro area. In the first example, the traditional econometric methods work better. In the second example, the machine learning methods, and especially, the hybrid methods work better.

2 Conventional Econometric Methods

In this section, we review some classic yet still popular econometric methods for the purpose of forecasting. We consider univariate and multivariate models. In general, there are two types of econometric methods that co-exist in the forecasting literature: reduced-form models and structural models. The reduced form of a system of equations is the result of solving the system for the endogenous variables.[2] On the other hand, equations of a structural-form model are estimated in their theoretically given form.

Within the class of univariate econometric models, we review linear (predictive) regression models, autoregressive (AR) models, autoregressive-moving average (ARMA) models, autoregressive fractional integral moving average (ARFIMA) models, heterogeneous AR (HAR) models, fractional continuous-time models, threshold autoregressive models, Markov switching models, local constant regression models, local polynomial regression models, and models with structural breaks. If one is interested in forecasting volatility but only has access to prices/returns, GARCH-type models or stochastic volatility (SV) models can be used.

Most of the univariate models can be extended to a multivariate setup. For example, a popular class of multivariate models for forecasting macroeconomic variables is reduced-form VAR models which are the multivariate extension to AR models. Popular multivariate models for variance and covariance of multiple assets include multivariate GARCH models (MGARCH) and multivariate SV (MSV) models. A class of methods that are unique to the multivariate setup are factor models and their variations, for example, factor-augmented VAR (FAVAR) models.

For the structural approach, we review both structure VAR (SVAR) models and dynamic stochastic general equilibrium (DSGE) models. These two important structural econometric models have been extensively adopted by many central banks to analyze financial mar-

[2] In other words, the reduced-form of an econometric model is one that has been rearranged algebraically so that each endogenous variable is on the left side of one equation and only predetermined variables are on the right side.

kets, explain macroeconomic phenomena, and to conduct economic forecasts. Structural models are important to understand causal relationships among variables, to do simulations, and to perform scenario analysis and counter-factual analysis.

SVAR usually contains a set of equations with each equation describing the type of decision rules motivated by economic theories. One example is that consumers demand a certain quantity of aggregate output based on the aggregate price level as well as how liquid they are, with the latter being measured by real money holdings. Clearly, SVAR aims to capture how endogenous variables are related to other endogenous variables and some exogenous variables. While SVAR facilitates interpreting the data, it makes the estimation more difficult due to the presence of endogeneity.

DSGE builds on explicit micro-foundations by allowing agents to do optimizations. Earlier efforts made in the literature are the developments of estimation methodology so that the estimation of variants of DSGE models can compete with more standard time series models such as VAR models. More recent efforts have also been made to show the usefulness of these models for the purpose of forecasting economic variables.

Besides the above models, we also review some useful techniques that are closely related to the application of these models. For many time series models, the choice of lag length and covariates can be critically important for forecasting. Hence, procedures for selecting lag length and covariates are explained in details. These procedures include various information criteria and cross-validation techniques. In the era of big data, dimensionality reduced techniques become increasingly important in practice. We cover the principal component regression and partial least squares regression.

An interesting idea of carrying out economic forecasts is to acknowledge that no model is correctly specified but several models are useful. In this case, one often finds that combining alternative econometric models (model averaging) yields better economic forecasting. Important decisions in model combination include best subset selection and choice of weights. We introduce frequentist model average methods and Bayesian model average methods.

When the forecasting performance is evaluated by the mean square forecast error (MSFE), the best forecast, which minimizes the MSFE, is known to be the conditional expectation; see for example, Diebold (2006). However, when other criteria are used, the best forecast may not be the conditional expectation. Throughout the report, we denote Ω_{t-1} as an information set containing data up to period $t-1$. If we do not talk about how to estimate model parameters from data, we simply assume model parameters are known.[3] If we talk about how to estimate model parameters, we then assume the true parameters are replaced with their estimators during the forecasting exercises.

2.1 Univariate econometric models

2.1.1 Predictive regression models

Predictive regression models specifies that the variable that one hopes to predict (say y_t) depends linearly on some lagged independent variables (say $\boldsymbol{x}_t$ which is $k \times 1$ dimensional) and on an error term. The simplest predictive regression model takes the form of

$$y_t = \beta_0 + \boldsymbol{\beta}_1^\top \boldsymbol{x}_{t-1} + \epsilon_t, \tag{1}$$

where β_0 is the intercept, $\boldsymbol{\beta}_1$ a $k \times 1$ vector of slope coefficients, and $\epsilon_t \sim iid(0, \sigma^2)$.

The one-step-ahead forecast of y_T is given by

$$\mathbb{E}[y_{T+1}|\Omega_T] = \beta_0 + \boldsymbol{\beta}_1^\top \boldsymbol{x}_T. \tag{2}$$

If a multi-step-ahead forecast of y_T is needed, one needs to know the future value of $\boldsymbol{x}_t$ or to have a separate time series model for $\boldsymbol{x}_t$ so that one can forecast future value of $\boldsymbol{x}_t$. Some well-known time series models are reviewed below.

[3] When we do not talk about parameter estimation for a model, it means that this model can be estimated by a standard method.

2.1.2 Autoregressive models

The AR model specifies that the output variable depends linearly on its own lagged values and an error term. The AR model is the building block for many other time series models. The AR(p) model is defined as

$$y_t = \beta_0 + \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \epsilon_t, \tag{3}$$

where β_0 is the intercept, $\beta_1, \ldots, \beta_p$ are the slope coefficients, and $\epsilon_t \sim iid(0, \sigma^2)$. The one-step-ahead forecast of y_T is given by

$$\mathbb{E}[y_{T+1}|\Omega_T] = \beta_0 + \beta_1 y_T + \cdots + \beta_p y_{T-p+1}. \tag{4}$$

The h-step-ahead forecast of y_T can be similarly obtained.

The autoregressive model can also be extended to include exogenous variables. For example,

$$y_t = \beta_0 + \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \gamma x_{t-1} + \epsilon_t. \tag{5}$$

This model includes an extra input variable x_{t-1} compared with Model (3). The one-step-ahead forecast of y_T is given by

$$\mathbb{E}[y_{T+1}|\Omega_T] = \beta_0 + \beta_1 y_T + \cdots + \beta_p y_{T-p+1} + \gamma x_T. \tag{6}$$

2.1.3 Moving average and ARMA models

The MA model is another common approach to modeling univariate time series. An MA(q) model with a constant term can be written as

$$y_t = \mu + \epsilon_t + \alpha_1 \epsilon_{t-1} + \cdots + \alpha_q \epsilon_{t-q}, \tag{7}$$

where $\alpha_1, \ldots, \alpha_q$ are the MA coefficients. The MA process is closely related to the AR process. In fact, any stationary AR(p) process can be represented as an MA(∞) process. This close relationship between AR and MA processes goes both ways. If the MA(q) is invertible, there exists a stationary AR(∞) process to represent MA(q).

When $q = 1$, the model is MA(1) and the one-step-ahead forecast of y_T is

$$\mathbb{E}[y_{T+1}|\Omega_T] = \mu + \alpha_1 E(\epsilon_T|y_1, \cdots, y_T),$$

where $\mathbb{E}(\epsilon_T|y_1, \cdots, y_T)$ can be derived from $\{y_1, \ldots, y_T\}$ if an assumption about ϵ_0 is imposed. For example, we can assume $\epsilon_0 \approx E(\epsilon_0) = 0$. Then by backward substitutions

$$\begin{aligned}
\mathbb{E}(\epsilon_T|y_1, \cdots, y_T) &= y_T - \mu - \alpha_1 \mathbb{E}(\epsilon_{T-1}|y_1, \cdots, y_T), \\
\mathbb{E}(\epsilon_{T-1}|y_1, \cdots, y_T) &= y_{T-1} - \mu - \alpha_1 \mathbb{E}(\epsilon_{T-2}|y_1, \cdots, y_T), \\
&\vdots \\
\mathbb{E}(\epsilon_1|y_1, \cdots, y_T) &= y_1 - \mu - \alpha_1 \epsilon_0 = y_1 - \mu.
\end{aligned}$$

When no assumption about ϵ_0 is imposed, the forecast can be obtained by the Kalman filter which will be reviewed in Section 2.1.9.

In practice, however, when we model the evolution of a time series using AR or MA, we may end up with overly complicated models, as p or q can be quite large. It is therefore desirable to employ a parsimonious model that has both AR and MA components. We can write an ARMA(p, q) model as:

$$y_t = \gamma + \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \epsilon_t + \alpha_1 \epsilon_{t-1} + \cdots + \alpha_q \epsilon_{t-q}, \tag{8}$$

which simply merges (3) and (7). If the AR part is stationary or the MA part is invertible, the ARMA(p, q) process can be always be represented as either an MA(∞) process or an AR(∞) process.

2.1.4 ARFIMA models

Over the past few decades, the phenomenon of long-range dependence has been widely observed in data from economics and finance. A partial list of references include Granger and Joyeux (1980), Lo (1991), Ding et al. (1993), Baillie et al. (1996), Andersen et al. (2003) in the domain of discrete-time, and Comte and Renault (1996), Comte and Renault (1998), Aït-Sahalia and Mancini (2008), Wang et al. (2019) in the domain of continuous time.

In discrete-time, autoregressive fractionally integrated moving average models (ARFIMA) extend ARMA models by allowing a non-integer value of differencing. Let L be the lag operator. For $-0.5 < d < 0.5$, $(1-L)^d$ is defined as

$$(1-L)^d = \sum_{j=0}^{\infty} \frac{\Gamma(j-d)}{\Gamma(-d)\Gamma(j+1)} L^j,$$

with $\Gamma(\cdot)$ being the gamma function. The ARFIMA(p,d,q) model can be rewritten as

$$\left(1 - \sum_{i=1}^{p} \beta_i L^i\right)(1-L)^d y_t = \gamma + \left(1 + \sum_{i=1}^{q} \alpha_i L^i\right) \epsilon_t. \tag{9}$$

One can forecast with an ARFIMA model. For example, to forecast with ARFIMA$(1,d,0)$, one may use the forecasting formula

$$\mathbb{E}[y_{T+1}|\Omega_T] = -\sum_{j=1}^{\infty} \pi_j y_{T-j+1} \text{ where } \pi_j = \frac{(j-d-2)!}{(j-1)!(-d-1)!}\left\{1 - \beta_1 - \frac{(1+d)}{j}\right\}. \tag{10}$$

2.1.5 HAR models

Following Fernandes et al. (2014), a popular way to model the long-range dependence is the heterogeneous autoregressive (HAR) model by Corsi (2009). The HAR model gains great popularity not only because it well approximates long-range dependence and multi-scaling properties of data, but it is also very easy to implement. The standard HAR model in Corsi (2009) postulates that y_{t+1} can be modeled by

$$y_{t+1} = \beta_0 + \beta_d \bar{y}_t^{(1)} + \beta_w \bar{y}_t^{(5)} + \beta_m \bar{y}_t^{(22)} + \epsilon_{t+1}, \tag{11}$$

where $\bar{y}_t^{(l)} = l^{-1} \sum_{s=1}^{l} y_{t-s}$ is the averages of the previous l periods of y from period t, $\epsilon_t \sim iid(0, \sigma^2)$. A typical choice in the literature for the lag index vector $\boldsymbol{l}$ is $[1, 5, 22]$, to mirror the daily, weekly, and monthly components in financial markets.

The HAR model can be easily estimated and also allows for a more persistence. For example, $\boldsymbol{l}$ can be $[1, 5, 10, 22, 66]$ to include the quarterly component. We can also consider incorporating exogenous regressors $\boldsymbol{z}_t = [z_{1t}, \ldots, z_{Kt}]^\top$ into Model (11), which leads to the so-called HARX model,

$$y_{t+1} = \beta_0 + \beta_d \bar{y}_t^{(1)} + \beta_w \bar{y}_t^{(5)} + \beta_m \bar{y}_t^{(22)} + \boldsymbol{\beta}_z^\top \boldsymbol{z}_t + \epsilon_{t+1}, \tag{12}$$

where $\boldsymbol{\beta}_z$ represents the effect of $\boldsymbol{z}_t$. Note that $\boldsymbol{z}_t$ is one period before the dependent variable y_{t+1}.

2.1.6 Fractional continuous-time models

In the literature of theoretical asset pricing, some financial variables (such as interest rates and logarithmic volatility) are assumed to follow a continuous-time model specified as

$$dy_t = \mu(\kappa - y_t)dt + \sigma dW_t, \tag{13}$$

where $\mu(\kappa - y_t)$ is a drift term and W_t is a one-dimensional Brownian motion. Comte and Renault (1998) proposed to model log-volatility using a fractional Brownian motion (fBM) to replace W_t in Model (13), ensuring long memory by choosing the Hurst parameter $H > 1/2$. The fBM $(B_t^H)_{t \in \mathbb{R}}$ with the Hurst parameter $H \in (0, 1)$ is a zero-mean Gaussian process with covariance functions as

$$Cov\left(B_t^H, B_s^H\right) = \frac{1}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right), \forall t, s \in R. \tag{14}$$

When $H = 1/2$, B_t^H becomes a standard Brownian motion W_t. When W_t in Model (13) is replaced with B_t^H, we call it the fractional Ornstein-Uhlenbeck (fOU) process.

Gatheral et al. (2018) and Wang et al. (2019) demonstrated that log-volatility of equities and exchange rates behaves essentially as an fOU process where H between 0.1 and 0.2. Gatheral et al. (2018) proposed the rough fractional stochastic volatility (RFSV) model in contrast to the model by Comte and Renault (1998). When κ is close to zero, the h-step-ahead forecasting formula with the fOU process is given by

$$\mathbb{E}\left[y_{T+h}|\Omega_T\right] = \frac{\cos(H\pi)}{\pi} h^{H+1/2} \int_{-\infty}^{T} \frac{y_s}{(T-s+h)(T-s)^{H+1/2}} ds. \tag{15}$$

When a truncated discrete record is available for y_t at $t = 1, \ldots, T$, the forecasting formula becomes

$$\mathbb{E}\left[y_{T+h}|\Omega_T\right] = \frac{\cos\left(H\pi\right)}{\pi} h^{H+1/2} \frac{\sum_{s=1}^{T} \frac{y_s}{(T-s+1+h)(T-s+1)^{H+1/2}}}{\sum_{s=1}^{T}(T-s+1+h)^{-1}(T-s+1)^{-H+1/2}}. \tag{16}$$

Note that the weights are normalized to sum to one.

The RFSV model is remarkably consistent with some financial time series data and delivers promising forecasting performance. It is highly parsimonious and even more so if we fix H at 0.14 as Gatheral et al. (2018) recommended. Wang et al. (2019) proposes an estimation method for H which is easy to implement. The asymptotic theory is also developed for this estimator. When fitting Model (13) to logarithmic realized volatility of equities and exchange rates, Wang et al. (2019) finds the evidence that the estimated H is around 0.15 and H is statistically significantly less than $1/2$.

2.1.7 Threshold autoregressive models

All the econometric models discussed so far specify linear dynamic relationships. When the relationship is not linear, a nonlinear model is needed. We now review a few popular nonlinear time series models.

The threshold autoregressive model (TAR) is a widely used nonlinear time series model. TAR is usually considered as an extension of the piecewise linear regression model with structure changes occurring in the threshold space (Tong and Lim, 1980). Let us start with a simple two-regime TAR model. Following Tsay and Chen (2018), a two-regime TAR model of order k with the threshold variable q_t takes the form of

$$y_t = \begin{cases} \phi_0 + \sum_{i=1}^{k_1} \phi_i y_{t-i} + \sigma_1 \epsilon_t, & \text{if } q_{t-d} \leq r \\ \theta_0 + \sum_{i=1}^{k_2} \theta_i y_{t-i} + \sigma_2 \epsilon_t, & \text{if } q_{t-d} > r \end{cases}, \tag{17}$$

where k_1 and k_2 are the AR orders, $\epsilon_t \sim iidN(0,1)$, r is the threshold value, $d > 0$ is the time lag. If we set $q_{t-d} = y_{t-d}$, Model (17) becomes the self-exciting TAR model (SETAR).

The TAR model in (17) can be rewritten in a more compact fashion:

$$y_t = \phi_0 + \sum_{i=1}^{k_1} \phi_i y_{t-i} + \mathbb{I}\left(q_{t-d} > r\right) \left(\beta_0 + \sum_{i=1}^{k_2} \beta_i y_{t-i}\right) + e_t, \tag{18}$$

where $e_t = \left(\sigma_1 + \sigma_2 \mathbb{I}\left(q_{t-d} > r\right)\right) \epsilon_t$ and the coefficient $\beta_i = \theta_i - \phi_i$ captures the structure change of the parameters.

Predicting with the TAR model can be obtained via simulations. Define

$$e_t = y_t - \phi_0 - \sum_{i=1}^{k} \phi_i y_{t-i} - \beta_0 c_t^{(b)} - \sum_{i=1}^{k} \beta_i y_{t-i}^{(b)}.$$

Algorithm 1 contains details of how to obtain prediction with TAR.

Algorithm 1 *h*-step-ahead forecast of TAR

```
1. For $m = 1, 2, \ldots, M$ :
    (a) Generate random samples $(e_{T+1}^{(m)}, e_{T+2}^{(m)}, \ldots, e_{T+h}^{(m)})$ from $(e_1, e_2, \ldots, e_T)$;
    (b) Get $(\hat{y}_{T+1,T}^{(m)}, \hat{y}_{T+2,T}^{(m)}, \ldots, \hat{y}_{T+h,T}^{(m)})$ recursively from the TAR model in (18).
2. Then we get the $h$-step-ahead forecast as
    $\hat{y}_{T+h,T} = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_{T+h,T}^{(m)}.$
```

2.1.8 Markov switching models

The Markov switching model (MSM) also describes the structure change of parameters. Following Ghysels and Marcellino (2018), we use a simple example to illustrate the difference between TAR and MSM. Let

$$y_t = (\phi_{01} + \phi_{11} y_{t-1}) S_t + (\phi_{02} + \phi_{12} y_{t-1}) (1 - S_t) + \epsilon_t, \tag{19}$$

where $\epsilon_t \sim iidN\left(0, \sigma^2\right)$ and $S_t \in \{0, 1\}$ is the state variable. In Model (19), the values of parameter change with the state variable. If S_t is determined by observed variables and threshold values, it is equivalent to a TAR model. If S_t is unobserved and follows a Markov chain, Model (19) becomes a MSM. A two-state autoregressive MSM is

$$y_t = \begin{cases} \phi_{0,0} + \sum_{j=1}^{k} \phi_{j,0} y_{t-j} + \sigma_0 \epsilon_t & \text{if } S_t = 0 \\ \phi_{0,1} + \sum_{j=1}^{k} \phi_{j,1} y_{t-j} + \sigma_1 \epsilon_t & \text{if } S_t = 1 \end{cases}, \tag{20}$$

where $\epsilon_t \sim iidN(0,1)$. The state transition of the model is governed by the transition probabilities

$$\begin{aligned} Pr(S_t = 1|S_{t-1} = 0) = \eta_0, \quad Pr(S_t = 0|S_{t-1} = 1) = \eta_1, \\ Pr(S_t = 0|S_{t-1} = 0) = 1 - \eta_0, \quad Pr(S_t = 0|S_{t-1} = 1) = 1 - \eta_1, \end{aligned} \tag{21}$$

where $0 < \eta_j < 1$. We can define the transition probability matrix as

$$\boldsymbol{P} = \begin{bmatrix} 1-\eta_0 & \eta_0 \\ \eta_1 & 1-\eta_1 \end{bmatrix}. \tag{22}$$

With MSM, we can make inference about the state variable with the filtering probability of S_t, $\Pr(S_t = i|\boldsymbol{y}^t, \boldsymbol{\theta})$, at time t, where $\boldsymbol{\theta} = (\boldsymbol{\phi}, \sigma_0^2, \sigma_1^2, \eta_0, \eta_1)^\top$ with $\boldsymbol{\phi} = \{\phi_{i,j}\}$ and $\boldsymbol{y}^t = (y_1, y_2, \ldots, y_t)^\top$. Following Tsay and Chen (2018), the one-step-ahead prediction probability is computed with the filtering probability as

$$\begin{aligned} Pr\left(S_t = i|\boldsymbol{y}^{t-1}, \boldsymbol{\theta}\right) &= \sum_{j=0,1} Pr\left(S_t = i|S_{t-1} = j, \boldsymbol{y}^{t-1}, \boldsymbol{\theta}\right) Pr\left(S_{t-1} = j|\boldsymbol{y}^{t-1}, \boldsymbol{\theta}\right) \\ &= \sum_{j=0,1} Pr(S_t = i|S_{t-1} = j)\, Pr\left(S_{t-1} = j|\boldsymbol{y}^{t-1}, \boldsymbol{\theta}\right), \end{aligned} \tag{23}$$

for $j = 0, 1$. The filtering probability can be recursively estimated by the one-step-ahead prediction probability

$$\Pr\left(S_t = j|\boldsymbol{y}^t, \boldsymbol{\theta}\right) = \frac{\Pr\left(y_t|S_t = j, y^{t-1}, \boldsymbol{\theta}\right) \Pr\left(S_t = j|\boldsymbol{y}^{t-1}, \boldsymbol{\theta}\right)}{\Pr\left(y_t|\boldsymbol{y}^{t-1}, \boldsymbol{\theta}\right)}. \tag{24}$$

Similar to TAR, predicting with MSM can be obtained via simulations. Algorithm 2 contains details of how to obtain prediction with MSM.

Algorithm 2 h-step-ahead forecast of MSM

```
1. For $m = 1, 2, \ldots, M$ :
    (a) Generate random samples $(\hat{\epsilon}_{T+1}^{(m)}, \hat{\epsilon}_{T+2}^{(m)}, \ldots, \hat{\epsilon}_{T+h}^{(m)})$ from the distribution of $\epsilon_t$;
    (b) Draw the state $S_T^{(m)} = v^{(m)}$ using the filtered state probability $Pr\left(S_T^{(m)} = j|\boldsymbol{y}^T, \boldsymbol{\theta}\right)$;
    (c) For $i = 1, \ldots, h$ :
        i. Conditioned on $S_{T+i-1}^{(m)}$, draw the state $S_{T+i}^{(m)}$ using the transition probability matrix $\boldsymbol{P}$;
        ii. Compute $\hat{y}_{T+i,T}^{(m)}$ with $S_{T+i}^{(m)}$.
2. Then we get the $h$-step-ahead forecast as
        $\hat{y}_{T+h,T} = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_{T+h,T}^{(m)}.$
```

2.1.9 Time-varying coefficient model

The coefficients in the time-varying coefficient (TVC) model change with time. In practice, the TVC model is usually specified in the state-space. A classic example of TVC model is the unobserved component model as in Harvey (1990). Let

$$\begin{aligned} y_t &= \mu_t + \xi_t, \\ \mu_t &= \mu_{t-1} + \eta_t, \end{aligned} \tag{25}$$

where y_t is the GDP (Gross Domestic Product), μ_t is the trend component of GDP, ξ_t follows the random walk process, and the error term η_t is uncorrelated with ξ_t. In Model (25), y_t is observed but μ_t is not. We usually denote μ_t as the state variable. Model (25) is a state-space model (SSM). The first equation of (25) is called the space equation while the second equation the state equation. The Kalman filter (Kalman, 1960) provides a means to forecast both the observed and state variables. It can also be used to calculate the observed-data likelihood function, which can then be used to obtain the maximum likelihood estimator of parameters.

Consider the following general linear Gaussian SSM

$$\begin{aligned} y_t &= \boldsymbol{A}^\top \boldsymbol{x}_t + \boldsymbol{H}^\top \boldsymbol{\xi}_t + w_t, & (26)\\ \boldsymbol{\xi}_t &= \boldsymbol{F}\boldsymbol{\xi}_{t-1} + \boldsymbol{v}_t, & (27)\\ \begin{pmatrix} w_t \\ \boldsymbol{v}_t \end{pmatrix} &\sim iidN\left[\boldsymbol{0}, \begin{pmatrix} R & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} \end{pmatrix}\right], & (28) \end{aligned}$$

where $y_t, \boldsymbol{x}_t$ are observables with $\boldsymbol{x}_t$ being the exogenous variables, $\boldsymbol{\xi}_t$ is the unobserved state variable, and $\boldsymbol{A}, \boldsymbol{H}, \boldsymbol{F}$ are matrices of parameters.

In fact, many models can be presented as a SSM model. For example, for the AR(p) model

$$y_t - \mu = \phi_1(y_{t-1} - \mu) + \cdots + \phi_p(y_{t-p} - \mu) + \epsilon_t,$$

if we define

$$\boldsymbol{\xi}_t = \begin{bmatrix} y_t - \mu \\ \vdots \\ y_{t-p+1} - \mu \end{bmatrix}, \boldsymbol{F} = \begin{bmatrix} \phi_1 & \cdots & \phi_{p-1} & \phi_p \\ 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}, \boldsymbol{v}_t = \begin{bmatrix} \epsilon_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \boldsymbol{Q} = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & \cdots & 0 \end{bmatrix},$$

then $\boldsymbol{x}_t = 1$, $\boldsymbol{A}^\top = \mu$, $\boldsymbol{H}^\top = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$, $w_t = 0$, $R = 0$ in the corresponding SSM representation. Taking the following MA(1) model for another example,

$$y_t = \mu + \epsilon_t + \theta\epsilon_{t-1},$$

if we define $\boldsymbol{\xi}_t = \begin{bmatrix} \epsilon_t \\ \epsilon_{t-1} \end{bmatrix}$, $\boldsymbol{F} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, $\boldsymbol{v}_t = \begin{bmatrix} \epsilon_t \\ 0 \end{bmatrix}$, $\boldsymbol{Q} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 0 \end{bmatrix}$, then $\boldsymbol{x}_t = 1$, $\boldsymbol{A}^\top = \mu$, $\boldsymbol{H}^\top = \begin{bmatrix} 1 & \theta \end{bmatrix}$, $w_t = 0$, $R = 0$ in the corresponding SSM representation.

In general, the likelihood function for SSM can be written as

$$p(y_1, ..., y_T | x_1, ..., x_T) = p(y_1|x_1) \prod_{t=2}^{T} p(y_t|\boldsymbol{x}_t, \Omega_{t-1}).$$

We define

$$\begin{aligned} \hat{\boldsymbol{\xi}}_{t|s} &= \mathbb{E}(\boldsymbol{\xi}_t|\Omega_s), & (29) \\ \boldsymbol{\Sigma}_{t|s} &= \mathbb{E}\left[\left(\boldsymbol{\xi}_t - \boldsymbol{\xi}_{t|s}\right)\left(\boldsymbol{\xi}_t - \boldsymbol{\xi}_{t|s}\right)^\top \middle| \Omega_s\right]. & (30) \end{aligned}$$

Clearly, $\hat{\boldsymbol{\xi}}_{t+1|t}$ is the predictor of $\boldsymbol{\xi}_{t+1}$. $\hat{\boldsymbol{\xi}}_{t|t}$ is called the filter of $\boldsymbol{\xi}_t$ while $\hat{\boldsymbol{\xi}}_{t|T}$ is called the smoother of $\boldsymbol{\xi}_t$. Denote $y_{t|t-1} := y_t|\boldsymbol{x}_t, \Omega_{t-1}$ and $\hat{y}_{t|t-1} := \mathbb{E}(y_t|\boldsymbol{x}_t, \Omega_{t-1})$. The Kalman filter is illustrated in Algorithm 3. Note that the observed-data likelihood at period t may be obtained from Step 2. If the observed-data likelihood is maximized over the parameter space, one obtains the maximum likelihood estimator.

2.1.10 Local constant regression models

So far, we have assumed that y_t is a parametric function of lagged values of y_t, such as $(y_{t-1}, y_{t-2}, \ldots, y_{t-k})$,

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-k}) + \epsilon_t. \tag{31}$$

with initial values y_0. It is straightforward to show that

$$\mathbb{E}[y_t|y_{t-1}, y_{t-2}, \ldots, y_{t-k}] = f(y_{t-1}, y_{t-2}, \ldots, y_{t-k}).$$

Denote $\boldsymbol{X}_{t-1} = (y_{t-1}, y_{t-2}, \ldots, y_{t-k})^\top$ and the initial vector $\boldsymbol{X}_{k-1} = (y_{k-1}, y_{k-2}, \ldots, y_0)^\top$. We can rewrite (31) as

$$y_t = f(\boldsymbol{X}_{t-1}) + \epsilon_t. \tag{32}$$

In this section, we introduce several nonparametric methods to estimate $f(\cdot)$ where the functional form of f is determined by data. Our discussion mainly follows Härdle et al.

Algorithm 3 Kalman filter

1. Set $\boldsymbol{\xi}_{1|0}$ to be distributed as the unconditional distribution so that $\hat{\boldsymbol{\xi}}_{1|0} = \mathbb{E}(\boldsymbol{\xi}_1)$, and $\Sigma_{1|0} = \mathbb{E}[(\boldsymbol{\xi}_1 - \mathbb{E}(\boldsymbol{\xi}_1))(\boldsymbol{\xi}_1 - \mathbb{E}(\boldsymbol{\xi}_1))^\top]$.

2. Then $y_1|\boldsymbol{x}_1 = \boldsymbol{A}^\top \boldsymbol{x}_1 + \boldsymbol{H}^\top \boldsymbol{\xi}_{1|0} + w_1$ so that

$$\begin{aligned} \hat{y}_{1|0} &= \mathbb{E}(y_1|\boldsymbol{x}_1) = \boldsymbol{A}^\top \boldsymbol{X}_1 + \boldsymbol{H}^\top \hat{\boldsymbol{\xi}}_{1|0}, \\ \mathbb{E}[(y_1 - \hat{y}_{1|0})^2] &= \boldsymbol{H}^\top \boldsymbol{\Sigma}_{1|0} \boldsymbol{H} + R. \end{aligned}$$

3. Let

$$\begin{aligned} \hat{\boldsymbol{\xi}}_{1|1} &= \mathbb{E}(\boldsymbol{\xi}_1|y_1, \boldsymbol{x}_1) \\ &= \mathbb{E}(\boldsymbol{\xi}_1|\boldsymbol{x}_1) + \mathbb{E}[(\boldsymbol{\xi}_1 - \hat{\boldsymbol{\xi}}_{1|0})(y_1 - \hat{y}_{1|0})] \\ &\quad \times \{E[(y_1 - \hat{y}_{1|0})^2]\}^{-1} \times (y_1 - \widehat{y}_{1|0}) \\ &= \boldsymbol{\xi}_1|\boldsymbol{x}_1 + \boldsymbol{\Sigma}_{1|0}\boldsymbol{H}(\boldsymbol{H}^\top \boldsymbol{\Sigma}_{1|0}\boldsymbol{H} + R)^{-1}(y_1 - \boldsymbol{A}^\top \boldsymbol{x}_1 - \boldsymbol{H}^\top \hat{\boldsymbol{\xi}}_{1|0}). \end{aligned}$$

   The associated $\boldsymbol{\Sigma}_{1|1} = \boldsymbol{\Sigma}_{1|0} - \boldsymbol{\Sigma}_{1|0}\boldsymbol{H}(\boldsymbol{H}^\top \boldsymbol{\Sigma}_{1|0}\boldsymbol{H} + R)^{-1}\boldsymbol{H}^\top \boldsymbol{\Sigma}_{1|0}$.

4. Then $\hat{\boldsymbol{\xi}}_{2|1} = \mathbb{E}(\boldsymbol{\xi}_2|y_1, \boldsymbol{x}_1) = \boldsymbol{F}\mathbb{E}(\boldsymbol{\xi}_1|y_1, X_1) = \boldsymbol{F}\hat{\boldsymbol{\xi}}_{1|1}$. The associated $\boldsymbol{\Sigma}_{2|1} = \boldsymbol{F}\boldsymbol{\Sigma}_{1|1}\boldsymbol{F}^\top + \boldsymbol{Q}$.

5. Repeat Steps 2-4 by rolling the sample forward.

6. To get the smoother, calculate

$$\begin{aligned} \hat{\boldsymbol{\xi}}_{t|T} &= \hat{\boldsymbol{\xi}}_{t|t} + \boldsymbol{J}_t(\boldsymbol{\xi}_{t+1} - \hat{\boldsymbol{\xi}}_{t+1|t}), \\ \boldsymbol{\Sigma}_{t|T} &= \boldsymbol{\Sigma}_{t|t} + \boldsymbol{J}_t(\boldsymbol{\Sigma}_{t+1|T} - \boldsymbol{\Sigma}_{t+1|t})\boldsymbol{J}_t^\top, \end{aligned}$$

   where $\boldsymbol{J}_t = \boldsymbol{\Sigma}_{t|t}\boldsymbol{F}^\top \boldsymbol{\Sigma}_{t+1|t}^{-1}$.

7. To obtain out-of-sample forecasts, calculate

$$\begin{aligned} \hat{\boldsymbol{\xi}}_{T+h|T} &= \boldsymbol{F}^h \hat{\boldsymbol{\xi}}_{T|T}, \\ \hat{y}_{T+h|T} &= \boldsymbol{A}^\top \boldsymbol{x}_{T+h} + \boldsymbol{H}^\top \boldsymbol{F}^h \hat{\boldsymbol{\xi}}_{T|T}. \end{aligned}$$

(1997) and Tsay and Chen (2018).

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$ be a k-dimensional realization of $\boldsymbol{X}_{t-1}$. Note that the conditional mean of y_t, i.e. $\mathbb{E}\left(y_t | \boldsymbol{X}_{t-1} = \boldsymbol{x}\right) = f(\boldsymbol{x})$ is the quantity of interest before a prediction is made. The sample can be rewritten as $((y_k, \boldsymbol{X}_{k-1}), (y_{k+1}, \boldsymbol{X}_k), \ldots, (y_T, \boldsymbol{X}_{T-1}))$. Denote the neighborhood of $\boldsymbol{x}$ as $N_i(\boldsymbol{x}) = \{t | \|\boldsymbol{X}_{t-1} - \boldsymbol{x}\| < h_i, t = k, \ldots, T\}$, where h_i is a given positive real number and $\| \cdot \|$ stands for the Euclidean norm. The term $N_i(\boldsymbol{x})$ consists of the time index of past k-dimensional vectors $\boldsymbol{X}_{t-1}$ that are in the h_i-neighborhood of $\boldsymbol{x}$.

Suppose f is continuous. A simple estimator of $\hat{f}$ is the sample mean of y_t's in $N_i(\boldsymbol{x})$ such that

$$\hat{f}\left(\boldsymbol{x}\right) = \frac{1}{\#N_i(\boldsymbol{x})} \sum_{t \in N_i(\boldsymbol{x})} y_t, \tag{33}$$

where $\#N_i(\boldsymbol{x})$ denotes the number of the k-dimensional vectors in $N_i(\boldsymbol{x})$. Equation (33) represents a kernel estimation of $f(\cdot)$ with a uniform kernel. The uniform kernel takes the form

$$K_0(u) = \frac{1}{2}\mathbb{I}(|u| \leq 1). \tag{34}$$

The Gaussian kernel is another popular choice:

$$K_\phi(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right). \tag{35}$$

There are other kernel functions that have been shown to be versatile in practice. To estimate f, Robinson (1983), Auestad and Tjøstheim (1990), Härdle and Vieu (1992) proposed a Nadaraya-Watson estimator

$$\hat{f}\left(\boldsymbol{x}\right) = \frac{\sum_{t=k+1}^{T} \prod_{i=1}^{k} K\left[\left(x_i - y_{t-i}\right)/h_i\right] y_t}{\sum_{t=k+1}^{T} \prod_{i=1}^{k} K\left[\left(x_i - y_{t-i}\right)/h_i\right]}, \tag{36}$$

where $K(\cdot)$ is a chosen kernel and h_i is the bandwidth for the i^{th} lagged variable.

In Equation (33), $\hat{f}\left(\boldsymbol{x}\right)$ is obtained by taking an average of specific y_t observations in the neighborhood of $\boldsymbol{x}$, henceforth the name, local constant regression. One alternative is to use polynomial regression on the observations in the same neighborhood, which gives rise to the so-called local polynomial regression. For the sake of notational simplicity, we assume $\boldsymbol{x}$ is one dimensional.

The concept of local polynomial regression is to perform the Taylor expansion of $f(\boldsymbol{z})$ at $\boldsymbol{x}$ such that

$$f(\boldsymbol{z}) \approx \sum_{j=0}^{m} \frac{f^{(j)}(\boldsymbol{x})}{j!}(\boldsymbol{z}-\boldsymbol{x})^{j} := \sum_{j=0}^{m} \beta_j(\boldsymbol{z}-\boldsymbol{x})^{j}, \tag{37}$$

where $f^{(j)}$ denotes the j^{th} derivative of f. Here β_j is estimated by minimizing the objective function of a locally weighted polynomial regression using all the points in the neighborhood of $\boldsymbol{x}$:

$$L(\boldsymbol{\beta}) = \sum_{t \in N_x} \left[y_t - \sum_{j=0}^{m} \beta_j \left(\boldsymbol{x}_{t-1}-\boldsymbol{x}\right)^{j} \right]^2 K\left(\frac{\boldsymbol{x}_{t-1}-\boldsymbol{x}}{h}\right), \tag{38}$$

where N_x denotes the neighborhood of $\boldsymbol{x}$ with bandwidth h and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^\top$. Note that if $m = 0$, Model (37) is simply a local constant regression. If $m = 1$, Model (38) can be rewritten as

$$L(\boldsymbol{\beta}) = \sum_{t \in N_x} \left[y_t - \beta_0 - \beta_1 \left(\boldsymbol{x}_{t-1}-\boldsymbol{x}\right)\right]^2 K\left(\frac{\boldsymbol{x}_{t-1}-\boldsymbol{x}}{h}\right), \tag{39}$$

which is a weighted least squares estimator of β_0 and β_1 in the neighborhood of x. The estimator $\hat{\beta}_0$ and $\hat{\beta}_1$ Model (39) is called the local linear regression.

Under the assumption that the underlying $f(\boldsymbol{x})$ is continuously differentiable, the local polynomial estimator is superior to the kernel estimator with smaller bias, faster convergence rate, and smaller mini-max risk (Tsay and Chen, 2018). It also performs better on the boundary of the observed data. Unfortunately, both the kernel estimator and the local polynomial estimator subject to the curse-dimensionality problem when k takes a moderate or large values, making them difficult to implement in a data-rich environment.

2.1.11 Models with a structure break

Assume the DGP for $(y_t, \boldsymbol{X}_t^\top)$ for $t = 1, 2, \ldots, T$ is

$$\begin{aligned} y_t &= \boldsymbol{\beta}_1^\top \boldsymbol{X}_t + \epsilon_t, \qquad t \leq T_b, \\ y_t &= \boldsymbol{\beta}_2^\top \boldsymbol{X}_t + \epsilon_t, \qquad t > T_b, \end{aligned} \tag{40}$$

where y_t is a one dimensional response variable, $\boldsymbol{X}_t$ is a $k \times 1$ independent variable, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are both $k \times 1$ parameter vectors, ϵ_t is the error term with $\epsilon_t \sim iid(0, \sigma^2)$, and T_b is the break date. For convenience, we define $\tau = T_b / T$ as the break date fraction. The pre-break sample consists of the data in periods $t = 1, 2, \ldots, T_b$, while the remaining data define the post-break sample.

Following Hansen (2012), Model (40) can be rewritten in the following compact form

$$y_t = \boldsymbol{\beta}_1^\top \boldsymbol{X}_t \mathbb{I}\left(t \leq T_b\right) + \boldsymbol{\beta}_2^\top \boldsymbol{X}_t \mathbb{I}\left(t > T_b\right) + \epsilon_t. \tag{41}$$

Assume T_b is known and we want to test the null hypothesis that no structure break exists, that is,

$$H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2. \tag{42}$$

Chow (1960) proposed a F test for H_0. Note that if $\boldsymbol{X}_t = y_{t-1}$ and there is no prior information about the size of β_1, Jiang et al. (2019) demonstrated that the OLS estimator of β_1 follows a different asymptotic distribution, which can be used in testing the timing of the structure break.

2.1.11.1 Forecasting with breaks Following Hansen (2012), forecast based on structure break model should focus on the final break date since forecasting concentrates on the behavior of data in the future not in the past. Hansen (2012) proposed a procedure for forecasting after structure break. First, we test for the existence of breaks with Andrews' sup-F test. If there exists breaks, we estimate the break dates first, then forecast with the data after the final break date.

On the other hand, Pesaran and Timmermann (2007) pointed out that forecast based on the post-break period may not be optimal due to the well-known bias-variance tradeoff. That is, we may reduce the (forecast) variance by using some pre-break data in the cost of potentially increasing bias. Provided the break is not too large, pre-break data can be informative for forecasting outcomes even after the break.

Following Pesaran and Timmermann (2007), let m denote the starting point of the sample of the most recent observations to be used in estimation for the purpose of forecasting y_{T+1} based on (41) and information Ω_T. Denote $\boldsymbol{X}_{m,T}$ as the $(T-m+1)\times p$ matrix of observations on the regressors such that $rank\left(\boldsymbol{X}_{m,T}\right) = p$, while $\mathbf{Y}_{m,T}$ is the $(T-m+1)\times 1$ vector of observations on the dependent variable. Defining the quadratic form $\mathbf{Q}_{\tau,T_i} = \boldsymbol{X}_{\tau,T_i}^{\top}\boldsymbol{X}_{\tau,T_i}$ so that $\mathbf{Q}_{\tau,T_i} = 0$ if $\tau > T_i$, the OLS estimator of $\boldsymbol{\beta}$ based on the sample from m to $T(m < T-p+1)$ is given by

$$\widehat{\boldsymbol{\beta}}_T(m) = \mathbf{Q}_{m,T}^{-1}\boldsymbol{X}_{m,T}^{\top}\mathbf{Y}_{m,T}. \tag{43}$$

Then, the one-step-ahead forecasting error is

$$e_{T+1}(m) = y_{T+1} - \widehat{y}_{T+1} = \left(\boldsymbol{\beta}_2 - \widehat{\boldsymbol{\beta}}_T(m)\right)^{\top}\boldsymbol{X}_T + \epsilon_{T+1}. \tag{44}$$

From (44), the MSFE is $\mathbb{E}\left[e_{T+1}^2(m)\right]$.

Under some regularity conditions, Pesaran and Timmermann (2007) showed that the optimal pre-break window that minimizes the MSFE gets longer if (i) the signal-to-noise ratio ω/σ^2 is smaller; (ii) the size of the break $(\beta_1-\beta_2)^2$ is smaller; (iii) the post-break sample size is smaller.

Instead of using the post-break window, Pesaran and Timmermann (2007) proposed to use a cross-validation method for selecting the optimal window. Let the pseudo-out-of-sample MSFE be

$$\text{MSFE}(m|T,\tilde{\omega}) = \tilde{\omega}^{-1} \sum_{\tau=T-\tilde{\omega}}^{T-1} \left(y_{\tau+1} - \boldsymbol{X}_{\tau}^{\top} \hat{\boldsymbol{\beta}}_{m:\tau} \right)^2, \tag{45}$$

where $\tilde{\omega}$ is the number of the last observations held out for prediction and $\hat{\boldsymbol{\beta}}_{m:\tau}$ is the OLS estimate based on the observation window $[m,\tau]$. Let $\hat{T}_b$ be the estimate of break date and $\underline{\omega}$ be the minimum number of observations needed for estimation. The optimal window size m^* is defined as

$$m^*\left(T,\hat{T}_1,\underline{\omega},\tilde{\omega}\right) = \underset{m=1,\ldots,\min\left(\hat{T}_1+1,T-\underline{\omega}-\tilde{\omega}\right)}{\arg\min} \text{MSFE}(m|T,\tilde{\omega}). \tag{46}$$

The forecasts of y_{T+1} is then

$$\hat{y}_{T+1}\left(T,\hat{T}_1,m^*\right) = \boldsymbol{X}_T^{\top}\hat{\boldsymbol{\beta}}_{m^*:T}. \tag{47}$$

Pesaran and Timmermann (2007) also proposed averaging forecasts across estimation windows for y_{T+1} based on MSFE,

$$\hat{y}_{T+1,W}\left(T,\hat{T}_1,\tilde{\omega}\right) = \frac{\sum_{m=1}^{\hat{T}_1+1}\left(\boldsymbol{X}_T^{\top}\hat{\boldsymbol{\beta}}_{m:T}\right)\text{MSFE}(m|T,\tilde{\omega})}{\sum_{m=1}^{\hat{T}_1+1}\text{MSFE}(m|T,\tilde{\omega})}. \tag{48}$$

If the break date is unknown, it can be shown that

$$\hat{y}_{T+1,W}(T,\underline{\omega},\tilde{\omega}) = \frac{\sum_{m=1}^{T-\underline{\omega}-\tilde{\omega}}\left(\boldsymbol{X}_T^{\top}\hat{\boldsymbol{\beta}}_{m:T}\right)\text{MSFE}(m|T,\tilde{\omega})}{\sum_{m=1}^{T-\underline{\omega}}\text{MSFE}(m|T,\tilde{\omega})}. \tag{49}$$

Pesaran and Pick (2011) showed that the model averaging method improves forecasts without relying on estimates of break dates and size.

Pesaran et al. (2013) proposed to forecast y_{T+1} based on weighted observations and derived the weights that minimizes the MSFE of the resulting forecast. For model (41), the estimator of slope parameter over the whole weighted sample is

$$\hat{\boldsymbol{\beta}}_T(\mathbf{w}) = \left(\sum_{t=1}^{T} w_t \mathbf{x}_t \boldsymbol{X}_t^\top\right)^{-1} \sum_{t=1}^{T} w_t \boldsymbol{X}_t \mathbf{y}_t, \tag{50}$$

where $\sum_{t=1}^{T} w_t = 1$. Then, the one-step-ahead forecasting error is

$$e_{T+1}(\mathbf{w}) = y_{T+1} - \widehat{y}_{T+1} = \left(\boldsymbol{\beta}_2 - \widehat{\boldsymbol{\beta}}_T(\mathbf{w})\right)^\top \boldsymbol{X}_T + \epsilon_{T+1}, \tag{51}$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_T)$ and the scaled MSFE can be defined as

$$\text{MSFE}_s(\mathbf{w}) = \text{E}\left[\sigma^{-2} e_{T+1}^2(\mathbf{w}) | \boldsymbol{X}_t, t = 1, 2, \ldots, T+1\right]. \tag{52}$$

The optimal weights can then be estimated by

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \text{MSFE}_s(\mathbf{w}) \tag{53}$$

subject to $\sum_{t=1}^{T} w_t = 1$. In general, (53) has to be solved numerically. Once we obtain $\mathbf{w}^*$, the forecast of y_{T+1} is simply

$$\hat{y}_{T+1}\left(T, \hat{T}_b, \mathbf{w}^*\right) = \boldsymbol{X}_T^\top \hat{\boldsymbol{\beta}}(\mathbf{w}^*). \tag{54}$$

Pesaran et al. (2013) proved that the optimal averaged forecast can achieve smaller MSFE than those by the post-break window method and the cross-validation method in Pesaran and Timmermann (2007).

2.1.12 GARCH models

Perhaps the most famous model that describes volatility is the generalized autoregressive conditional heteroskedasticity (GARCH) model of Bollerslev (1986). As the name indicates, GARCH is the generalized version of the autoregressive conditional heteroskedasticity (ARCH) model. Proposed by Engle (1982), the ARCH model assumes the variance of the current error term as a function of lagged squared errors. Essentially, the ARCH model assumes the squared return follows an AR model, whereas the GARCH model extends the ARCH model by assuming the squared return follows the ARMA model.

Let y_t denote the return of an asset at time t (with the unconditional mean removed). Then, an ARCH(q) process can be written as

$$y_t = \sigma_t \epsilon_t; \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i y_{t-i}^2,$$

where $\alpha_i > 0$ for all i, and $\epsilon_t \sim iid(0,1)$. The GARCH(p,q) process is

$$y_t = \sigma_t \epsilon_t; \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i y_{t-i}^2 + \sum_{i=1}^{p} \beta_j \sigma_{t-j}^2,$$

where $\alpha_i > 0$, $\beta_j > 0$ for all i, j. GARCH models are usually estimated by the maximum likelihood method.

There are various extensions on GARCH. For example, the integrated GARCH imports a unit root, hence requiring $\sum_{i=1}^{q} \alpha_i + \sum_{i=1}^{p} \beta_j = 1$; the exponential GARCH by Nelson (1991) focuses on $\ln \sigma_t^2$ instead of σ_t^2, hence imposing no sign restrictions for parameters; GARCH-in-mean model adds the conditional variance or standard deviation into the mean equation; and the Quadratic GARCH model by Sentana (1995) allows for asymmetric effects of positive and negative shocks.

2.1.13 Stochastic volatility models

Unlike ARCH-type models, stochastic volatility (SV) models specify volatility as a separate random process, which provides certain advantages over the ARCH-type models for modeling the dynamics of asset returns (Kim et al., 1998). The basic SV model specifies

$$y_t = \sigma_t \epsilon_t; \qquad \ln \sigma_t^2 = \alpha + \phi \ln \sigma_{t-1}^2 + \sigma_v v_t, \tag{55}$$

u_t and $v_t \sim iidN(0,1)$ and $corr(u_t, v_{t+1}) = 0$. The ARCH class of models assumes that the variance is a simple function of lagged "news". Alternatively, in the SV model it is assumed that the log-variance is itself a stochastic process. The continuous time version of this type of model has been widely used to price options, for example, Hull and White (1987) and Heston (1993).

In the SV model, the variance is latent and the likelihood function does not have a closed-form expression. Consequently, the maximum likelihood estimation of the SV model is more difficult than that of the ARCH-type models. However, with the development of effective estimation methods in recent years, the difficulties in estimating SV models has disappeared; see the survey by Shephard (2005).

The basic SV model may be approximated by a SSM for which the Kalman filter technique can be applied. This estimation method, originally suggested by Harvey et al. (1994), is termed the quasi-maximum likelihood method. The SSM that is used to approximate the SV model has the expression:

$$\begin{aligned} \ln y_t^2 &= -1.27 + \ln \sigma_t^2 + e_t, & e_t &\sim iidN(0, \pi^2/2), \\ \ln \sigma_t^2 &= \alpha + \phi \ln \sigma_{t-1}^2 + \sigma_v v_t, & v_t &\sim iidN(0,1). \end{aligned}$$

Clearly one can obtain forecast of $\ln(\sigma_t^2)$ as shown in Yu (2002).

Many statistically more efficient estimation methods and more complicated SV specifications have been proposed in the literature. Examples include Jacquier et al. (1994), Kim et al. (1998), Yu (2005), Jacquier et al. (2004), Yu (2012). For a review of SV models, see Shephard (2005).

One estimation method, which also provides forecast of σ_t^2 as a by-product, is based on Bayesian Markov chain Monte Carlo (MCMC). Various MCMC algorithms have been proposed to sample from the posterior distributions of the parameters in the context of the basic SV model. An early example is the single-move Metropolis-Hastings (MH) algorithm developed by Jacquier et al. (1994). To achieve better simulation efficiency, Kim et al.

(1998) developed multi-move MCMC algorithms.

As σ_t^2 is latent, to facilitate the Bayesian computing, one may enlarge the parameter space by including $(\sigma_1^2, \ldots, \sigma_{T+1}^2)$ as additional parameters. This technique, due to Tanner and Wang (1987), is known as data augmentation. A fully Bayesian model consists of the joint prior distribution of (α, ϕ, σ_v), and $(\sigma_1^2, \ldots, \sigma_{T+1}^2)$, and the joint distribution of the observables, $(y_1, \ldots, y_T)$. Bayesian inference is based on the posterior distribution of the unobservables given $(y_1, \ldots, y_T)$. Let p denote the probability density function. By successive conditioning, the joint prior density is

$$p(\alpha, \phi, \sigma_v, \sigma_1^2, \ldots, \sigma_{T+1}^2) = p(\alpha, \phi, \sigma_v) p(\sigma_1^2 | \phi, \sigma_v) \prod_{t=1}^{T} p(\sigma_{t+1}^2 | \sigma_t^2, \alpha, \phi, \sigma_v). \tag{56}$$

The likelihood, $p(y_1, \ldots, y_T | \alpha, \phi, \sigma_v, \sigma_1^2, \ldots, \sigma_{T+1}^2)$ is given by

$$p(y_1, \ldots, y_T | \alpha, \phi, \sigma_v, \sigma_1^2, \ldots, \sigma_{T+1}^2) = \prod_{t=1}^{T} p(y_t | \sigma_t^2, \alpha, \phi, \sigma_v), \tag{57}$$

If the prior distributions are independent, then, by Bayes' theorem, the joint posterior distribution of the unobservables given the data is proportional to the prior times likelihood, that is,

$$\begin{aligned} p(\alpha, \phi, \sigma_v, \sigma_1^2, \ldots, \sigma_{T+1}^2 | y_1, \ldots, y_T) \quad \propto \quad & p(\alpha) p(\phi) p(\sigma_v) p(\sigma_1^2 | \phi, \sigma_v) \\ & \textstyle\prod_{t=1}^{T} \{ p(\sigma_{t+1}^2 | \sigma_t^2, \alpha, \phi, \sigma_v) p(y_t | \sigma_t^2, \alpha, \phi, \sigma_v) \}. \end{aligned}$$

MCMC algorithms are designed to draw correlated samples, or more precisely stationary and ergodic Markov chains, from the posterior distributions. Once the chains have converged and the number of draws is large enough, a nonparametric approach may be used to approximate any posterior distribution arbitrarily well. In particular, one can obtain the posterior mean, posterior variance, and credible interval.

As a by-product to the Bayesian analysis, one also obtains MCMC samples for the latent variables. Since the posterior distribution $p(\sigma_{T+1}^2 | y_1, \ldots, y_T)$ can be approximated arbitrarily well by MCMC samples, its expectation forms the best one-step-ahead forecast of σ_t^2 in terms of MSFE as far as forecasting is concerned. If a k-step-ahead forecast is needed, one can treat σ_{T+k}^2 as an additional parameter in data augmentation and obtain the posterior distribution $p(\sigma_{T+k}^2|y_1, \ldots, y_T)$. Then the expectation of this conditional distribution forms the best k-step-ahead forecast of σ_t^2

2.2 Multivariate econometric models

The univariate models can be extended to a multivariate setup in straightforwardly. For example, a popular class of multivariate models for forecasting macroeconomic variables is reduced-form VAR models which are the multivariate extension to AR models. A popular multivariate model for forecasting variance and covariance of multiple assets is multivariate GARCH models (MGARCH). A class of methods which are unique to the multivariate setup are factor models and factor-augmented VAR (FAVAR) models. Importantly, most structural models do not have univariate counterparts as equations specified in these models typically correspond to economic theory or economic restrictions.

2.2.1 Vector autoregressive models

A vector autoregressive model (VAR) of order p, usually denoted as VAR(p), for a m-dimensional vector of variables $\boldsymbol{y}_t = (y_{1t}, y_{2t}, \ldots, y_{mt})^\top$ can be written as

$$\boldsymbol{y}_t = \boldsymbol{\mu} + \boldsymbol{\Phi}_1 \boldsymbol{y}_{t-1} + \ldots + \boldsymbol{\Phi}_p \boldsymbol{y}_{t-p} + \boldsymbol{\epsilon}_t, \tag{58}$$

where $\boldsymbol{\Phi}_i$ is a $m \times m$ matrix for $i = 1, \ldots, p$, $\boldsymbol{\mu}$ is a m-dimensional intercept, $\boldsymbol{\epsilon}_t \sim (0, \boldsymbol{\Sigma})$. While it is typically assumed that $\boldsymbol{\epsilon}_t$ is uncorrelated over t but $\boldsymbol{\Sigma}$ is not diagonal in general. Therefore, the elements in $\boldsymbol{\epsilon}_t$ are contemporaneously correlated. The number of parameters in Model (58) is $m + mp^2 + \frac{m(m+1)}{2}$ which quickly increases as m increases.

Note that Model (58) can be written in a compact form as

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{x}_t + \boldsymbol{\epsilon}_t, \tag{59}$$

where

$$\boldsymbol{B} = \left(\boldsymbol{\mu}, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_p\right), \boldsymbol{x}_t = \left(\mathbf{1}, \boldsymbol{y}_{t-1}^\top, \ldots, \boldsymbol{y}_{t-p}^\top\right)^\top. \tag{60}$$

VAR in the form of (59) can be treated as a system of m^{th} equations with the i^{th} equation being

$$y_{t,i} = \boldsymbol{B}_{i\cdot}\boldsymbol{x}_t + \epsilon_{t,i}, \tag{61}$$

for $i = 1, 2, \ldots, m$, where $\boldsymbol{B}_{i\cdot}$ is the i^{th} row of $\boldsymbol{B}$.

Forecasting with VAR is similar to AR case and the h-step-ahead forecast is

$$\hat{\boldsymbol{y}}_{T+h,T} = \boldsymbol{\Phi}_1\hat{\boldsymbol{y}}_{T+h-1,T} + \boldsymbol{\Phi}_2\hat{\boldsymbol{y}}_{T+h-2,T} + \boldsymbol{\Phi}_p\hat{\boldsymbol{y}}_{T+h-p,T}, \tag{62}$$

where $\hat{\boldsymbol{y}}_{T+h-j,T}$ is the $(h-j)$-step-ahead forecast and $\hat{\boldsymbol{y}}_{T+h-j,T} = \boldsymbol{y}_{T+h-j}$ for $j \geqslant h$.

An important concept in VAR is the Granger causality. In Model (58), y_{2t} Granger causes y_{1t} if $y_{2,t}$ helps predict $y_{1,t}$ at some stage in the future. Consider a three-dimensional VAR model,

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} + \begin{bmatrix} \Phi_{11}^1 & \Phi_{12}^1 & \Phi_{13}^1 \\ \Phi_{21}^1 & \Phi_{22}^1 & \Phi_{23}^1 \\ \Phi_{31}^1 & \Phi_{32}^1 & \Phi_{33}^1 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ y_{3,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \\ \epsilon_{3,t} \end{bmatrix}, \tag{63}$$

$y_{2,t}$ does not Granger-cause $y_{1,t}$ if $\Phi_{12}^1 = 0$ which means that $y_{2,t}$ does not depend on the lag of $y_{1,t}$. Note that Granger causality only means that $y_{2,t}$ can be used to improve the forecast of $y_{1,t}$ if $\Phi_{12}^1 \neq 0$. It does not mean the real causal relationship between $y_{1,t}$ and $y_{2,t}$.

2.2.2 Factor models

For many countries, there exists a rich array of macroeconomic time series and financial time series (i.e. a large m in Model (58)). If m is very large, and even when p is moderately small, the total number of parameters in Model (58) can be oversized.[4] Typically a model with too many parameters does not perform well out-of-sample.

To reduce the dimensionality and to extract the information from the large number of time series, factor analysis has been widely used in the empirical macroeconomic literature and in the empirical finance literature. For example, by extending the static factor models previously developed for cross-sectional data, Geweke (1977) proposed the dynamic factor model for time series data. Many empirical studies, such as Sargent and Sims (1977), Giannone et al. (2004), have reported evidence that a large fraction of variance of many macroeconomic series can be explained by a small number of dynamic factors. Stock and Watson (2002) showed that dynamic factors extracted from a large number of predictors lead to improvement in predicting macroeconomic variables. Not surprisingly, high dimensional dynamic factor models have become a popular tool for macroeconomists and policymakers in a data-rich environment. An excellent review on the dynamic factor models is given by Stock and Watson (2011).

The dynamic factor model is given by

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{F}_t \boldsymbol{L}^\top + \boldsymbol{\epsilon}_t, \\ \boldsymbol{F}_t &= \boldsymbol{F}_{t-1} \boldsymbol{\Phi}^\top + \boldsymbol{\eta}_t, \end{aligned} \tag{64}$$

where $\boldsymbol{y}_t$ is a $1 \times m$ vector of time series variables, $\boldsymbol{F}_t$ a $1 \times K$ vector of unobserved latent factors which contains the information extracted from all m time series, $\boldsymbol{L}$ an $m \times K$ factor loading matrix, $\boldsymbol{\Phi}$ the $K \times K$ autoregressive parameter matrix of unobserved latent factors. Typically K is much smaller than m. For example, m can be as large as a few hundreds while K is usually a single-digit number. It is assumed that $\boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{\eta}_t \sim N(\boldsymbol{0}, \boldsymbol{Q})$.

[4] For example, if m=100 and p is 4, the number of parameters contained in Model (58) is 40100.

For the purpose of identification, $\boldsymbol{\Sigma}$ is assume to be diagonal and $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\eta}_t$ are assumed to be independent with each other. Following Bernanke et al. (2005), we set the first $K \times K$ block in the loading matrix $\boldsymbol{L}$ to be the identity matrix.

Clearly Model (64) is a SSM for which the Kalman filter is applicable to obtain forecasts of y_t. While the Kalman filter may be used to obtain the maximum likelihood estimates of the model, if m is large, even when p is small, the parameter space is of a high dimension, making maximum likelihood estimation not operational. To avoid this numerical issue, one can use a Bayesian method to sample from the posterior distribution; see for example, Aguilar and West (2000), Bai and Wang (2015).

2.2.3 Factor-augmented vector autoregressive models

Following Bernanke et al. (2005), let $\boldsymbol{y}_t$ be a $N \times 1$ stationary time series with a large N, $\boldsymbol{X}_t$ a M vector of observable variables which drive the dynamics of the economy (such as the federal funds rate). Suppose the information included in $\boldsymbol{y}_t$ can be summarized by a $K \times 1$ unobservable variables $\boldsymbol{F}_t$ where $N \gg K$.

Following Bernanke et al. (2005), $\boldsymbol{y}_t$ is driven both by $\boldsymbol{F}_t$ and $\boldsymbol{X}_t$ as

$$\boldsymbol{y}_t = \boldsymbol{\Lambda}^F \boldsymbol{F}_t + \boldsymbol{\Lambda}^X \boldsymbol{X}_t + \boldsymbol{\epsilon}_t, \tag{65}$$

where $\boldsymbol{\Lambda}^F$ is an $N \times K$ matrix of factor loadings, $\boldsymbol{\Lambda}^X$ is a $N \times M$ matrix. The joint dynamics of $(\boldsymbol{F}_t, \boldsymbol{X}_t)$ are given by

$$\begin{bmatrix} \boldsymbol{F}_t \\ \boldsymbol{X}_t \end{bmatrix} = \Phi(L) \begin{bmatrix} \boldsymbol{F}_{t-1} \\ \boldsymbol{X}_{t-1} \end{bmatrix} + \boldsymbol{\eta}_t \tag{66}$$

where $\Phi(L)$ is a conformable lag polynomial of finite order d and $\boldsymbol{\eta}_t \sim N(0, \boldsymbol{Q})$. Equation (66) is referred to as the factor-augmented VAR (FAVAR) model (Bernanke et al., 2005).

2.2.3.1 Estimation of FAVAR Although Equation (66) is a VAR of $(\boldsymbol{F}_t, \boldsymbol{Y}_t)$, it can not be estimated directly since $\boldsymbol{F}_t$ is unobservable. Bernanke et al. (2005) proposed a two-step method which estimates $\boldsymbol{F}_t$ from Equation (65) in the first step, and then estimate Equation (66) using the results from the first step.

In the first step, following Bernanke et al. (2005), the principal components analysis (PCA) is used to extract the first K components denoted by $\hat{\boldsymbol{C}}_t\left(\boldsymbol{F}_t, \boldsymbol{X}_t\right)$. It is invalid to directly estimate VAR with $\hat{\boldsymbol{C}}_t\left(\boldsymbol{F}_t, \boldsymbol{X}_t\right)$ and $\boldsymbol{X}_t$ since $\hat{\boldsymbol{C}}_t\left(\boldsymbol{F}_t, \boldsymbol{X}_t\right)$ involves $\boldsymbol{X}_t$. To remove the dependence of $\hat{\boldsymbol{C}}_t\left(\boldsymbol{F}_t, \boldsymbol{X}_t\right)$ on $\boldsymbol{X}_t$, Bernanke et al. (2005) divided $\boldsymbol{y}_t$ into two groups, "slow-moving" variables and "fast-moving" variables. Slowing-moving variables, such as wages, are assumed not to respond contemporaneously to unexpected change in policy instrument, while fast-moving variables, such as asset prices, are assumed respond contemporaneously to unexpected change in policy instrument.

Since slow-moving variables are uncorrelated with $\boldsymbol{X}_t$, the principal components from them, $\hat{\boldsymbol{F}}_t^s$, are also uncorrelated with $\boldsymbol{X}_t$. Then by estimating the following regression

$$\hat{\boldsymbol{C}}_t\left(\boldsymbol{F}_t, \boldsymbol{X}_t\right) = \boldsymbol{\beta}_s \hat{\boldsymbol{F}}_t^s + \boldsymbol{\beta}_X \boldsymbol{X}_t + \boldsymbol{v}_t, \tag{67}$$

the dependence of $\hat{\boldsymbol{C}}_t\left(\boldsymbol{F}_t, \boldsymbol{X}_t\right)$ on $\boldsymbol{X}_t$ can be removed, leading to

$$\hat{\boldsymbol{F}}_t = \hat{\boldsymbol{C}}_t\left(\boldsymbol{F}_t, \boldsymbol{X}_t\right) - \hat{\boldsymbol{\beta}}_X \boldsymbol{X}_t. \tag{68}$$

In the second step, we use $\boldsymbol{X}_t$ and $\hat{\boldsymbol{F}}_t$ to estimate (66).

From (66), suppose that we have a FAVAR(d)

$$\begin{bmatrix} \boldsymbol{F}_t \\ \boldsymbol{X}_t \end{bmatrix} = \Phi^1 \begin{bmatrix} \boldsymbol{F}_{t-1} \\ \boldsymbol{X}_{t-1} \end{bmatrix} + \Phi^2 \begin{bmatrix} \boldsymbol{F}_{t-2} \\ \boldsymbol{X}_{t-2} \end{bmatrix} + \cdots + \Phi^d \begin{bmatrix} \boldsymbol{F}_{t-d} \\ \boldsymbol{X}_{t-d} \end{bmatrix} + \boldsymbol{\eta}_t. \tag{69}$$

Then the forecast of $\left(\boldsymbol{F}_{T+h}, \boldsymbol{X}_{T+h}\right)$ can be obtained as in (62) since it is a VAR model. The forecast of $\left(\boldsymbol{F}_{T+h}, \boldsymbol{X}_{T+h}\right)$ can be then used to forecast $\boldsymbol{y}_{T+h}$ based on Equation (65).

2.2.4 Multivariate GARCH models

Let $\boldsymbol{y}_t$ denote a vector of N log-returns $\boldsymbol{y}_t = (y_{1t}, y_{2t}, \cdots, y_{Nt})^\top$. The key variable that multivariate GARCH (MGARCH) tries to model is the conditional covariance matrix, $\mathbb{E}_{t-1}(\boldsymbol{y}_t\boldsymbol{y}_t^\top) := \boldsymbol{H}_t$. It is a $N \times N$ dimensional symmetric and positive definite matrix. There are three general approaches in modeling $\boldsymbol{H}_t$. The first one is to extend the univariate GARCH framework to a multivariate version in which both variances and covariances are allowed to be time-varying. The second one is to model the dynamics of the conditional correlation coefficients. The third one is to use dimension reduction techniques such as factor models. Here we only review two MGARCH models, one from each of the first two approaches. The literature is well reviewed in Bauwens et al. (2006).

Following Engle and Kroner (1995), the BEKK specification[5] for a MGARCH(1,1) model has the form

$$\boldsymbol{H}_t = \boldsymbol{C}\boldsymbol{C}^\top + \boldsymbol{A}\,\boldsymbol{y}_{t-1}\boldsymbol{y}_{t-1}^\top\,\boldsymbol{A}^\top + \boldsymbol{B}\,\boldsymbol{H}_{t-1}\boldsymbol{B}^\top, \tag{70}$$

where $\boldsymbol{C}$ is a $(N \times N)$ lower triangular matrix of unknown parameters, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are $(N \times N)$ matrices each containing N^2 unknown parameters associated with the lagged disturbances and the lagged conditional covariance matrix, respectively.

A drawback of the BEKK model is that it has to so many parameters when N is even moderately large. For example, when $N = 3$ there are 24 parameters. When $N = 5$ there are 65 parameters. That explains why the BEKK model has not found many applications in practice.

The dynamic conditional correlation (DCC) model of Engle (2002) deals with the correlation coefficients directly and is specified as

$$\boldsymbol{H}_t = \boldsymbol{S}_t\,\boldsymbol{R}_t\,\boldsymbol{S}_t, \tag{71}$$

[5] The BEKK stands for Baba, Engle, Kraft and Kroner. An early version of the paper was written by Baba, Engle, Kraft, and Kroner, which led to the acronym BEKK and was used in Engle and Kroner (1995) for the new parameterization presented.

where

$$\boldsymbol{S}_t = \begin{bmatrix} \sqrt{h_{11t}} & & 0 \\ & \ddots & \\ 0 & & \sqrt{h_{NNt}} \end{bmatrix}, \tag{72}$$

and $\boldsymbol{R}_t$ is the conditional correlation matrix. DCC assumes

$$h_{iit} = \alpha_{i0} + \alpha_{i1} y_{i\,t-1}^2 + \beta_{i1} h_{ii\,t-1}, \qquad i = 1, 2, \cdots N, \tag{73}$$

$$\boldsymbol{R}_t = \operatorname{diag}(\boldsymbol{Q}_t)^{-1/2}\, \boldsymbol{Q}_t \operatorname{diag}(\boldsymbol{Q}_t)^{-1/2}, \tag{74}$$

where $\boldsymbol{Q}_t$ is a pseudo correlation matrix which evolves as

$$\boldsymbol{Q}_t = (1 - \alpha - \beta)\boldsymbol{Q} + \alpha \boldsymbol{z}_{t-1} \boldsymbol{z}_{t-1}^\top + \beta \boldsymbol{Q}_{t-1}. \tag{75}$$

Forecast in $\boldsymbol{H}_t$ can be carried out in the same way as in univariate GARCH models.

2.2.5 Multivariate stochastic volatility models

While MGARCH models assume $\boldsymbol{H}_t$ as a function of past returns, in multivariate stochastic volatility (MSV) models, $\boldsymbol{H}_t$ depends on a separate error term. One of the simplest MSV model was proposed in Harvey et al. (1994) as

$$\begin{aligned} \boldsymbol{y}_t &= \boldsymbol{H}_t^{1/2} \boldsymbol{\epsilon}_t, \\ \boldsymbol{H}_t &= \operatorname{diag}\{\exp(h_{1t}/2), \ldots, \exp(h_{Nt}/2)\} := \operatorname{diag}\{\exp(\boldsymbol{h}_t/2)\}, \\ \boldsymbol{h}_{t+1} &= \boldsymbol{\mu} + \boldsymbol{\phi} \diamond \boldsymbol{h}_t + \boldsymbol{v}_{t+1}, \\ \begin{bmatrix} \boldsymbol{\epsilon}_t \\ \boldsymbol{v}_{t+1} \end{bmatrix} &\sim N\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{P}_\epsilon & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_v \end{bmatrix} \right), \end{aligned}$$

where $\boldsymbol{h}_t = (h_{1t}, \ldots, h_{Nt})^\top$, $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$ are $N \times 1$ parameter vectors, the operator "$\diamond$" denotes the Hadamard (or element-by-element) product, $\boldsymbol{P}_\epsilon := (\rho_{ij})$ is the correlation matrix, and

$\boldsymbol{\Sigma}_v$ is a symmetric positive definite matrix. Clearly, $\rho_{ii} = 1$ for all i.

Many other MSV models have been proposed in recent years. Various specifications in the context of 2 dimensional case can be found in Yu and Meyer (2006), some of which have straightforward high-dimensional extensions. A review of the literature on estimating MSV models can be found in Asai et al. (2006). To forecast future values of $\boldsymbol{h}_t$, if a Bayesian MCMC method is used, one can use the data augmentation technique by treating the future values of $\boldsymbol{h}_t$ as the parameters, as explained in the section where the univariate SV models were discussed.

2.2.6 Structure vector autoregressive models

Suppose we have a three dimensional VAR(1) model, for real GDP growth (Δy_t), inflation (π_t) and the interest rate (r_t), as

$$\begin{bmatrix} \Delta y_t \\ \pi_t \\ r_t \end{bmatrix} = \begin{bmatrix} \Phi_{11}^1 & \Phi_{12}^1 & \Phi_{13}^1 \\ \Phi_{21}^1 & \Phi_{22}^1 & \Phi_{23}^1 \\ \Phi_{31}^1 & \Phi_{32}^1 & \Phi_{33}^1 \end{bmatrix} \begin{bmatrix} \Delta y_{t-1} \\ \pi_{t-1} \\ r_{t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \\ \epsilon_{3,t} \end{bmatrix}. \tag{76}$$

The VAR as in Equation (76) is denoted as the reduced-form VAR. With Model (76), we can describe the dynamic properties of the three variables from the lagged coefficients (Φ_{ii}^1), explore the interaction between any two variables from the cross-variable coefficients $(\Phi_{ij}^1$, $i \neq j)$ and forecast the future values of the variables.

In practice, however, it is also important to understand the effect of a shock over time on the different variables and the contribution of a shock to the behaviour of the different variables. We refer to the first as Impulse Response Function (IRF) analysis and the second as Forecast Error Variance Decomposition (FEVD). In IRF and FEVD, we try to analyze the effect of structure shocks.

We can not interpret the reduced-form error term (i.e. $\epsilon_t := (\epsilon_{1,t}, \epsilon_{2,t}, \epsilon_{3,t})^\top$) as structure shocks, since it is impossible to isolate the effect of different shocks because they are correlated. In other words, the fact that the covariance matrix of ϵ_t is not diagonal makes it difficult to interpret the impact of shocks.

To implement the policy analysis, such as the IRF and FVED, we need orthogonal shocks. In this section, we introduce the Structure VAR (SVAR) model as

$$\boldsymbol{A}\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{y}_{t-1} + \boldsymbol{u}_t, \quad \boldsymbol{u}_t \sim iid(\boldsymbol{0}, \boldsymbol{I}), \tag{77}$$

where $\boldsymbol{A}$ is an invertible matrix, $\boldsymbol{u}_t$ are serially uncorrelated and independent of each other and can be interpreted as structure shocks since $\boldsymbol{I}$ is an identity matrix. Then the structure form of Model (76) is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{31} & a_{33} \end{bmatrix} \begin{bmatrix} \Delta y_t \\ \pi_t \\ r_t \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} \Delta y_{t-1} \\ \pi_{t-1} \\ r_{t-1} \end{bmatrix} + \begin{bmatrix} u_{\Delta yt} \\ u_{\pi t} \\ u_{rt} \end{bmatrix}, \tag{78}$$

where $u_{\Delta yt}$, $u_{\pi t}$ and u_{rt} can be interpreted as aggregate shock, cost-push shock and monetary policy shock respectively. And Model (76) also can be expressed into

$$\begin{cases} a_{11}\Delta y_t + a_{12}\pi_t + a_{13}r_t = b_{11}\Delta y_{t-1} + b_{12}\pi_{t-1} + b_{13}r_{t-1} + u_{\Delta yt} \\ a_{21}\Delta y_t + a_{22}\pi_t + a_{13}r_t = b_{21}\Delta y_{t-1} + b_{22}\pi_{t-1} + b_{23}r_{t-1} + u_{\pi t} \\ a_{31}\Delta y_t + a_{32}\pi_t + a_{33}r_t = b_{31}\Delta y_{t-1} + b_{32}\pi_{t-1} + b_{33}r_{t-1} + u_{rt} \end{cases}, \tag{79}$$

which is a linear equation system.

SVAR usually contains a set of equations with each equation describing the type of decision rules motivated by economic theory. One example is that consumers demanded a certain quantity of aggregate output based on the aggregate price level as well as how liquid they were, with the latter being measured by real money holdings. Clearly, SVAR aims to capture how endogenous variables are related to other endogenous variables and some exogenous variables. While SVAR facilitates interpreting data, it makes the estimation more difficult due to the presence of endogeneity which means that SVAR can not be estimated by equation-by-equation OLS. For instance, in the equation for Δy_t, we have

$$\Delta y_t = \frac{1}{a_{11}} \left(-a_{12}\pi_t - a_{13}r_t + b_{11}\Delta y_{t-1} + b_{12}\pi_{t-1} + b_{13}r_{t-1} + u_{\Delta yt} \right), \tag{80}$$

with the assumption that $a_{11} \neq 0$, where $cov(\pi_t, u_{\Delta yt}) \neq 0$ because of the contemporaneous dependence of Δy_t on π_t.

To solve this problem, we transform SVAR to a reduced-form VAR by pre-multiplying Equation (77) by A^{-1}

$$\boldsymbol{y}_t = \boldsymbol{\Phi}^1 \boldsymbol{y}_{t-1} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim WN(\boldsymbol{0}, \boldsymbol{\Sigma}), \tag{81}$$

where $\boldsymbol{\Phi}^1 = \boldsymbol{A}^{-1}\boldsymbol{B}$, $\boldsymbol{\epsilon}_t = \boldsymbol{A}^{-1}\boldsymbol{u}_t$ and $\boldsymbol{\Sigma} = \boldsymbol{A}^{-1}\left(\boldsymbol{A}^{-1}\right)^\top$. We can estimate Equation (81) using equation by equation OLS to get $\hat{\boldsymbol{\Phi}}^1$ and $\hat{\boldsymbol{\Sigma}}$ and then recover $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{B}}$. Here the key step is to recover the $\hat{\boldsymbol{A}}$ from $\hat{\boldsymbol{\Sigma}}$, that is,

$$\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{A}}^{-1}\left(\hat{\boldsymbol{A}}^{-1}\right)^\top. \tag{82}$$

Note that $\hat{\boldsymbol{A}}^{-1}$ has m^2 different elements since it's not symmetric. However, both $\hat{\boldsymbol{A}}^{-1}\left(\hat{\boldsymbol{A}}^{-1}\right)^\top$ and $\hat{\boldsymbol{\Sigma}}$ are symmetric so that we can only get $\frac{m(m+1)}{2}$ non-redundant equations from (82). Here we have $\frac{m(m+1)}{2}$ equations for m^2 unknown elements so there are more than one solution to Equation (82). This means $\hat{A}$ is not identifiable. There exists several approaches to identify $\hat{A}$ with the help of economic theory.

2.2.6.1 Short-run restrictions From Model (82), there are $\frac{m(m+1)}{2}$ equations for m^2 unknown elements. We still need extra $\frac{m(m-1)}{2}$ constraints. The short-run restriction depends on the Cholesky decomposition. The Cholesky decomposition of a positive definite matrix is a decomposition of the form $\boldsymbol{\Sigma} = \boldsymbol{L}^\top\boldsymbol{L}$, where $\boldsymbol{L}^\top$ is a unique lower triangular matrix with real and positive diagonal entries. By setting the $\frac{m(m-1)}{2}$ upper triangular elements of $\boldsymbol{A}$ to 0, we have $\boldsymbol{L}^\top = \boldsymbol{A}^{-1}$. Then $\boldsymbol{A}^{-1}$ is a lower triangular matrix, so is A. $\hat{\boldsymbol{A}}$ can be recovered from the Cholesky decomposition of $\hat{\boldsymbol{\Sigma}}$, that is, $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{L}}^\top\hat{\boldsymbol{L}}$.

With the short-run restrictions, Model (78) can be written as

$$\begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{31} & a_{33} \end{bmatrix} \begin{bmatrix} \Delta y_t \\ \pi_t \\ r_t \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} \Delta y_{t-1} \\ \pi_{t-1} \\ r_{t-1} \end{bmatrix} + \begin{bmatrix} u_{\Delta yt} \\ u_{\pi t} \\ u_{rt} \end{bmatrix}. \tag{83}$$

From Model (83), the economic meaning is that Δy_t is only contemporaneously affected by $u_{\Delta yt}$, not by $u_{\pi t}$ and u_{rt}. Similarly, π_t is only contemporaneously affected by $u_{\Delta yt}$ and $u_{\pi t}$, not by u_{rt}, and r_t is contemporaneously affected by $u_{\Delta yt}$, $u_{\pi t}$ and u_{rt}. Note that different ordering of the variables may led to different results,. In practice we can consider different ordering to evaluate the sensitivity.

2.2.6.2 IRF An IRF describes the evolution of the variable of interest (for example, the q^{th} element of $\boldsymbol{y}_t$, $y_{q,t}$, in (77)) along a specified time horizon after a structure shock change $\boldsymbol{s}_{j,t}$ which is due to the change of the j^{th} element of $\boldsymbol{u}_t$ in (77) in a given moment. Here a structure shock change is defined as a vector with one element equal to 1 and all the others equal to 0 such as $\boldsymbol{s}_{j,t} = [0\ 0\ \cdots\ 1\ \cdots 0]^\top$ where only the j^{th} element is 1.

The SVAR model (77) can be rewritten as

$$\boldsymbol{y}_t = \boldsymbol{\Phi} \boldsymbol{y}_{t-1} + \boldsymbol{A}^{-1} \boldsymbol{u}_t, \quad \boldsymbol{u}_t \sim (\boldsymbol{0}, \boldsymbol{I}), \tag{84}$$

where $\boldsymbol{\Phi} = \boldsymbol{A}^{-1}\boldsymbol{B}$ under the condition that $\boldsymbol{A}$ is identifiable. By using the lag operator, (84) can be written as $\boldsymbol{y}_t = (\boldsymbol{I} - \boldsymbol{\Phi L})^{-1} \boldsymbol{A}^{-1} \boldsymbol{u}_t$, which admits an MA($\infty$) representation

$$\boldsymbol{y}_t = \boldsymbol{A}^{-1}\boldsymbol{u}_t + \boldsymbol{\Phi A}^{-1}\boldsymbol{u}_{t-1} + \boldsymbol{\Phi}^2 \boldsymbol{A}^{-1} \boldsymbol{u}_{t-2} + \cdots. \tag{85}$$

Then, holding other variables constant, the change of the q^{th} element of $\boldsymbol{y}$ at period $t+i$, $y_{q,t+i}$, in response to a unit change to the j^{th} structure shock $\boldsymbol{u}_{j,t}$ at period t is

$$\mathcal{IR}_{i,qj} := \frac{\partial y_{q,t+i}}{\partial \boldsymbol{u}_t} \boldsymbol{s}_{j,t} = \boldsymbol{\Psi}_{q\cdot}^{i\top} \boldsymbol{s}_{j,t}, \tag{86}$$

where $\boldsymbol{\Psi}^i = \boldsymbol{\Phi}^i \boldsymbol{A}^{-1}$ and $\boldsymbol{\Psi}^i_{q\cdot}$ denotes the q^{th} row of $\boldsymbol{\Psi}^i$ for $q, j = 1, 2, \ldots, m$ and $i = 0, 1, \ldots$. Note that the $(q, j)^{th}$ element of matrix $\boldsymbol{\Phi}^i$ is the impulse response of $y_{q,t+i}$ with respect to the j^{th} structure shock $\boldsymbol{s}_{j,t}$. $\boldsymbol{\Phi}^i$ is a impulse response matrix at $t + i$.

The IRFs measure the response of current and future values of each of the variables to a one-unit increase in the current value of one of the structural shocks, assuming that this shock returns to zero in subsequent periods and that all other shocks are equal to zero (Cesa-Bianchi et al., 2015).

2.2.6.3 FEVD FEVD is a way to measure how important each shock is in explaining the forecast error variance of each variable. It is the fraction of the forecast error variance of each variable due to each shock at different forecasting horizon. To illustrate the basic idea of FEVD, consider a two dimensional SVAR model:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} A^{-1}_{11} & A^{-1}_{12} \\ A^{-1}_{21} & A^{-1}_{22} \end{bmatrix} \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix}. \tag{87}$$

The one-step-ahead forecast of $\boldsymbol{y}_{T+1}$ based on $(\boldsymbol{y}_T, \boldsymbol{y}_{T-1}, \ldots, \boldsymbol{y}_1)$, $\boldsymbol{y}_{T+1,T}$ is

$$\boldsymbol{y}_{T+1,T} = \boldsymbol{\Phi} \boldsymbol{y}_T, \tag{88}$$

and the forecast error variances of $e_{1,T+1}$ and $e_{2,T+1}$ are

$$\operatorname{Var}(e_{1,T+1}) = \left(A^{-1}_{11}\right)^2 + \left(A^{-1}_{12}\right)^2, \tag{89}$$
$$\operatorname{Var}(e_{2,T+1}) = \left(A^{-1}_{21}\right)^2 + \left(A^{-1}_{22}\right)^2, \tag{90}$$

since the covariance matrix of $\boldsymbol{u}_{T+1}$ is an identity matrix.

From (89) and (90), we can do forecast error variance decomposition. For instance, the fraction of one-step-ahead forecast error variance of the first variable $y_{1,T+1}$ explained by the first structural shock $u_{1,T+1}$ and the second shock $u_{1,T+1}$ are

$$\text{FEVD}_{1,1} = \frac{\left(A_{11}^{-1}\right)^2}{\left(A_{11}^{-1}\right)^2 + \left(A_{12}^{-1}\right)^2}, \text{FEVD}_{1,2} = \frac{\left(A_{12}^{-1}\right)^2}{\left(A_{11}^{-1}\right)^2 + \left(A_{12}^{-1}\right)^2}. \tag{91}$$

2.2.6.4 Long-run restrictions As we have discussed in the previous sections, identification of the shocks is needed to compute IRF and FEVD. The short-run restrictions impose the constraints in a contemporaneous way, but economic theory tell us less about the short term behavior than the long-term behavior. For instance, the positive aggregate demand shocks can affect output in short-run, but have no effect in the long-run. The long-run restrictions use the economic theory about the long-run economic behavior to identify the structure shocks. The long-run restrictions were proposed by Blanchard and Quah (1989) to identify supply and demand shocks.

Recall the SVAR model can be written as

$$\boldsymbol{y}_t = \boldsymbol{\Phi}\boldsymbol{y}_{t-1} + \boldsymbol{A}^{-1}\boldsymbol{u}_t, \quad \boldsymbol{u}_t \sim (\boldsymbol{0}, \boldsymbol{I}), \tag{92}$$

where $\boldsymbol{\Phi} = \boldsymbol{A}^{-1}\boldsymbol{B}$. Then the impulse response of a structure shock at time t can be expressed as $\boldsymbol{A}^{-1}$. The effect is $\boldsymbol{\Phi}^i\boldsymbol{A}^{-1}$ after i periods. Then the long-run effect of the structure shocks is defined as the sum of the impulse response at each period:

$$\boldsymbol{D} = \left(\boldsymbol{I} + \boldsymbol{\Phi} + \boldsymbol{\Phi}^2 + \cdots\right)\boldsymbol{A}^{-1} = (\boldsymbol{I} - \boldsymbol{\Phi})^{-1}\boldsymbol{A}^{-1}, \tag{93}$$

where $\boldsymbol{D}$ denotes the long-run effect. Note that

$$\boldsymbol{D}\boldsymbol{D}^\top = (\boldsymbol{I} - \boldsymbol{\Phi})^{-1}\boldsymbol{A}^{-1}(\boldsymbol{A}^{-1})\left((\boldsymbol{I} - \boldsymbol{\Phi})^{-1}\right)^\top = (\boldsymbol{I} - \boldsymbol{\Phi})^{-1}\boldsymbol{\Sigma}\left((\boldsymbol{I} - \boldsymbol{\Phi})^{-1}\right)^\top. \tag{94}$$

When an estimate of $\boldsymbol{D}\boldsymbol{D}^\top$ is available, say $\widehat{\boldsymbol{D}\boldsymbol{D}^\top}$, we can use the Cholesky decomposition to recover $\hat{\boldsymbol{D}}$ which means that we restrict the long-run effect $\boldsymbol{D}$ to be a lower triangular matrix. And then we can recover $\hat{\boldsymbol{A}}$ from (93), $\hat{\boldsymbol{A}}^{-1} = \left(\boldsymbol{I} - \hat{\boldsymbol{\Phi}}\right)\hat{\boldsymbol{D}}$.

2.2.6.5 Sign restrictions Instead of imposing constraints on the coefficients directly, the sign restrictions impose constraints to the IRF. Recall that the identification problem of SVAR

$$\boldsymbol{\Sigma} = \boldsymbol{A}^{-1}(\boldsymbol{A}^{-1})^{\top}. \tag{95}$$

Without constraints, the solution of (95) is not unique. If some $\boldsymbol{A}^{-1}$ satisfies (95), then any other matrix takes the form $\boldsymbol{A}^{-1}\boldsymbol{P}^{\top}$ is also valid if $\boldsymbol{P}^{\top}\boldsymbol{P} = \boldsymbol{I}$. The short-run and long-run restrictions impose $\frac{m(m-1)}{2}$ constraints to obtain a unique solution of $\boldsymbol{A}$.

The basic idea of sign restrictions is to find a set of $\boldsymbol{A}$ such that the IRF satisfies some properties according to economic theory. For example, the monetary contractions should raise the interest rate and lower prices, while a positive demand shock should raise the output and prices (Uhlig (2005)). The algorithm from Danne (2015) illustrates the procedure of the sign restrictions.

Algorithm 4 The Sign Restriction of SVAR

```
1. From reduced-form VAR, obtain the estimator $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{\Sigma}}$.
2. For $m = 1, 2, \ldots, M$:
    (a) Draw a random orthonormal matrix $\boldsymbol{P}$, compute $\hat{\boldsymbol{A}}_m^{-1} = \hat{\boldsymbol{A}}^{-1}\boldsymbol{P}^{\top}$;
    (b) Compute IRF$^{(m)}$ based on $\hat{\boldsymbol{A}}_m^{-1}$;
        i. If IRF$^{(m)}$ satisfies the sign restrictions, keep it;
        ii. If IRF$^{(m)}$ doesn't satisfy the sign restrictions, discard it.
3. For the remained $N$ replications, report the median impulse response.
```

2.2.7 Dynamic stochastic general equilibrium models

Dynamic stochastic general equilibrium (DSGE) models build on explicit micro-foundations by allowing agents to do optimization. They have become very popular in macroeconomics over the last 30 years. Earlier efforts made in the literature are the developments of estimation methodology so that the estimation of variants of DSGE models can compete with more standard time series models such as VAR.

Estimation and evaluation of the DSGE models require one to solve them and then to construct a linear or nonlinear state-space approximation. Bayesian methods have been widely applied to estimate the DSGE models. For a linear Gaussian approximation, the Kalman filter can be used to compute the likelihood function; for example, Schorfheide (2000), Lubik and Schorfheide (2006), An and Schorfheide (2007), among others. For a non-linear non-Gaussian approximation, Fernández-Villaverde and Rubio-Ramírez (2005) used the particle filter to calculate the likelihood. More recent efforts have also been made to show the usefulness of these models for the purpose of forecasting economic variables. See Herbst and Schorfheide (2015) for a comprehensive literature review.

2.2.7.1 A Small-Scale New Keynesian DSGE Model We begin with a small-scale new Keynesian DSGE model from An and Schorfheide (2007) which consists of a final goods-producing firm, a continuum of intermediate goods-producing firms, a representative household, a monetary authority and a fiscal authority.

The representative final goods-producing firm in a perfectly competitive market combines a continuum of intermediate goods indexed by $j \in [0,1]$ using the technology

$$Y_t = \left(\int_0^1 Y_t\,(j)^{1-\nu}\, dj \right)^{\frac{1}{1-\nu}}.$$

Here $1/\nu > 1$ represents the elasticity of demand for each intermediate good. The firm takes input prices $P_t(j)$ and output prices P_t as given. Profit maximization is as follows:

$$\max_{Y_t(j)} P_t Y_t - \int_0^1 Y_t(j) P_t(j)\, dj. \tag{96}$$

Intermediate good j is produced by a monopolist with linear production technology

$$Y_t(j) = A_t N_t(j),$$

where A_t is an exogenous productivity process that is common to all firms and $N_t(j)$ is the labor input of firm j. Labor is hired in a perfectly competitive market at the real wage W_t. Firms face nominal rigidities in terms of quadratic price adjustment costs

$$AC_t(j) = \frac{\phi}{2}\left(\frac{P_t(j)}{P_{t-1}(j)} - \pi\right)^2 Y_t(j),$$

where ϕ governs the price stickiness in the economy and π is the steady-state inflation rate associated with the final good. Firm j chooses its labor input $N_t(j)$ and $P_t(j)$ to maximize the present value of future profits:

$$\max_{P_t(j)} E_t\left[\sum_{s=0}^{\infty} \beta^s Q_{t+s|t}\left(\frac{P_{t+s}(j)}{P_{t+s}} Y_{t+s}(j) - \frac{W_{t+s}}{P_{t+s}} N_{t+s}(j) - AC_{t+s}(j)\right)\right], \tag{97}$$

where $Q_{t+s|t}$ is the time t value of a unit of the consumption good in period $t+s$ to the household, which is treated as exogenous by j

The representative household has positive utility from real money balances M_t/P_t and consumption C_t relative to a habit stock which is given by the level of technology A_t. And it has negative utility from hours worked H_t. Then the representative household maximizes its utility as follows

$$E_t\left[\sum_{s=0}^{\infty} \beta^s \left(\frac{(C_{t+s}/A_{t+s})^{1-\tau} - 1}{1-\tau} + \chi_M \ln\left(\frac{M_{t+s}}{P_{t+s}}\right) - \chi_H H_{t+s}\right)\right], \tag{98}$$

with budget constraints

$$P_t C_t + B_t + M_t - M_{t-1} + T_t = W_t H_t + R_{t-1} B_{t-1} + P_t D_t + P_t SC_t,$$

where β is the discount factor, $1/\tau$ is the intertemporal elasticity of substitution, and χ_M and $\chi_H = 1$ are scale factors that determine steady-state real money balances and hours worked. B_t is the nominal government bonds which can be traded and pay the gross interest rate R_t. The representative household receives the real profits D_t from the firms and pays the government the lump-sum taxes T_t. SC_t is the net cash inflow from trading a full set of state-contingent securities.

Monetary policy is described by an interest rate feedback rule of the form

$$R_t = R_t^{*1-\rho_R} R_{t-1}^{\rho_R} e^{\epsilon_{R,t}}, \tag{99}$$

where $\epsilon_{R,t}$ is a monetary policy shock and R_t^* is the nominal target rate. The central bank reacts to inflation and deviations of output from potential output

$$R_t^* = r\pi^* \left(\frac{\pi_t}{\pi^*}\right)^{\psi_1} \left(\frac{Y_t}{Y_t^*}\right)^{\psi_2}, \tag{100}$$

where r is the steady-state real interest rate, π_t is the gross inflation rate defined as $\pi_t = P_t/P_{t-1}$, and π^* is the target inflation rate, Y_t^* is the potential output.

The government spending is a fraction ζ_t of aggregate output Y_t, where $\zeta_t \in [0, 1]$ follows an exogenous process. The government's budget constraints is given by

$$P_t G_t + R_{t-1} B_{t-1} = T_t + B_t + M_t - M_{t-1}, \tag{101}$$

where $G_t = \zeta_t Y_t$.

The law of aggregate productivity A_t is

$$\ln A_t = \ln \gamma + \ln A_{t-1} + \ln z_t, \tag{102}$$

where $\ln z_t = \rho_z \ln z_{t-1} + \epsilon_{z,t}$. Let $g_t = 1/\left(1 - \zeta_t\right)$ denote the government spending which follows

$$\ln g_t = \left(1 - \rho_g\right) \ln g + \rho_g \ln g_{t-1} + \epsilon_{g,t}. \tag{103}$$

The monetary policy shock $\epsilon_{R,t}$, the government spending shock $\epsilon_{g,t}$, the technology shock $\epsilon_{z,t}$ are assumed to be serially uncorrelated. The three shocks are independent of each other at all leads and lags and are normally distributed with mean zero and standard deviation σ_z, σ_g and σ_g, respectively.

The market-clearing conditions are given by

$$Y_t = C_t + G_t + AC_t, \tag{104}$$

$$H_t = N_t. \tag{105}$$

From the first order conditions of (96), (97), (98) and the market-clearing conditions (104) and (105), the following relationships can be derived

$$\begin{aligned}
1 &= \mathbb{E}_t \left[e^{-\tau \widehat{c}_{t+1} + \tau \widehat{c}_t + \widehat{R}_t - \widehat{z}_{t+1} - \widehat{\pi}_{t+1}} \right], && (106) \\
\frac{1-\nu}{\nu \phi \pi^2} \left(e^{\tau \widehat{c}_t} - 1 \right) &= \left(e^{\widehat{\pi}_t} - 1 \right) \left[\left(1 - \frac{1}{2\nu} \right) e^{\widehat{\pi}_t} + \frac{1}{2\nu} \right], && (107) \\
&\quad - \beta E_t \left[\left(e^{\widehat{\pi}_{t+1}} - 1 \right) e^{-\tau \widehat{c}_{t+1} + \tau \widehat{c}_t + \widehat{y}_{t+1} - \widehat{y}_t + \widehat{\pi}_{t+1}} \right], && \\
e^{\widehat{c}_t - \widehat{y}_t} &= e^{-\widehat{g}_t} - \frac{\phi \pi^2 g}{2} \left(e^{\widehat{\pi}_t} - 1 \right)^2, && (108)
\end{aligned}$$

where $c_t = C_t/A_t$, $y_t = Y_t/A_t$ are the detrended variables, $\widehat{y}_t$, $\widehat{\pi}_t$, $\widehat{c}_t$ are the percentage deviations from the steady-state for the output, the inflation, the consumption.

From (99), (100), (102) and (103), the monetary policy rule and the shock process can be rewritten in deviation form as

$$\widehat{R}_t = \rho_R \widehat{R}_{t-1} + (1-\rho_R)\,\psi_1 \widehat{\pi}_t + (1-\rho_R)\,\psi_2\,(\widehat{y}_t - \widehat{g}_t) + \epsilon_{R,t}, \tag{109}$$

$$\widehat{g}_t = \rho_g \widehat{g}_{t-1} + \epsilon_{g,t}, \tag{110}$$

$$\widehat{z}_t = \rho_z \widehat{z}_{t-1} + \epsilon_{z,t}, \tag{111}$$

where $\widehat{R}_t$, $\widehat{g}_t$, $\widehat{z}_t$ are the percentage deviations from the steady-state for the interest rate, the government expenditure, and the technology growth rate.

2.2.7.2 Casting DSGE model into a state-space form Equations (106) – (111) constitute a DSGE model. There are two main concerns in the estimation process. First, the system is nonlinear. Second, $\widehat{y}_t$, $\widehat{\pi}_t$, $\widehat{R}_t$, $\widehat{c}_t$, $\widehat{g}_t$, $\widehat{z}_t$ are all unobservable.

By log-linearization, we approximate (106) to (108) as

$$\widehat{y}_t = E_t\left[\widehat{y}_{t+1}\right] + \widehat{g}_t - E_t\left[\widehat{g}_{t+1}\right] - \frac{1}{\tau}\left(\widehat{R}_t - E_t\left[\widehat{\pi}_{t+1}\right] - E_t\left[\widehat{z}_{t+1}\right]\right), \tag{112}$$

$$\widehat{\pi}_t = \beta E_t\left[\widehat{\pi}_{t+1}\right] + \kappa\left(\widehat{y}_t - \widehat{g}_t\right), \tag{113}$$

$$\widehat{y}_t = \widehat{c}_t + \widehat{g}_t, \tag{114}$$

where $\kappa = \tau\frac{1-v}{v\pi^2\phi}$. Then equations (112) – (114) and (109) – (111) now constitute a linear rational expectation equation system (LRE).

The numerical solution of this LRE system takes the form

$$\boldsymbol{s}_t = \Phi_s\left(\boldsymbol{\theta}\right)\boldsymbol{s}_{t-1} + \Phi_\epsilon\left(\boldsymbol{\theta}\right)\boldsymbol{\epsilon}_t, \tag{115}$$

where $\boldsymbol{s}_t = \left[\hat{y}_t, \hat{c}_t, \hat{\pi}_t, \widehat{R}_t, \epsilon_{R,t}, \hat{g}_t, \hat{z}_t\right]'$ and $\boldsymbol{\theta}$ is the parameters which will be defined later.

Equation (115) is a state equation for unobservable state variable vector s_t. We define a set of measurement equations to relate the state variables to a set of observed variables:

$$
\begin{aligned}
YGR_t &= \gamma^{(Q)} + 100\left(\widehat{y}_t - \widehat{y}_{t-1} + \widehat{z}_t\right), && (116)\\
INFL_t &= \pi^{(A)} + 400\widehat{\pi}_t, && (117)\\
INT_t &= \pi^{(A)} + r^{(A)} + 4\gamma^{(Q)} + 400\widehat{R}_t, && (118)
\end{aligned}
$$

where YGR_t is the quarter-to-quarter per capita GDP growth rates, $INFL_t$ and INT_t are the annualized quarter-to-quarter inflation rates and the annualized quarter-to-quarter nominal interest rates, respectively. The parameters $\gamma^{(Q)}$, $\pi^{(A)}$ and $r^{(A)}$ are

$$\gamma = 1 + \frac{\gamma^{(Q)}}{100}, \beta = \frac{1}{1 + r^{(A)}/400}, \pi = 1 + \frac{\pi^{(A)}}{400},$$

where γ/β and π are the steady-states of $\widehat{R}_t$ and $\widehat{\pi}_t$, respectively. Note that Equations (116) – (118) can be reexpressed as

$$\boldsymbol{y}_t = \Psi_0(\theta) + \Psi_1(\theta)\boldsymbol{s}_t, \tag{119}$$

where $\boldsymbol{y}_t = (YGR_t, INFL_t, INT_t)'$ and $\boldsymbol{\theta} = \left[\tau, \kappa, \psi_1, \psi_2, \rho_R, \rho_g, \rho_z, r^{(A)}, \pi^{(A)}, \gamma^{(Q)}, \sigma_R, \sigma_g, \sigma_z\right]'$. Then equation (115) and (119) cast the DSGE model into a SSM.

2.2.7.3 Bayesian estimation of DSGE model The new Keynesian DSGE model can be described as a SSM with six state equations in (115) and three measurement equations in (119). While in principle the maximum likelihood method can be used to estimate the DSGE model via the Kalman filter, Fernández-Villaverde (2010) pointed out that the likelihood function of DSGE model is full of local maxima and minima and has a flat surface. Hence, the results from the optimization of likelihood is not reliable. For these reasons, Bayesian MCMC methods have become the standard approach to estimate DSGE models nowadays.

The idea behind MCMC is to simulate a Markov chain whose equilibrium distribution is $p\left(\boldsymbol{\theta}|\boldsymbol{y},M\right)$. In the Metropolis-Hastings Algorithm, the transition kernel $q\left(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}\right)$ is used to generate a proposed new value, $\boldsymbol{\theta}^{cand}$ for the chain. $\boldsymbol{\theta}^{cand}$ is accepted as the new state randomly with a particular probability.

The Random Walk Metropolis-Hastings (RWMH) algorithm for Bayesian DSGE estimation is proposed by Schorfheide (2000). An and Schorfheide (2007) used transition mixtures to deal with a multi-modal posterior distribution. Chib and Ramamurthy (2010) proposed to use a Metropolis-within-Gibbs algorithm that cycles over multiple, randomly selected blocks of parameters. Strid et al. (2010) proposed an adaptive hybrid Metropolis-Hastings samplers and Herbst (2010) developed a Metropolis-within-Gibbs algorithm that uses the information from the Hessian matrix to construct parameter blocks that maximize within-block correlations at each iteration.

2.2.7.4 Forecasting To forecast, note that the predictive distribution is defined as

$$p\left(\boldsymbol{y}_{T+1:T+H}|\boldsymbol{y}^{T}\right)=\int p\left(\boldsymbol{y}_{T+1:T+H}|\boldsymbol{\theta},\boldsymbol{y}^{T}\right)p\left(\boldsymbol{\theta}|\boldsymbol{y}^{T}\right)d\boldsymbol{\theta}, \tag{120}$$

where $\boldsymbol{y}_{T+1:T+H}$ denotes $\left(\boldsymbol{y}_{T+1},\boldsymbol{y}_{T+2},\ldots,\boldsymbol{y}_{T+H}\right)$. From (120), we can generate draws from the predictive distribution as

$$\begin{aligned} & p\left(\boldsymbol{y}_{T+1:T+H}|\boldsymbol{y}^{T}\right) \\ & \quad=\int_{(\boldsymbol{s}_T,\boldsymbol{\theta})}\left[\int_{\boldsymbol{s}_{T+1:T+H}}p\left(\boldsymbol{y}_{T+1:T+H}|\boldsymbol{s}_{T+1:T+H}\right)p\left(\boldsymbol{s}_{T+1:T+H}|\boldsymbol{s}_T,\boldsymbol{\theta},\boldsymbol{y}^{T}\right)d\boldsymbol{s}_{T+1:T+H}\right]. \\ & \qquad\times p\left(\boldsymbol{s}_T|\boldsymbol{\theta},\boldsymbol{y}^{T}\right)p\left(\boldsymbol{\theta}|\boldsymbol{y}^{T}\right)d\left(\boldsymbol{s}_T,\boldsymbol{\theta}\right)\end{aligned} \tag{121}$$

2.3 Lag length and model specification techniques

Given that so many alternative models can be used to generate forecasts, it is important to know which model should be used. In some cases, choice of model specification is amount to choice of lag length. In this section we discuss some techniques to choose lag length and model specification.

Algorithm 5 Draw from the Predictive Distribution

```
1. For $j = 1, 2, \ldots, N_{sim}$:
    (a) Draw $\boldsymbol{\theta}^j$ from $p\left(\boldsymbol{\theta}|\boldsymbol{y}^T\right)$;
    (b) Draw $\boldsymbol{s}_T^j$ from $p\left(\boldsymbol{s}_T|\boldsymbol{\theta}^j\right)$ which can be computed by Kalman filter;
    (c) Draw $\boldsymbol{\epsilon}_{T+1:T+H}^j$ from a multivariate normal distribution with mean $\mathbf{0}$ and a diagonal covariance matrix which has $\sigma_R^j$, $\sigma_z^j$ and $\sigma_g^j$;
    (d) Compute $\boldsymbol{s}_{T+1:T+H}^j$ from state equation (27) with $\boldsymbol{\theta}^j$ and $\boldsymbol{s}_T^j$;
    (e) Compute $\boldsymbol{y}_{T+1:T+H}^j$ from (26) with $\boldsymbol{\theta}^j$ and $\boldsymbol{s}_{T+1:T+H}^j$;
2. For $h = 1, 2, \ldots, H$, the $h$-step-ahead prediction of $\boldsymbol{y}_{T+h}$ is
       $\hat{\boldsymbol{y}}_{T+h,T} = \frac{1}{M}\sum_{j=1}^{M} \boldsymbol{y}_{T+h}^j.$
```

2.3.1 Akaike information criterion

One of the most widely used model selection method is the Akaike information criterion (AIC) proposed by Akaike (1973). This method generally involves calculating AIC for all of the candidate models and ranking the criterion functions accordingly. One model is selected at the end of this procedure, which generates the term “model selection”.

AIC is an asymptotically unbiased estimator of the Kullback-Leibler (K-L) distance between the true DGP and the predictive density of the candidate model. The AIC method can be applied to select model only when the likelihood functions for all candidate models are available. Given the linear model (1) with homoscedastic error term, the AIC typically takes the following functional form:

$$\text{AIC}_m = -2\hat{\ell}_{T,m} + 2d_m,, \tag{122}$$

where $\hat{\ell}_{T,m}$ is the maximized likelihood values of a candidate model $\mathcal{M}_m$ with d_m predictors. The first term in Equation (122) is to measure the model fit and the second term is the penalty for model complexity. The best model is selected by picking the one with the lowest value of (122).

Hurich and Tsai (1989) proposed the exact estimator of the K-L distance between the true DGP and the predictive density of the candidate model for the linear Gaussian regression models. The so-called finite-sample corrected AIC (AIC$_c$) is

$$\text{AIC}_{c,m} = \text{AIC}_m + \frac{2(d_m+1)(d_m+2)}{T-d_m-2}. \tag{123}$$

In practice, the AIC$_c$ method tends to have better finite-sample performance than the conventional AIC method under the assumption that the true DGP is linear Gaussian and the error term is iid.

A correctly specified model is a model that is the same as true DGP with some appropriate parameter values. AIC type information criterion achieves asymptotic efficiency, in the sense that their predictive performance are asymptotically equivalent to the best offered by the candidate models, when the candidate model set contains no more than one correctly specified model (Ding et al., 2019).

2.3.2 Mallow's C$_p$

Mallow's $\mathbf{C}_p$ (Mallows, 1973) provides an asymptotically unbiased estimator of the mean squared forecast error (MSFE) for a candidate model $\mathcal{M}_m$ with d_m predictors:

$$\mathbf{C}_{p,m} = \frac{1}{T}\left(\text{SSR}_m + 2d_m\hat{\sigma}^2\right), \tag{124}$$

where $\hat{\boldsymbol{\beta}}$ is the estimator of $\boldsymbol{\beta}$ and $\hat{\sigma}^2$ is a consistency estimator of the variance σ^2. In practice, $\hat{\sigma}^2$ is usually obtained by the largest model that includes all the potential predictors. Similar to AIC, we choose the best model by picking the one with the lowest value of (124). Note that Mallow's $\mathbf{C}_p$ is often used as a stopping rule for stepwise regression which we will discuss later.

2.3.3 Bayesian information criterion

BIC by Schwarz (1978) takes the following form:

$$\mathrm{BIC}_m = -2\hat{\ell}_{T,m} + d_m \log T. \tag{125}$$

Comparing to AIC, the penalty coefficient is replaced by the logarithm of the sample size instead of 2.

A model selection procedure is consistent if the true DGP is selected with probability approaching one as the sample size goes to infinity under the assumption that the true DGP is in the candidate model set. Consistency is important if our aim is to identify the true DGP. BIC is consistent. On the other hand, AIC chooses a larger model than true DGP with a positive probability as the sample size goes to infinity.

2.3.4 Hannan-Quinn information criterion

Hannan and Quinn (1979) proposed the HQ information criterion to select the order of autoregressive model:

$$\mathrm{HQ}_m = -2\hat{\ell}_{T,m} + d_m \log\log T. \tag{126}$$

If the true DGP is a fixed order autoregressive model, HQ is consistent. Note that the penalty term is $d_m \log\log T$ which is usually a very small number in order to guarantee consistency. However, when no fixed-dimension true model exists, neither BIC nor HQ is efficient (Shao, 1997).

2.3.5 Deviance information criterion

An information criterion based on a Bayesian estimator is the deviance information criterion (DIC) proposed by Spiegelhalter et al. (2002) and justified by Li et al. (2019). Let $\mathbf{y} = (y_1, \ldots, y_T)^\top$ be the data and $\boldsymbol{\theta}$ be the model parameters. Denote $D(\boldsymbol{\theta}) = -2 \ln p(\mathbf{y}|\boldsymbol{\theta})$, the DIC statistic of Spiegelhalter et al. (2002) is given by

$$\text{DIC} = D\left(\bar{\boldsymbol{\theta}}\right) + 2P_D, \tag{127}$$

where $\bar{\boldsymbol{\theta}}$ is the posterior mean of $\boldsymbol{\theta}$, and P_D, known as "effective number of parameters", is given by:

$$P_D = -2 \int [\ln p(\mathbf{y}|\boldsymbol{\theta}) - \ln p(\mathbf{y}|\bar{\boldsymbol{\theta}})] p(\boldsymbol{\theta}|\mathbf{y}) \mathrm{d}\boldsymbol{\theta}. \tag{128}$$

Spiegelhalter et al. (2002) interprets $D\left(\bar{\boldsymbol{\theta}}\right)$ as the Bayesian measure of model fit and P_D as the penalty term to measure model complexity.

Under some regularity conditions, Li et al. (2019) showed that $D\left(\bar{\boldsymbol{\theta}}\right) + 2\hat{\ell}_{T,m} = o_p(1)$ and $P_D - d_m = o_p(1)$, where $\hat{\ell}_{T,m}$ and d_m are defined in Equation (122). Hence, DIC can be understood as a Bayesian version of AIC.

2.3.6 Cross-validation

2.3.6.1 Prediction errors Our discussion mainly follows Hastie et al. (2009). Before we discuss cross-validation (CV) methods, we need to first introduce some concepts on various types of prediction error. Given a training set $\Omega = \{(x_1, y_1), (x_2, y_2), \ldots (x_T, y_T)\}$, the extra-sample error of a predictive function $\hat{f}$ is defined as

$$\text{Err}_{extra} = \mathbb{E}_{X^0, Y^0} \left[L(Y^0, \hat{f}(X^0)) | \Omega \right], \tag{129}$$

where L is a loss function, for instance, the square loss function $L(Y^0, \hat{f}(X^0)) = (Y^0 - \hat{f}(X^0))^2$, and (X^0, Y^0) is a new point (or a vector) drawn from the same distribution as

Ω. Notation "extra" means that input vector X^0 does not need to coincide with $\boldsymbol{x} = \{x_1, x_2, \ldots, x_T\}$. If $(\boldsymbol{X}^0, \boldsymbol{Y}^0)$ is a N-dimension vector, Equation (129) takes the form

$$\mathrm{Err}_{extra} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{X}^0, \boldsymbol{Y}^0}\left[L(Y_i^0, \hat{f}(X_i^0))|\Omega\right].$$

Note that Equation (129) is a conditional expectation depending on Ω. We can define the expected extra-sample error as

$$\mathcal{R}_{extra} = \mathbb{E}_{\Omega}\mathbb{E}_{\boldsymbol{X}^0, \boldsymbol{Y}^0}\left[L(\boldsymbol{Y}^0, \hat{f}(\boldsymbol{X}^0))|\Omega\right], \tag{130}$$

where the above expectation is taken with respect to the distribution of Ω.

If the output of the new data, Y^0, is generated following $Y_i^0 = f(x_i) + \epsilon_i^0$ with x_i being the input variable for function $f(\cdot)$ and ϵ_i^0 being the ith new error term, we can define the in-sample error as

$$\mathrm{Err}_{in} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{Y^0}\left[L(Y_i^0, \hat{f}(x_i))|\Omega\right]. \tag{131}$$

Then the expected in-sample error is simply

$$\mathcal{R}_{in} = \mathbb{E}_{\mathrm{y}}(\mathrm{Err}_{in}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\mathrm{y}}\mathbb{E}_{Y^0}\left[L(Y_i^0, \hat{f}(x_i))|\Omega\right]. \tag{132}$$

The training error is usually defined as

$$\overline{\mathrm{err}} = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \hat{f}(x_i)\right). \tag{133}$$

The difference between Err_{in} and $\overline{\mathrm{err}}$ is usually called the optimism (denoted as op):

$$\mathrm{op} = \mathrm{Err}_{in} - \overline{\mathrm{err}}, \tag{134}$$

and the expectation of op is defined as

$$\omega = \mathbb{E}_{\mathrm{y}}(\mathrm{op}). \tag{135}$$

Henceforth, from definitions (132), (134) and (135), the expected in-sample error can be expressed as

$$\mathbb{E}_{\mathrm{y}}(\mathrm{Err}_{in}) = \mathbb{E}_{\mathrm{y}}(\overline{\mathrm{err}}) + \omega. \tag{136}$$

If we use $\mathcal{R}_{in}$ as a criterion for model selection purposes, an estimator of $\mathcal{R}_{in}$ takes the form

$$\hat{\mathcal{R}}_{in} = \overline{\mathrm{err}} + \hat{\omega}, \tag{137}$$

where $\hat{\omega}$ is an estimator of ω defined in (135). Such general estimator can take various forms and can adopt many popular model selection methods such as AIC, BIC, Mallows C_p among others.

On the other hand, a typical CV method uses $\mathcal{R}_{extra}$ as model selection measure. Similar to the estimator of $\mathcal{R}_{in}$, the estimator of $\mathcal{R}_{extra}$ also takes various forms and can adopt nonparametric loss functions or machine learning techniques.

2.3.6.2 *K*-fold cross-validation A conventional validation approach is to split the data set into two parts. One part is called the “training set”, which we use to estimate the model. The other part is the “validation set”, of which the data is used to evaluate the estimated model from the training set. But such approach has two main issues: first, different data split leads to different result; second, only a subset of data is used to estimate the model, which leads to substantial loss in information.

The K-fold CV, on the other hand, circumvents the issues caused by the conventional validation approach. A typical K-fold CV randomly splits the data into K subsets of approximately equal size. The k^{th} part is treated as the validation set, and the other $K-1$ parts (together) are used as the training set to first estimate the model then evaluate the k^{th} part.

For example, we can use the estimated model by the training set to predict the validation set and estimate the corresponding MSFE. We repeat the above process for $k = 1, 2, \ldots, K$, which results in the K-fold CV estimates of the expected extra-sample prediction error

$$\mathrm{CV}(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(x_i)}), \tag{138}$$

where $\kappa : \{1, 2, \ldots, N\} \rightarrow \{1, \ldots, K\}$ is a index function indicates the number of folds each observation belongs to and $\hat{f}^{-k}(x)$ denote the fitted model with the k-th part of the data removed. If $K = T$ which means that each fold contains exactly one observation, this is the leave-one-out cross validation (LOOCV).

A key point in CV is the choice of K. For example, LOOCV only removes one point each time, it is an approximately unbiased estimator of $\mathcal{R}_{extra}$. But the correlation between the estimators from different folds is large because there are potentially many repeated observations in the training sets of different folds. Another problem of LOOCV is the high computation cost. Numbers like $K = 5$ and $K = 10$ are popular among practitioners.

2.4 Dimension reduction techniques

2.4.1 Principal component regression

The principal component regression (PCR) is a regression analysis technique that is based on principal component analysis (PCA). PCA was first proposed in 1901 by Karl Pearson. It is a statistical procedure that converts a set of possibly correlated variables into a set of linearly uncorrelated variables (named principal components). In practice, PCA is often implemented by the eigenvalue decomposition of the sample covariance matrix or the sample correlation matrix of data or singular value decomposition of the data matrix.

Given the standardized data matrix $\boldsymbol{X} = [X_1, X_2, \ldots]$ with dimension $T \times p$ (here we allow $p \gg T$). The PCA reconstructs $\boldsymbol{X}$ by

$$\boldsymbol{X} = \mathbf{Z}\boldsymbol{\alpha}^{\top}, \tag{139}$$

where $\mathbf{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_p]$ is the principal component matrix and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_p]$ is the loading matrix. Equation (139) can be rewritten as

$$\boldsymbol{X} = \sum_{j=1}^{p} \boldsymbol{Z}_j \boldsymbol{\alpha}_j^{\top}. \tag{140}$$

It is straightforward to show that each principal component $\boldsymbol{Z}_j = \boldsymbol{X}\boldsymbol{\alpha}_j$.

Suppose the sample covariance matrix up to sample size T is defined as $\boldsymbol{\Sigma} = \boldsymbol{X}^{\top}\boldsymbol{X}$. Then the sample variance of $\boldsymbol{Z}_j$ is $\boldsymbol{\alpha}_j^{\top}\boldsymbol{\Sigma}\boldsymbol{\alpha}_j$. It can be shown that the j^{th} loading vector $\boldsymbol{\alpha}_j$ is the eigenvector corresponding to the j^{th} large eigenvalue of $\boldsymbol{\Sigma}$, λ_j. Then we have $\boldsymbol{\alpha}_j^{\top}\boldsymbol{\Sigma}\boldsymbol{\alpha}_j = \lambda_j^2$ where $\lambda_1 > \lambda_2 > \ldots > \lambda_p$. For better understanding, we describe PCA in Algorithm 6. In each iteration k, PCA tries to find the k^{th} loading vector α_k which maximizes the sample variance $\boldsymbol{\alpha}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{\alpha}$ with the constraints $\boldsymbol{\alpha}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{\alpha}_i = 0$ for $i = 1, 2, \ldots, k-1$.

Algorithm 6 Principal Component Analysis

1. For $k = 1$, the first principal component is computed as

$$\boldsymbol{\alpha}_1 = \arg\max_{\boldsymbol{\alpha} \in \mathbb{R}^p} \; \boldsymbol{\alpha}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{\alpha}$$

   subject to $\boldsymbol{\alpha}^{\top}\boldsymbol{\alpha} = 1$.

2. For $k = 2, 3, \ldots, p$, the k-th component is computed as

$$\boldsymbol{\alpha}_k = \arg\max_{\alpha \in \mathbb{R}^p} \boldsymbol{\alpha}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{\alpha}$$

   subject to $\boldsymbol{\alpha}^{\top}\boldsymbol{\alpha} = 1$ and $\boldsymbol{\alpha}^{\top}\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{\alpha}_i = 0$ for $i = 1, 2, \ldots, k-1$.

3. Select the first K components for dimension reduction.

The main idea of PCR is to replace the p columns in the data matrix $\boldsymbol{X}$ with their uncorrelated K principal components from PCA. That is, we regress $\boldsymbol{y}$ on the first K PC's $\mathbf{Z}^K = [Z_1, Z_2, \dots, Z_K]$ by OLS,

$$\boldsymbol{y} = \mathbf{Z}^K \boldsymbol{\beta}^{\mathbf{Z^K}} + \boldsymbol{\epsilon}. \tag{141}$$

The PCR estimator is $\hat{\boldsymbol{\beta}}^{\mathbf{Z^K}} = \left(\mathbf{Z}^{K\top}\mathbf{Z}^K\right)^{-1}\mathbf{Z}^{K\top}\boldsymbol{y}$. Suppose the loading matrix for the first K components is $\boldsymbol{\alpha}^K = [\alpha_1, \alpha_2, \dots, \alpha_K]$. Then we have

$$\mathbf{Z}^K = \boldsymbol{X}\boldsymbol{\alpha}^K. \tag{142}$$

From (142), we can rewrite (141) as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\alpha}^K\boldsymbol{\beta}^{\mathbf{Z^K}} + \epsilon = \boldsymbol{X}\boldsymbol{\beta}^{PCR} + \boldsymbol{\epsilon}. \tag{143}$$

Then we have $\hat{\boldsymbol{\beta}}^{PCR} = \boldsymbol{\alpha}^K\left(\mathbf{Z}^{K\top}\mathbf{Z}^K\right)^{-1}\mathbf{Z}^{K\top}\boldsymbol{y}$. Equation (143) is more convenient to use for prediction. Given a new set of observations $\boldsymbol{x}^{new}$ $(p \times 1)$, the prediction for $\boldsymbol{y}^{new}$ is simply

$$\widehat{\boldsymbol{y}}^{new} = \left(\boldsymbol{x}^{new}\right)^\top \hat{\boldsymbol{\beta}}^{PCR}.$$

2.4.2 Partial least squares regression

The forecasting performance of PCR depends on two assumptions. First, a small number of components from PCA explain most variation in $\boldsymbol{X}$. Second, the first few components are the most relevant to the response variable $\boldsymbol{y}$. The objective of PCA is to find the components which can explain the variation in $\boldsymbol{X}$ as much as possible. However, these selected components may not be correlated with $\mathbf{y}$. In fact, PCA is usually categorized as an unsupervised learning method that does not consider the response variable $\boldsymbol{y}$ during its execution.

Partial least squares (PLS), on the other hand, incorporates the information from $\boldsymbol{y}$ to decompose the $\boldsymbol{X}$ matrix. The PLS method was originated in Wold's nonlinear iterative partial least squares algorithm (Wold, 1966). To explain PLS, suppose we have

$$\boldsymbol{X} = \mathbf{L}^K \boldsymbol{\Gamma}^{K\top} + \mathbf{E} \tag{144}$$

where $\mathbf{L}^K = [\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_K]$ is a $T \times K$ latent (unobserved) component matrix, $\boldsymbol{\Gamma}^K = [\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \ldots, \boldsymbol{\Gamma}_K]$ is the $p \times K$ loading matrix, $\mathbf{E}$ is the $T \times p$ error matrix and $K < p$ is the number of latent components we need.

Given Equation (144), the k^{th} latent component $\boldsymbol{L}_k$ can be expressed as $\boldsymbol{L}_k = \boldsymbol{X}\Gamma_k$. Let the sample covariance of $\boldsymbol{L}_k$ and the response variable $\boldsymbol{y}$ be $\boldsymbol{\alpha}_k^\top \boldsymbol{X}^\top \boldsymbol{y}$, where $\boldsymbol{\alpha}_k$ is some coefficient vector we want to estimate. The basic idea of PLS is to estimate $\boldsymbol{\alpha}_k$ by maximizing the square of the covariance $\boldsymbol{\alpha}_k^\top \boldsymbol{X}^\top \boldsymbol{y}$ in an iterative fashion. The associated estimation algorithm is presented in Algorithm 7.

Algorithm 7 Partial Least Squares

1. For $k = 1$, the first principal component is computed as

   $$\boldsymbol{\alpha}_1 = \underset{\boldsymbol{\alpha} \in \mathbb{R}^p}{\arg\max} \ \boldsymbol{\alpha}^\top \boldsymbol{X}^\top \boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\alpha}$$

   subject to $\boldsymbol{\alpha}^\top \boldsymbol{\alpha} = 1$.

2. For $k = 2, 3, \ldots, p$, the k-th component is computed as

   $$\boldsymbol{\alpha}_k = \underset{\alpha \in \mathbb{R}^p}{\arg\max} \ \boldsymbol{\alpha}^\top \boldsymbol{X}^\top \boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\alpha}$$

   subject to $\boldsymbol{\alpha}^\top \boldsymbol{\alpha} = 1$ and $\boldsymbol{\alpha}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\alpha}_i = 0$ for $i = 1, 2, \ldots, k-1$.

3. Select the first K latent components for dimension reduction.

Different from PCA, PLS maximizes the covariance between the latent components and response variable $\boldsymbol{y}$ subject to constraints. This results in higher predictive power of the latent components on $\boldsymbol{y}$ comparing to PCA. Similar to PCR, the partial least squares regression (PLSR) simply regresses the response variable $\boldsymbol{y}$ on the K latent components $\mathbf{L}^K$

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\Gamma}^K \boldsymbol{\beta}^{\mathbf{L}^\mathbf{K}} + \boldsymbol{\epsilon} = \boldsymbol{X}\boldsymbol{\beta}^{PLSR} + \boldsymbol{\epsilon}, \tag{145}$$

where $\hat{\boldsymbol{\beta}}^{PLSR} = \boldsymbol{\Gamma}^K \boldsymbol{\beta}^{\mathbf{L}^\mathbf{K}}$ with $\boldsymbol{\beta}^{\mathbf{L}^\mathbf{K}} = \left(\mathbf{L}^{K^\top}\mathbf{L}^K\right)^{-1}\left(\mathbf{L}^K\right)^\top \boldsymbol{y}$.

2.5 Model averaging

The model selection methods aim to select the best model from the candidate set and use it to make inference or predict future values. An alternative to the strategy of selecting the best available model is the model averaging, that is to average the estimators or predictions from a collection of plausible models. We do so because we acknowledge that there is no best model and all models are somewhat useful. In this section, we review Bayesian and frequentist model averaging methods. Following Ding et al. (2019), use $\mathcal{M}_m = \{p_\theta : \theta_m \in \Theta_m\}$ to denote a model, and $\{\mathcal{M}_m\}_{m \in \mathbb{M}}$, is the candidate model set index by $m \in \mathbb{M}$, the dimension of $\mathcal{M}_m$ is d_m.

2.5.1 Bayesian model averaging (BMA)

Given a model $m \in \mathbb{M}$, we can obtain the posterior density $\pi\left(\theta_m|\boldsymbol{y}, \mathcal{M}_m\right)$ using Bayes's theorem

$$\pi\left(\theta_m|\boldsymbol{y}, \mathcal{M}_m\right) = \frac{p\left(\boldsymbol{y}|\theta_m, \mathcal{M}_m\right)\pi\left(\theta_m|\mathcal{M}_m\right)}{\int p\left(\boldsymbol{y}|\theta_m, \mathcal{M}_m\right)\pi\left(\theta_m|\mathcal{M}_m\right) d\theta_m}, \tag{146}$$

where $\pi\left(\theta_m|\mathcal{M}_m\right)$ is the prior density of θ_m given $\mathcal{M}_m$. The posterior model probability given the observed data, $\pi\left(\mathcal{M}_m|\boldsymbol{y}\right)$, is

$$\pi\left(\mathcal{M}_m|\boldsymbol{y}\right) = \frac{\pi\left(\boldsymbol{Y}|\mathcal{M}_m\right)\pi\left(\mathcal{M}_m\right)}{\sum_{m=1}^{M}\pi\left(\boldsymbol{y}|\mathcal{M}_m\right)\pi\left(\mathcal{M}_m\right)}, \tag{147}$$

where $\pi\left(\mathcal{M}_m\right)$ is the prior probability of $\mathcal{M}_m$ and $\pi\left(\boldsymbol{y}|\mathcal{M}_m\right) = \int p\left(\boldsymbol{y}|\theta_m, \mathcal{M}_m\right)\pi\left(\theta_m|\mathcal{M}_m\right) d\theta_m$ denotes the marginal likelihood of model $\mathcal{M}_m$.

$\pi\left(\mathcal{M}_m|\boldsymbol{y}\right)$ can be used either for model selection directly or for the weight of model averaging. Suppose the probability of future value y^{new} given observed data and model $\mathcal{M}_m$ is

$$\pi\left(y^{new}|\boldsymbol{y},\mathcal{M}_m\right)=\int p\left(y^{new}|\theta_m,\mathcal{M}_m\right)\pi\left(\theta_m|\boldsymbol{y},\mathcal{M}_m\right)d\theta_m. \tag{148}$$

Then the BMA prediction density of y^{new} is given by

$$\pi(y^{new}|\boldsymbol{y})=\sum_{m=1}^{M}\pi\left(y^{new}|\boldsymbol{y},\mathcal{M}_m\right)\pi\left(\mathcal{M}_m|\boldsymbol{y}\right), \tag{149}$$

which is an average of the predictions from different models weighted by the posterior model probabilities. For more details, see Fragoso et al. (2018).

2.5.2 Frequentist model averaging (FMA)

Let $\hat{f}_m$ denotes the prediction from the m^{th} model, and w_m to be the weight for model $\mathcal{M}_m$, where $m=1,2,\ldots,M$. Then set $\mathbf{w}=(w_1,w_2,\ldots,w_M)$ to be the vector of weights satisfies $0\le w_m\le 1$ and $\sum_{m=1}^{M}w_m=1$. Hence we define the FMA estimator as

$$\hat{f}(\mathbf{w})=\sum_{m=1}^{M}w_m\hat{f}_m. \tag{150}$$

In the frequentist framework, information criteria such as BIC and AIC can be used to construct $\mathbf{w}$. For BIC

$$w_m^{\text{BIC}}=\frac{\exp\left(-\frac{1}{2}\text{BIC}_m\right)}{\sum_{j=1}^{M}\exp\left(-\frac{1}{2}\text{BIC}_j\right)}, \tag{151}$$

and for AIC

$$w_m^{\text{AIC}}=\frac{\exp\left(-\frac{1}{2}\text{AIC}_m\right)}{\sum_{j=1}^{M}\exp\left(-\frac{1}{2}\text{AIC}_j\right)}, \tag{152}$$

where BIC_m and AIC_m denote the BIC and AIC values of the m^{th} model.

3 Machine Learning Methods

Unlike econometric methods, machine learning methods often do not rely on the assumption of true DGP. Even under the true DGP, the relationship between the output and input variables can be highly complicated, involving high levels of nonlinearity and interactive terms, which cannot be described by an analytic function. Henceforth, the primary target of most machine learning methods is not to find a DGP or anything related to the discovery of the true DGP, such as properties of parameter estimation, statistical inference of parameters and specification test of a candidate model. Most of them are algorithm-based. Often the primary target of machine learning methods is the prediction. Clearly, as far as the prediction is concerned, machine learning methods can pose a great deal of challenges to conventional econometric methods, as shown in several recent studies; see Biau and D'elia (2010), Jung et al. (2019), and Chuku et al. (2019) in forecasting GDP growth rates, Tiffin (2016) in nowcasting GDP growth rates, and Medeiros et al. (2019) in forecasting inflation.

In this section, we investigate the mechanism of some popular machine learning methods. We start with the multivariate adaptive regression splines. As the multivariate adaptive regression splines predetermines a number of choices such as choosing the number of knots and variable selection, we explain how to use the penalized regression techniques and variable selection techniques to make these choices. Most of these methods follow the linear formulation. Therefore, they are closely related to conventional econometric methods and have been studied extensively by econometricians. Forecasting based on these methods are quite similar to forecasting using econometric methods. One needs to estimate the coefficient (in vector form) first, then pre-multiply the input variables to the estimated coefficient vector to obtain the forecasts.

Then, we introduce five tree-based algorithms, including the regression tree, bagging tree, random forest, boosting tree, and the popular M5' algorithm. We also cover the basic concepts of neural networks and explain the working principles of support vector machine for regression. These methods do not impose linear restriction and rely on nonparametric algorithms or kernel tricks to formulate the model. Some methods (bagging tree, random forest, boosting, and M5') generate forecasts by aggregating forecasts from a series of learners and/or generated pseudo-data (via bootstrap), henceforth, are given the name ensemble methods.

3.1 Multivariate adaptive regression splines

As an adaptive procedure for regression, the MARS method excels at solving high dimensional problems caused by a large set of input variables. Following Friedman (1991), the MARS method uses expansions in piecewise linear basis functions of the form $(x-h)_+$[6] and $(h-x)_+$ such that

$$(x-h)_+ = \begin{cases} x-h & \text{if } x>h \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad (h-x)_+ = \begin{cases} h-x & \text{if } x<h \\ 0 & \text{otherwise} \end{cases},$$

where the two piecewise linear functions is a reflected pair with a knot at the value t. This concept is illustrated in Figure 1(a), in which we forecast the CBOE Volatility Index (VIX) with one-period lag of the logarithm of one-month crude oil futures contract (OIL) as the sole predictor. We fit the actual data (dots) with four piecewise linear regressions (solid lines), where the three knots are indicated by circle $\circ$ symbols. It is obvious that the MARS regression fits the actual data better than a simple linear regression.

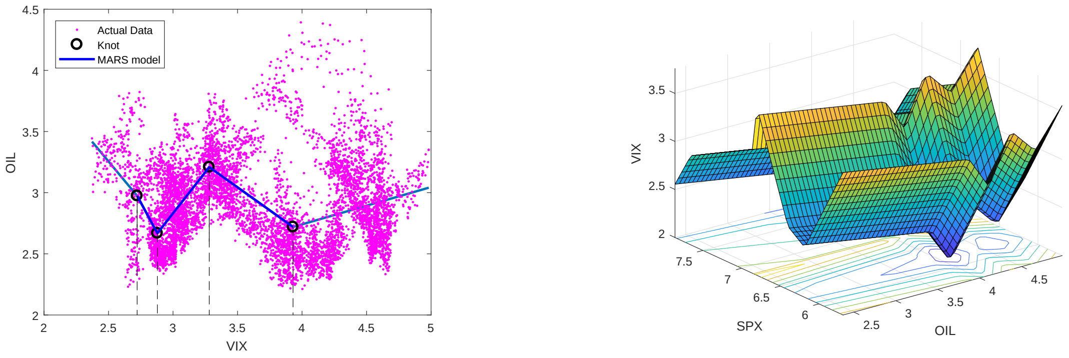
We form reflected pairs for each input $\boldsymbol{X}_j$ with knots at each observed value x_{tj} for $t=1,\ldots,n$ and $j=1,\ldots,k$. The collection of basis function is

$$\mathcal{F} = \left\{ (\boldsymbol{X}_j - h)_+,\ (h - \boldsymbol{X}_j)_+ \right\}, \tag{153}$$

where $h \in \{x_{1j},\ldots,x_{nj}\}$ for $j=1,\ldots,k$. The full data is separated into S subsamples by

[6] The term A_+ represents the positive part of A. It is also called a hinge function that can be expressed as $\max(0,A)$.

Figure 1: Using the MARS Method to Forecast VIX



(a) Forecast VIX with One Predictor

(b) Forecast VIX with Two Predictors

Note: Figure 1(a) depicts forecasting VIX with predictor OIL using the MARS method, where the actual data, piecewise linear regressions, and knots are represented by dots, solid lines, and circles, respectively. Figure 1(b) plots the estimated MARS regression splines for a large set of input variables in a 3D figure with OIL and SPX being the selected predictor. Lines on the OIL-SPX plane are the estimated knots.

the knots and a model is fitted locally to each subsample. We use functions from the set $\mathcal{F}$ and build a model of the form

$$y_t = f(\boldsymbol{X}_t) + \epsilon_t = \beta_0 + \sum_{s=1}^{S} \beta_s F_s(\boldsymbol{X}_t) + \epsilon_t, \tag{154}$$

where β_0 is a constant, the coefficients β_s are estimated by standard linear regression for each subsample, and each function $F_s(\boldsymbol{X}_t)$ is an element in the set $\mathcal{F}$ or a product of such elements. The key of the MARS method lies in the construction of the functions $F_s(\boldsymbol{X}_t)$.

The model-building procedure is carried out in two phases: the forward stage and the backward stage. In the forward stage, we start with a model consisting of just the constant term (the mean of y_t). We then repeatedly adds basis function in pairs as show in (153) to the model. We find the pair of basis functions that gives the maximum reduction in SSR. To add a new basis function, the MARS method search over all combinations of the following: (i) existing variables; (ii) all input variables; and (iii) all values of each input variable. At the end of the forward stage, we have a large model of the form (154).

Model (154) typically overfits the data, therefore, we apply a backward deletion stage to mitigate the overspecification. We remove terms one by one, deleting the least effective term whose removal causes the smallest increase in SSR at each step until we find the best model of each size (number of variables) λ. In practice, we estimate the optimal value of λ by minimizing the following generalized cross-validation criterion with constraint $\lambda \in \mathbb{Z}$,

$$\mathrm{GCV}(\lambda) = \frac{\sum_{t=1}^{n} \left(y_t - \hat{f}_\lambda(\boldsymbol{X}_t)\right)^2}{\left(1 - \tilde{\lambda}/n\right)^2},$$

where $\hat{f}_\lambda(\boldsymbol{X}_t)$ is the prediction of estimated best model based on size λ and $\tilde{\lambda}$ is the effective number of parameters in the model.[7]

As a demonstration, we extend the VIX forecasting exercise in Figure 1(a) by adding the logarithm of the S&P500 index (SPX) as a new predictor. We apply the MARS process discussed above and plot the estimated MARS regression splines in a 3D figure with OIL and SPX being the predictors in Figure 1(b). Note that, instead of being points in Figure 1(a), knots in a 3D figure are lines, as shown on the OIL-SPX plane.

3.2 Penalized regression

The placement of knots, the number of knots, and the degree of the polynomial can be seen as tuning parameters, which are subject to manipulation by a data analyst. The tuning process can be very complicated, since there are at least three of them that must be tuned simultaneously. Moreover, there is little or no formal theory to justify the tuning.

On the other hand, a useful alternative is to alter the fitting process itself so that the tuning is accomplished automatically, guided by clear statistical reasoning. One popular

[7] This number accounts for both the number of variables and the number of parameters used in selecting the optimal knots. If there are K knots in the forward process, the formula for $\tilde{\lambda}$ is

$$\tilde{\lambda} = \lambda + cK,$$

in which some simulation results suggest that one should set $c = 3$.

approach is to combine a mathematical penalty[8] with the loss function to be optimized. This leads to a very popular approach called penalized regression which has led a wide range applications in the machine learning literature.

Consider a conventional regression analysis with an indicator variable as the sole regressor. As the regression coefficient increases in absolute value, the resulting step function will have a step of increasing size. The difference between the conditional mean of y_t when the indicator is 0 compared to the conditional means of Y when the indicator is 1 is larger. The larger the regression coefficient the rougher the fitted values.

Strategies that are designed to control the magnitude of the coefficients are called shrinkage or regularization. Two popular proposals have been offered for how to control the complexity of the fitted values:

1. constrain the sum of the absolute values of the regression coefficients to be less than some constant C (sometimes called an L_1-penalty); and
2. constrain the sum of the squared regression coefficients to be less than some constant C (sometimes called an L_2-penalty).

In this section, we introduce the following popular penalized regression methods: ridge regression, least absolute shrinkage selective operator (LASSO), elastic net, and adaptive LASSO.

3.2.1 Ridge regression

Suppose that for a conventional fixed $\boldsymbol{X}$ regression, one adopts the constraint that the sum of the p squared regression coefficients is less than C. This constraint leads directly to

[8] The penalty imposes greater losses as a mean function becomes more complicated. For greater complexity to be accepted, the fit must be improved by an amount that is larger than the penalty. The greater complexity has to be worth it.

ridge regression.[9] The task is to obtain values for the regression coefficients so that

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} \left[\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right], \tag{155}$$

where $\boldsymbol{\beta} = [\beta_1, ..., \beta_p]^\top$ does not include β_0. In Equation (155), the usual expression for SSR has a new component - the sum of the squared regression coefficients multiplied by a constant λ. This is a L_2 penalty. Note that λ is a tuning parameter that determines how much weight is given to the penalty.

It follows that the ridge regression estimator is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \tag{156}$$

where $\boldsymbol{I}$ is a $p \times p$ identity matrix. Note that the column of 1s for the intercept is dropped from $\boldsymbol{X}$ and β_0 is estimated separately.[10] In Equation (156), the value of λ is added to the main diagonal of the cross-product matrix $\boldsymbol{X}^\top \boldsymbol{X}$, which determines how much the estimated regression coefficients are shrunk toward zero. With non-zero λ, the shrinkage becomes a new source of bias. However, while biased, the reduced variance of ridge estimates often result in a smaller mean square error when compared to least-squares estimates.

Ridge regression reduces mean squared error by the trade off between the prediction bias and variance. The common trend is the variance decreases and bias increases as λ increase. This can be illustrated by a simulation similar to Hastie et al. (2009). Here the data is simulated from a linear model with $T = 50$, $p = 30$, and the variance of the error term $\sigma^2 = 1$ with different coefficients 1) Case 1, 10 large coefficients (between 0.5 and 1), 20 small (between 0 and 0.3); 2) Case 2, 30 large coefficients (between 0.5 and 1); 3) Case 3, 10 large coefficients (between 0.5 and 1), 20 exactly 0. In all three cases, 50 data is

[9] In machine learning literature, it is sometimes called weight decay.

[10] By default, $\boldsymbol{\beta}$ is computed after centering and scaling the predictors to have mean 0 and standard deviation 1. The model does not include a constant term, and $\boldsymbol{X}$ should not contain a column of 1s.

Table 1: Linear Regression and Ridge Regression

	Linear Regression			Ridge Regression			
Case	Bias^2	Var	MSE	Bias^2	Var	MSE	λ^*
1	0.006	0.627	0.633	0.073	0.410	0.483	0.402
2	0.006	0.628	0.634	0.051	0.499	0.550	0.251
3	0.006	0.627	0.633	0.076	0.411	0.487	0.372

generated as training set and another 50 is generated as the test data. We use training data to estimate model then compare the forecasting performance of linear regression and ridge regression. In all three cases, ridge regression achieves lower mean squared error by the trade-off between increasing bias and reducing variance. Even in Case 2, ridge regression still outperforms linear regression.

Note that in the third case we presented, the true DGP requires certain coefficients to be exactly 0. However, different from the LASSO method that we are about to discuss, ridge coefficients are almost never shrunk to exactly 0. This partially explains the high biases induced by the ridge regression. On the other hand, the ridge method is not optimal in selecting variables and should not be used as a model selection device in practice.

3.2.2 LASSO regression

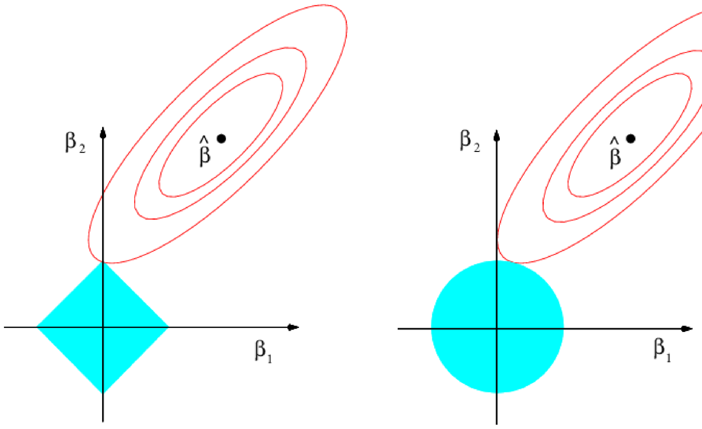
Suppose that one proceeds as in ridge regression but now adopts the constraint that the sum of the absolute values of the regression coefficients (L_1-penalty) is less than some constant. This leads to a regression procedure known as the LASSO (Tibshirani, 1996) whose estimated regression coefficients are defined by

$$\hat{\boldsymbol{\beta}} = \min_{\beta} \left[\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]. \tag{157}$$

Unlike the ridge penalty, the LASSO penalty leads to a nonlinear estimator, and a quadratic programming solution is needed. As before, the value of λ is a tuning parameter, a λ of zero yields the usual least squares results, and as the value of λ increases, the regression coefficients are shrunk toward zero.

Unlike the ridge regression, the LASSO regression is capable of shrinking coefficients to exactly 0 without setting $\lambda = \infty$. Therefore, it can be used as a variable selection tool in practice. This concept is illustrated geometrically as follows (James et al., 2013)

Figure 2: Visualization of the LASSO Regression and Ridge Regression



In Figure 2, the parameter $\boldsymbol{\beta}$ is two-dimensional and ellipses represent the contours of the residual sum of squares. The term $\hat{\boldsymbol{\beta}}$ represents the OLS estimator which is the unconstrained optimization solution. The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ for LASSO and $|\beta_1|^2 + |\beta_2|^2 \leq s$ for ridge. Clearly, if s is sufficiently large, the constraint regions will contain the OLS estimate. In this case, both the ridge regression and the LASSO regression are the same as the OLS estimates. Equations (155) and (157) indicate that the LASSO and ridge regression coefficients estimates are given by the first point at which an ellipse contacts the constraint region. Since ridge regression has a circular constraint with no sharp points, this intersection will not generally occur on an axis, and so the ridge regression coefficient estimates will be exclusively non-zero. On the other hand, as illustrate by the left subfigure of Figure 2, the ellipse will often intersect the constraint region at an axis, since the LASSO constraint has corners at each of the axes. When this occurs, one of the coefficients will zero.

Table 2: Linear Regression and LASSO Regression

	Linear Regression			LASSO Regression			
Case	Bias^2	Var	MSE	Bias^2	Var	MSE	λ^*
1	0.006	0.627	0.633	0.061	0.420	0.481	0.063
2	0.006	0.628	0.634	0.008	0.621	0.629	0.008
3	0.006	0.627	0.633	0.073	0.315	0.388	0.086

We replicate the simulation design in Section 3.2.2. We compare the LASSO regression with linear regression and present the results is in Table 2. LASSO achieves lower mean squared error in all three cases comparing to the linear regression. Combining the results in Tables 1 and 2, we note that in Case 3, LASSO outperforms both the linear and the ridge regression when there are many coefficients that are set to zero by DGP. In Case 2, however, since all the coefficients are nonzero and relatively large, the decrease in variance is lower than in other two cases. The performance of LASSO is similar to linear regression but worse than the ridge regression.

Results in Tables 1 and 2 show that ridge and LASSO achieve lower MSE than linear regression by the bias-variance trade-off but neither can universally dominate the other. The performance of different regularization methods depend on the structure of the data in practice.

3.2.3 Elastic net

Zou and Hastie (2005) pointed out that the LASSO solution paths are unstable when predictors are highly correlated. If there is a group of variables with strong correlations, the LASSO is indifferent among the predictor set. Zou and Hastie (2005) proposed the Elastic-Net as an improved version of the LASSO to overcome such limitation.

Following Zou and Hastie (2005), the Elastic Net is a regularization and variable selection procedure that makes use of the penalty which is a mixture of ridge and the LASSO penalties

$$\lambda \left[\frac{1}{2}(1-\alpha) \sum_{j=1}^{p} \boldsymbol{\beta}_j^2 + \alpha \sum_{j=1}^{p} \left| \boldsymbol{\beta}_j \right| \right], \tag{158}$$

where $\alpha \in [0,1]$ is called the mixing parameter and λ has the usual interpretation as in ridge and LASSO regression. The ℓ_1 part of the penalty in (158) implements variable selection and the ℓ_2 brings the grouping effect and stabilizes the ℓ_1 solution path. The Elastic Net includes ridge and the LASSO as its special case when $\alpha = 1$ and $\alpha = 0$ respectively.

Following Hastie et al. (2015), we introduce the following simulation study to illustrate the grouping effect in the Elastic Net and the LASSO. With sample size $n = 100$, independently generate two independent "hidden" (unobserved) factors $\boldsymbol{Z}_1$ and Z_2 from standardized normal distribution and construct the response vector $\boldsymbol{y}$ as

$$\boldsymbol{y} = 3Z_1 - 1.5Z_2 + 2\epsilon, \text{ with } \epsilon \sim N(0,1).$$

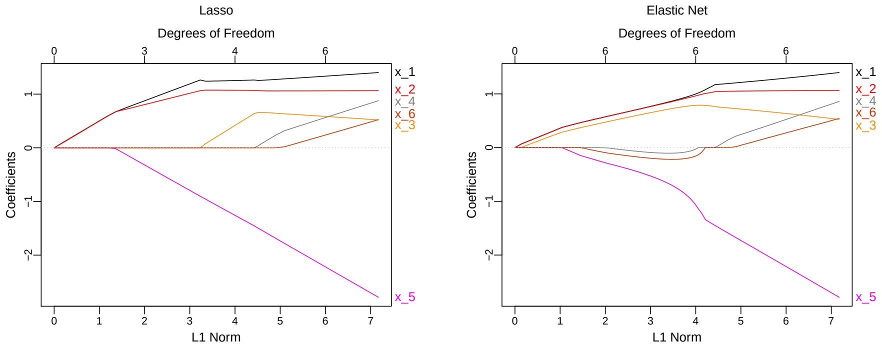
where Z_1 is a more important predictor since it is more relevant to y than Z_2. Suppose we can only observe predictors $\boldsymbol{X} = [X_1, X_2, \ldots, X_6]$ which are the approximates of Z_1 and Z_2:

$$X_j = Z_1 + \xi_j/5, \text{ with } \xi_j \sim N(0,1) \text{ for } j = 1,2,3,$$
$$X_j = Z_2 + \xi_j/5, \text{ with } \xi_j \sim N(0,1) \text{ for } j = 4,5,6.$$

If $X_1, X_2, \ldots, X_6$ are used to predict y, fit the model on (X, y) with the LASSO and the Elastic Net respectively. We expect the better method is able to pick up X_1, X_2 and X_3 as a group (the Z_1 group).

Figure 3 shows the results of the variable selection by the LASSO and the Elastic Net ($\alpha = 0.5$) as the norm of the coefficients increase (λ decreases). The left panel shows the results of the LASSO and right panel is for the Elastic Net. We can see that the Elastic Net is capable identify the Z_1 group as the most important variables when the norm of the coefficients is relatively small.

Figure 3: LASSO vs. Elastic Net



3.2.4 Adaptive LASSO

Fan and Li (2001) and Zou (2006) argued that the LASSO may not satisfy the oracle property which means to asymptotically identify the right subset model with probability converging to 1 and has the optimal estimation rate. Zou (2006) proposed the adaptive LASSO as a weighted version of the LASSO which satisfies the oracle property.

Following Zou (2006), the adaptive LASSO can be defined as follows

$$\hat{\boldsymbol{\beta}}^{\text{ada}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2T}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_n \sum_{j=1}^{p} w_j \left|\beta_j\right|, \tag{159}$$

where $\boldsymbol{w} = (w_1, w_2, \dots, w_p)^\top$ is a known vector of weights. In practice, $w_j = 1/\left|\hat{\beta}_j^{ini}\right|^\gamma$, where $\hat{\beta}_j^{ini}$ is a root-n estimator of β_j (this condition can be weakened) and $\gamma > 0$. Zou (2006) proved that the adaptive LASSO will satisfy the oracle property with some appropriate choice of λ_n.

The adaptive LASSO can be treated as a two-step method. The first step to fit the data with LASSO and get the parameter estimate $\hat{\boldsymbol{\beta}}^{LASSO}$ where the optimal tuning parameter value λ_{LASSO} is obtained by cross-validation. The second step is to plug $\hat{\boldsymbol{\beta}}^{LASSO}$ into (159) to get the adaptive LASSO estimator $\hat{\boldsymbol{\beta}}^{ada}$. By allowing a relatively higher penalty for zero coefficients and lower penalty for nonzero coefficients, the adaptive LASSO is designed to reduce the estimation bias and improve variable selection accuracy relative to the standard

LASSO approach.

The weighted ℓ_1 penalty and the Elastic Net penalty improve the LASSO in two different directions. The adaptive LASSO achieves the oracle property and the Elastic net handles the collinearity. The adaptive Elastic-Net proposed by Zou and Zhang (2009) is to combine the ideas of the weighted ℓ_1 penalty and the Elastic-net regularization to improve the LASSO in both directions

$$\hat{\boldsymbol{\beta}}^{\text{AdaEnet}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2T} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \left[\frac{1}{2}(1-\alpha) \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} w_j \left|\beta_j\right| \right],$$

where $w_j = \left(|\hat{\beta}_j^{\text{enet}}|\right)^{-\gamma}, j = 1, 2, \ldots, p$ and $\hat{\beta}_j^{\text{enet}}$ is the Elastic Net estimator.

3.3 Variable selection techniques

When a large set of input variables are available, we need to use a subset of best variables. The best subset is to compare all possible candidate models by using cross-validated prediction error, Mallows' C_p, AIC, BIC, or adjusted R^2. In practice, the candidate model set is first divided into $p + 1$ sub-groups by the number of predictors contained in each candidate model. Then the best model in k-th sub-group is selected as $\mathcal{M}_k$ and we choose the single best model from $\mathcal{M}_0, \ldots, \mathcal{M}_p$. Following James et al. (2013), the best subset method can be described as follows

Algorithm 8 Best Subset Selection

```
1. Let $\mathcal{M}_0$ denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$ :
    (a) Fit all $\begin{pmatrix} p \\ k \end{pmatrix}$ models that contain exactly $k$ predictors;
    (b) Pick the best among these $\begin{pmatrix} p \\ k \end{pmatrix}$ models, and call it $\mathcal{M}_k$ . Here best is defined as having $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, Mallows' $C_p$, AIC, BIC, or adjusted $R^2$.
```

Since in each sub-group, all the models contain the same number of predictors, we only need to compare the model fit. That's why R^2 is used within each sub-group but adjusted R^2 is used cross different subgroups.

Zou (2006) pointed out that best subset selection has two limitations. First, the computation cost is very high when p is very large, for example, if $p = 20$, we need to compare ten million models. Second, subset selection is extremely variable because of its inherent discreteness (Fan and Li, 2001).

3.3.1 Forward step selection

Following James et al. (2013), forward stepwise selection starts with an empty model that contains no predictors, then adds predictors to the model in an iterative fashion until all of the predictors are included in the model. The whole process can be described in the following algorithm.

Algorithm 9 Forward Step Selection

```
1. Let $\mathcal{M}_0$ denote the null model, which contains no predictors.
2. For $k = 0, \ldots, p-1$ :
    (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor;
    (b) Choose the best among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here best is defined as having highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p(\text{AIC}), \text{BIC}$, or adjusted $R^2$.
```

Comparing with the best subset selection method, in each sub-group we only need to compare $p-k$ models not 2^k. This substantially reduces the computational cost. On the other hand, due to its limited candidate model set, the forward step selection may not pick the best possible model out of all the 2^p candidate models.

3.3.2 Backward stepwise selection

Like the forward stepwise selection, the backward stepwise selection provides an efficient alternative to the best subset selection. It starts with the full model incorporating all predictors, and then iteratively removes the least useful predictor one-at-a-time. Similar to the forward stepwise selection, there are also $p-k$ models in each sub-group for the backward stepwise selection. Following James et al. (2013), the method can be described as the following:

Algorithm 10 Backward Stepwise Selection

```
1. Let $\mathcal{M}_p$ denote the full model, which contains all $p$ predictors.
2. For $k = p, p-1, \ldots, 1$ :
    (a) Consider all $k$ models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors;
    (b) Choose the best among these $k$ models, and call it $\mathcal{M}_{k-1}$. Here best is defined as having highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p(\text{AIC}), \text{BIC}$, or adjusted $R^2$.
```

The backward selection approach searches through $1 + p(p+1)/2$ models, which is much smaller than 2^k even for large p. However, similar to the forward stepwise selection, the backward stepwise selection is also not guaranteed to yield the best possible model. For instance, suppose that with $p = 3$, the best two-variable model incorporates X_2 and X_3, but the overall best possible model contains only X_1. Then, the backward stepwise selection fails to select the best possible model since X_1 is dropped in the first step.

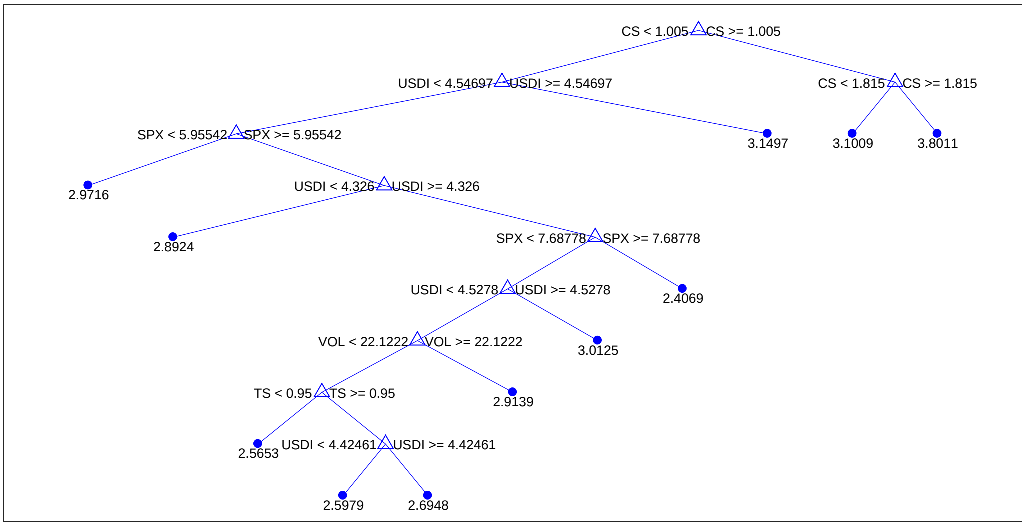
3.4 Regression tree

Breiman et al. (1984) proposed the Classification and Regression Trees (CART) method, in which Classification mostly deals with the categorical response of nonnumerical symbols and texts and Regression Trees concentrate purely on quantitative responses variables. Given the numerical nature of our data set, we only consider the second part of CART.

The trick in applying RT is to find the best split. Consider a sample of $\{y_t, x_t\}_{t=1}^n$. A simple regression will yield a sum of squared residuals, SSR_0. Suppose we can split the original sample into two sub-samples such that $n = n_1 + n_2$. The RT method finds the best split of a sample to minimize the sum of squared residuals (SSR) from the two sub-samples.[11] That is, the SSR values computed from each sub-sample should follow: $\text{SSR}_1 + \text{SSR}_2 \leq \text{SSR}_0$. We can continue splitting until we reach a pre-determined boundary.

A representation of applying RT on VIX forecasting is depicted in Figure 4. We use one-period lag of the variables from Table 3. Each triangle △ symbol stands for a splitting node with splitting conditions displayed around the node. The terminal node is represented by a dot ● symbol with terminal value. In a forecasting practice, specific values of the predictor will fall into specific terminal nodes following the tree structure from top to bottom. The specific terminal values are the forecasting results associated to specific values of the predictor.

Figure 4: Forecasting VIX Using RT



Note: Figure 4 represents the tree structure of using RT to forecast VIX with a list of input variables described in Table 3. Each triangle △ symbol stands for a splitting node with splitting conditions displayed around the node. The terminal node is represented by a dot ● symbol with terminal value.

In fact, the RT and MARS methods have strong similarities. The MARS forward stage is the same as the RT tree-growing process, if we replace the piecewise linear basis functions

[11] By no means, the SSR is the only criterion can be used to split the sample. In Section 3.9, we introduce the popular M5 and M5' methods which rely on the reduction of standard deviation to locate the best split.

in MARS by step functions[12] and replace a model term in MARS by the interaction if the term is involved in a multiplication.

In general, an RT outperforms conventional regressions as it yields smaller SSR values. If the data are stationary and ergodic, the RT method demonstrates better forecasting accuracy. Intuitively, for cross-sectional data, the RT method performs better because it removes heterogeneity problems by splitting the sample into clusters with heterogeneous features; for time series data, a good split should coincide with jumps and structure breaks, and therefore, it fits the data to the model better.

We have thus far focused on statistical procedures that produce a single set of results: regression coefficients, measures of fit, residuals, classifications, and others. There is but one regression equation, one set of smoothed values, or one classification tree. Obviously, one won't learn much through just one set of results. In the following sections, we shift to statistical learning that builds on many sets of outputs aggregated to produce results. Such algorithms make a number of passes over the data. On each pass, inputs are linked to outputs just as before. But the ultimate results of interest are the collection of all the results from all passes over the data. These methods crucially depend on a technique called the "bootstrap", which we discuss in length in the next section.

3.4.1 Regression Tree and Local constant model

Following Athey and Imbens (2019), we can interpret the regression tree as a local constant model. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$ be a k-dimensional realization of $\boldsymbol{X}_t$ on which the conditional mean of y_t, i.e. $\mathbb{E}\left(y_t | \boldsymbol{X}_t = \boldsymbol{x}\right) = f(\boldsymbol{x})$ is of interest. Then the data sample can be rewritten as $\left\{(y_1, \boldsymbol{X}_1), (y_2, \boldsymbol{X}_2\right\}, \ldots, (y_T, \boldsymbol{X}_T))$. Denote the neighborhood of $\boldsymbol{x}$ as $N(\boldsymbol{x}) = \{t | \, \|\boldsymbol{X}_i - \boldsymbol{x}\| < h, t = 1, \ldots, T\}$, where h is a given positive real number and $\| \cdot \|$ stands for the Euclidean norm. The term $N(\boldsymbol{x})$ consists of the index of k-dimensional vectors $\boldsymbol{X}_t$ that are in the h-neighborhood of $\boldsymbol{x}$.

[12] Here, we define the step functions as $\mathbb{I}_{x-t>0}$ and $\mathbb{I}_{x-t\leq 0}$, where $\mathbb{I}_{\{\cdot\}}$ equals to 1 if the subscript condition is satisfied and equals to 0 otherwise. The term t is the knot defined in Section 3.1

We let f be continuous. A simple estimator of $\hat{f}$ is the sample mean of y_t's in $N(\boldsymbol{x})$:

$$\hat{f}(\boldsymbol{x}) = \frac{1}{\#N(\boldsymbol{x})} \sum_{t \in N(\boldsymbol{x})} y_t$$

where $\#N(\boldsymbol{x})$ denotes the number of the k-dimensional vectors in $N(\boldsymbol{x})$. Note that the above equation represents a kernel estimation of $f(\cdot)$ with a uniform kernel.

In a regression tree model, the leaf can be regarded as a set of nearest neighbors for the given observation $\boldsymbol{x}$. The estimator of a single regression tree is in fact a matching estimator (with non-conventional algorithm) of selecting the nearest neighbor to $\boldsymbol{x}$. A typical local constant model creates a neighborhood around a target observation based on the Euclidean distance to each point, while tree-based neighborhoods are rectangles. Specially, the regression tree estimator derives a weighting function for a given test point by counting the share of trees where a particular observation is in the same leaf as the test point. The difference between typical kernel weighting functions and regression tree based weighting functions is that the tree weights are adaptive. That is, if a covariate has little effect, it will not be used in splitting leaves, and thus the weighting function will not be very sensitive to distance along that covariate.[13]

3.5 Bootstrap

The term bootstrap, which was introduced to statistics by Efron (1979), is taken from the phrase "to pull oneself up by ones own bootstraps". It is the practice of estimating properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution. Bootstrapping relies heavily on random sampling with replacement.[14] A bootstrap sample is always a subset of the original sample.

[13] Athey and Imbens (2019) argued that the regression tree is particularly effective in settings with a large number of features that are not related to the outcome, that is, settings with sparsity since the splits will generally ignore those covariates, and the performance will remain strong.

[14] The principle of simple random sampling is that every object has the same probability of being chosen. In small populations and often in large ones, such sampling is typically done without replacement, i.e., one deliberately avoids choosing any member of the population more than once. Although simple random sampling

3.5.1 Basic concept

A Russian matryoshka doll is a nest of wooden figures, usually with slightly different features painted on each. Call the outer figure doll 0, the next figure doll 1, and so on. See Figure 5. Suppose we are not allowed to observe doll 0 – it represents the population in a sampling scheme. We wish to estimate the area n_0 of red cheek[15] on her face. Let n_i denote the red cheek area on the face of doll i. Since doll 1 is smaller than doll 0, n_1 is likely to be an underestimate of n_0, but it seems reasonable to suppose that the ratio of n_1 and n_2 should be close to the ratio of n_0 to n_1. That is $n_1/n_2 \approx n_0/n_1$, so that $\hat{n}_0 = n_1^2/n_2$ might be a reasonable estimate of n_0.

Figure 5: A Russian Matryoshka Doll

The key feature of this argument is our hypothesis that the relationship between n_2 and n_1 should closely resemble that between n_1 and the unknown n_0. Under the (fictitious) assumption that the relationships are identical, we equated the two ratios and obtained

can be conducted with replacement instead, this is less common and would normally be described more fully as simple random sampling with replacement, in which the random sampling exhibit independence. Note that sampling done without replacement is no longer independent, but still satisfies exchangeability, hence many results still hold.

[15] This is actually an art representation of freckles.

our estimate $\hat{n}_0$. Of course, we could refine the argument by delving more deeply into the nest of dolls, adding correction terms to $\hat{n}_0$ so as to take account of the relationship between doll i and doll $i+1$ for $i \geq 2$.

The above intuition implies that a population from sample data (sample → population) can be modeled by resampling the sample data and performing inference about a sample data from resampled data (resampled → sample). As the population is unknown, the true error in a sample statistic against its population value is unknown. In bootstrap samples, the population is in fact the sample data, and this is known; hence the quality of inference of the true sample from resampled data (resampled → sample) is measurable.

In any sample data, a specific dependent observation y_i is always tied to a vector of the independent observations $\boldsymbol{X}_i$, as if they are a pair. Although we mainly focus on the dependent variable $\boldsymbol{y}$ in the previous example, the index `IN` that describes bootstrap sample should also be applied to the independent variable $\boldsymbol{X}$. In this fashion, each bootstrap sample consists of a pair $\{\boldsymbol{y}^{(b)}, \boldsymbol{X}^{(b)}\}$ for $b = 1, \ldots, B$. The bootstrapping procedure we described above is also called pairs bootstrap.

3.5.2 Bootstrap in time series

The pairs bootstrap is usually executed for the cross-sectional data. When the data is time series having dependent observations, we need to replace step (i) with specific bootstrap methods for time series based on different assumptions. A straightforward way is to bootstrap the residuals instead of observations. For observations following a stationary Markov chain with finite state-space, Kulperger and Prakasa Rao (1989) initiated the Markov bootstrap method.

If we are not willing to assume a specific structural form for time series (e.g., stationary and weakly dependent), we can use the moving block bootstrap (MBB) method formulated by Künsch (1989). Instead of performing single-data resampling, Künsch (1989) advocated the idea of resampling blocks of observations at a time. By retaining the neighboring observations together within each block, the dependence structure of the random variable at short lag distances is preserved. The block length is predetermined and has impact on the final output to certain degree. We can rely on cross validation methods to select the optimal block length. See Kreiss and Lahiri (2012) for a detailed literature review.

3.6 Bagging tree

We also consider the bootstrap aggregation (BAG) technique developed in Breiman (1996). Unlike the RT method, the BAG method involves a training process where the level of training is predetermined. The BAG algorithm is summarized as below:

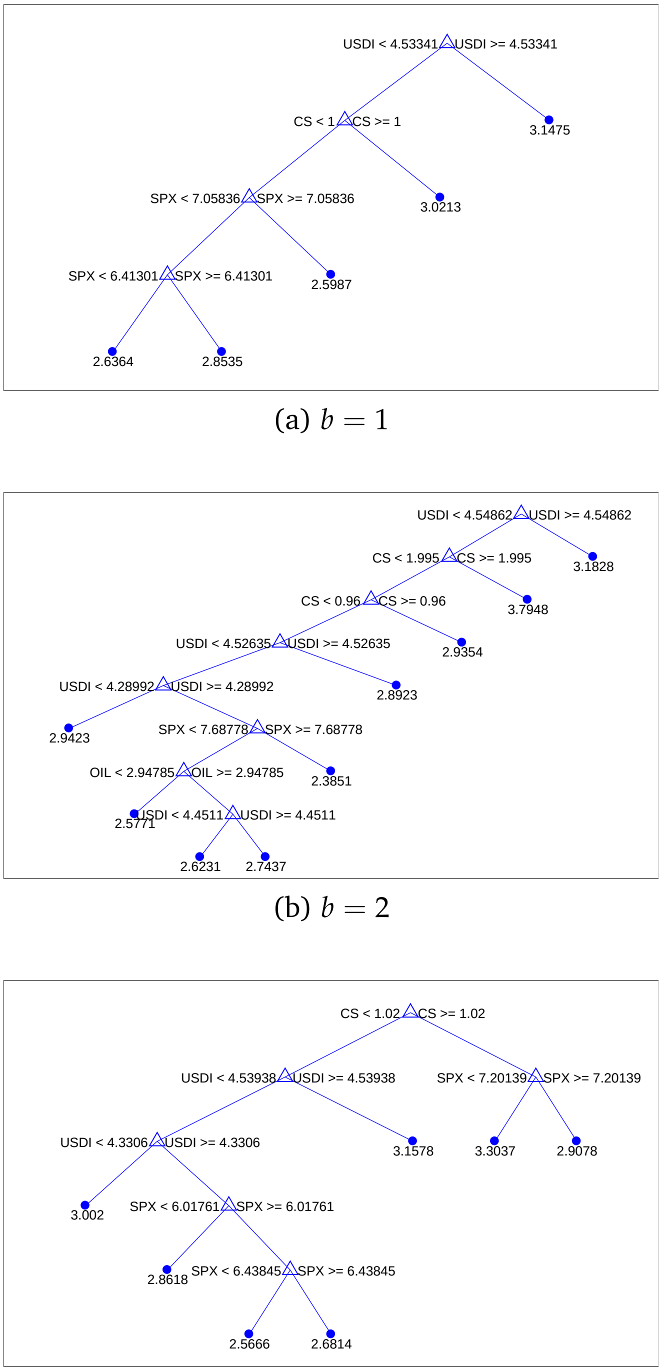
Algorithm 11 Bootstrap Aggregation

```
1. Take a random sample with replacement from the data.
2. Construct a regression tree.
3. Use the regression tree to make forecast, $\hat{f}$.
4. Repeat steps (i) to (iii), $b = 1, \ldots, B$ times and obtain $\hat{f}^b$ for each $b$.
5. Take a simple average of the $B$ forecasts $\hat{f}_{\text{BAG}} = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b$ and consider the averaged value $\hat{f}_{\text{BAG}}$ as the final forecast.
```

For most of the part, the more bootstrap samples in the training process, the better the forecast accuracy. However, more bootstrap samples means longer computational time. A balance needs to be found between accuracy and time costs and constraints.

As an illustration, we continue the VIX forecasting example and show 3 bagging tree structures. Given the time series nature of the data, we use MBB with block size 150 to resample the data. Results are presented in Figure 6(a) – 6(c) respectively. Each bagging tree structure is different from the original regression tree depicted in Figure 4. Of course, Figure 6 is merely a demonstration of different tree structures in a bagging process. In a forecasting practice, we shall make forecasts using more bagging trees for the same predictor value and take the simple average of the forecasts as the final output.

Figure 6: Bagging Trees on Forecasting VIX with Multiple Input Variables



(a) $b = 1$

(b) $b = 2$

(c) $b = 3$

Note: Figure 6(a) – 6(c) depict three typical bagging tree structures using MBB as the resampling technique. Each triangle △ symbol stands for a splitting node with splitting conditions displayed around the node. The terminal node is represented by a dot ● symbol with terminal value.

3.7 Random forest

Random forest (RF) by Breiman (2001) is a modification of bagging that builds a large collection of de-correlated trees, and then averages them. Similar to BAG, RF also constructs B new trees with (conventional or MBB) bootstrap samples from the original data set. But for RF, as each tree is constructed, we take a random sample (without replacement) of q predictors out of the total K $(q < K)$ predictors before each node is split. Such process is repeated for each node. Note that if $q = K$, RF is equivalent to BAG. Eventually, we end up with B trees like BAG and the final RF forecast is calculated as the simple average of forecasts from each tree.

3.8 Boosting tree

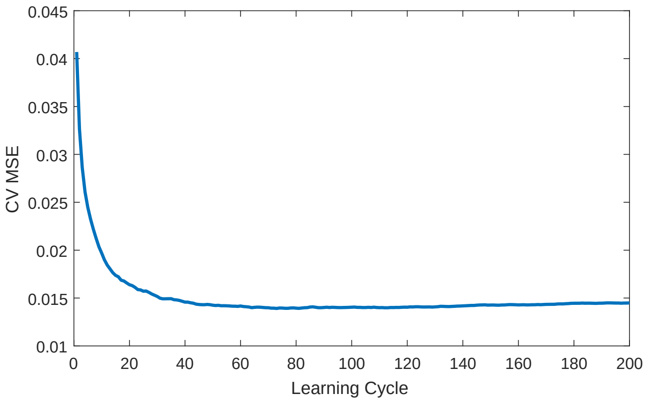
The RT method can respond to highly local feature of the data, since it capitalizes on very flexible fitting procedures. An alternative method to accommodate highly local features of the data is to give the observations responsible for the local variation more weight in the fitting process. If a fitting function fits those observations poorly, we reapply that function with extra weight given to the observations poorly fitted. For a large number of trials, we assign relatively more weights to the poorly fitted observations, hence, combine the outputs of many weak fitting functions to produce a powerful committee, as described in Hastie et al. (2009, Chapter 10).

The procedure we just described is called boosting. Although they assemble similarities, the boosting method is fundamentally different from the RF method. Boosting works with the full training sample and all of the predictors. Within each iteration, the poorly fitted observations are given more relative weight, which eventually forces the (poor) fitting functions to evolve in boosting. We usually denote the number of iterations as *learning cycle* of the boosting process. Moreover, the final output values are a weighted average over a large set of earlier fitting results instead of simple average as in the RF method.

Many of the boosting methods are designed for classification issues, for example, the most popular boosting algorithm `AdaBoost.M1` by Freund and Schapire (1997). For numerical analysis, we favor the simpler least squares boosting (LSB) that fits RT ensembles. In line with Hastie et al. (2009, Chapter 8), at every step, the LSB method applies a new learning tree to the difference between the observed response and the aggregated prediction of all trees grown previously.

We revisit the VIX forecasting exercises, this time, using the LSB method with input variables described in Table 3. We compute the cumulative MSE from a five-fold cross-validation for different numbers of learning cycle (iteration). The following Figure 7 describes the relationship between CV MSE and learning cycle. As we can see, the CV MSE shrinks as the learning cycle increases and eventually becomes steady once the learning cycle exceeds 40.

Figure 7: Learning RT Boosting



Note: We apply the LSB method to the VIX forecasting exercises and depict the relationship between 5-fold CV MSE and learning cycle. The CV MSE shrinks becomes steady once the learning cycle exceeds 40.

3.9 M5' algorithm

All decision tree algorithms discussed above base their forecasts on a set of piecewise local constant model. In fact, numerous researchers in machine learning have developed algorithms[16] that estimate regression models in the leaf nodes to not just aid in prediction, but also simplify the tree model structure. That is, these researchers often suggest that the gains in prediction from using a piecewise linear model could allow one to grow shorter trees that are more parsimonious. Not surprisingly, ex ante from an econometrics perspective the success of these linear tree algorithms clearly depend on both the source and amount of heterogeneity in the underlying data.

Perhaps the best known of the linear regression tree algorithms is the M5 algorithm of Quinlan (1992) that was further clarified in the M5' algorithm of Wang and Witten (1997). The M5' algorithm builds subgroups using the same algorithm as RT but a multiple regression models is estimated in the terminal node. The model in each leaf only contains the independent variables encountered in split rules in the leaf node's sub-tree and are simplified to reduce a multiplicative factor to inflate estimated error.

Moreover, the M5' model tree uses a different criteria to construct splits in the tree. Splits are based on minimizing the intra-subset variation in the output values down each branch. In each node, the standard deviation of the output values for the examples reaching a node is taken as a measure of the error of this node and calculating the expected reduction in error as a result of testing each attribute and all possible split values. The attribute that maximizes the expected error reduction is chosen. The standard deviation reduction (SDR) is calculated by

$$SDR = sd(T) - \sum_i sd(T_i) \times |T_i|/|T|,$$

where T is the set of examples that reach the node and T_is are the sets that result from splitting the node according to the chosen attribute (in case of multiple split). As usual, the splitting process will terminate if the output values of all the instances that reach the node vary only slightly or only a few instances remain.

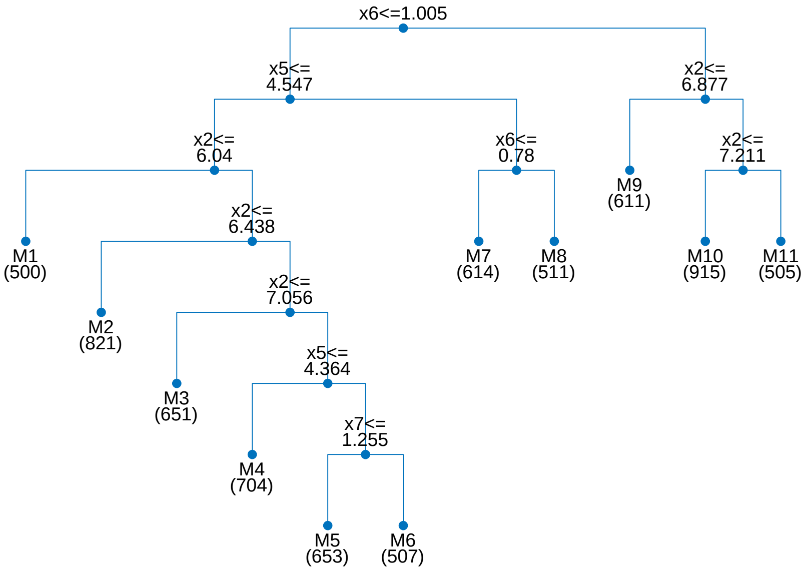
Similar to M5 once the tree has been grown, M5' estimates a multivariate linear model

[16] See Quinlan (1992), Chaudhuri et al. (1995), Kim and Loh (2003), Vens and Blockeel (2006), among others

in each tree leaf that only includes variables that were used in the subtree of this node. Thus, the M5' model tree is also analogous to using piecewise linear functions in each leaf.

We apply the M5' algorithm to the VIX forecasting exercises and depict the model tree in Figure 8. We use one-period lag of the variables from Table 3. Each dot ● symbol stands for a splitting node with splitting conditions going leftward displayed above the node.[17] The terminal node is represented by "M#" with number of observations contained in the leaf within parentheses. Unlike CART, we do not model the subsamples within a leaf by its sample mean but use a linear regression model instead. The tree is pruned with restriction such that the number of obs. within a leaf should be no less than 500. Inputs x2 to x8 correspond to the variables listed in Table 3, respectively.

Figure 8: M5' Model Tree Plot



Note: The tree is pruned with restriction such that the number of obs. within a leaf should be no less than 500. Inputs x2 to x8 correspond to the variables listed in Table 3, respectively.

[17] We use the MATLAB package written by Gints Jekabsons (http://www.cs.rtu.lv/jekabsons/). The general display of the plots is a bit different from MATLAB's built-in figure.

3.10 Neural network

The neural network (NN) become a hype word recently, even more so since the flourishing of big data analytics. The NN model can be categorized as nonlinear statistical models. The basic motivation of NN can date back to McCulloch and Pitts (1943). In this section, we first describe the most widely used (vanilla) neural net[18] discussed in Rumelhart et al. (1986). Then, we move on to introduce a more complicated nonlinear autoregressive network with exogenous inputs (NARX) that is commonly used in time-series modeling.

An (vanilla) NN is a two-stage regression or classification model. Given the nature of our data set, we concentrate on applying the NN method to the former. Let y_t and $\boldsymbol{X}_t$ (with constant term) be the output and input measurements respectively. We create derived features Z_{mt} from linear combinations of $\boldsymbol{X}_t$ for $m = 1, \dots, M$, where the Z_{mt} terms are called hidden units[19] and M is a predetermined number. The hidden units Z_{mt} is connected with inputs $\boldsymbol{X}_t$ through a so-called activation function $g(\cdot)$ such that

$$Z_{mt} = g(\boldsymbol{X}_t \boldsymbol{\alpha}_m),$$

where $\boldsymbol{\alpha}_m = [\alpha_{m1}, \dots, \alpha_{mK}]^\top$ is a vector of coefficients associated with $\boldsymbol{X}_t$ for each hidden unit. The activation function is usually chosen to be the sigmoid $g(v) = 1/(1 + e^{-v})$. For convenience, we let $\boldsymbol{Z}_t \equiv [Z_{1t}, \dots, Z_{Mt}]$ and the $\boldsymbol{Z}_t$ group is usually referred as a hidden layer. Note that there can be more than one hidden layer. Then, the output y_t is modeled as a function of linear combination of $\boldsymbol{Z}_t$ such that

$$y_t = f(\boldsymbol{X}_t) + \epsilon_t = \boldsymbol{Z}_t \boldsymbol{\beta} + \epsilon_t, \tag{160}$$

where $\boldsymbol{\beta} = [\beta_1, \dots, \beta_M]^\top$ is the associated coefficient vector for $\boldsymbol{Z}_t$.

[18] The simple neural net is sometimes called the single hidden layer back-propagation network, or single layer perception.

[19] In the NN literature, the Z_{mt} terms are usually named as neurons, as each connection (synapse) between Z_{mt} can transmit a signal to another layer of Z_{mt} like neurons. Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs.

In the NN literature, coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}_m$ are often called weights. To estimate the weights, we use SSR as our measure of fit

$$\text{SSR}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{t=1}^{n} \text{SSR}_t(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{t=1}^{n} \big(y_t - f(\boldsymbol{X}_t)\big)^2,$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M]$. Unlike conventional regression estimation, we usually don't want the global minimizer of $\text{SSR}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ as the estimated weights in order to avoid overfitting. Instead, we impose some regularization (early stopping rule or penalty for complexity) and minimize $\text{SSR}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ by gradient descent, namely, the back-propagation algorithm.

Due to the compositional form of the model, the gradient can be easily derived using the chain rule for differentiation through a forward and backward sweep over the network. We define the following derivatives with respect to β_m and α_{mk} for $m = 1, \ldots, M$ and $k = 1, \ldots, K$

$$\begin{aligned} \frac{\partial \text{SSR}_t}{\partial \beta_m} &= -2\big(y_t - f(\boldsymbol{X}_t)\big)\boldsymbol{Z}_t \equiv \delta_t \boldsymbol{Z}_t, \\ \frac{\partial \text{SSR}_t}{\partial \alpha_{mk}} &= -2\big(y_t - f(\boldsymbol{X}_t)\big)\beta_m g'(\boldsymbol{X}_t \boldsymbol{\alpha}_m) x_{tk} \equiv s_{mt} x_{tk}, \end{aligned} \tag{161}$$

where δ_t and s_{mt} are the defined coefficients associated to $\boldsymbol{Z}_t$ and x_{tk} in (128). A gradient descent update at the $(r+1)^{th}$ iteration has the form:

$$\begin{aligned} \beta_m^{(r+1)} &= \beta_m^{(r)} - \gamma_r \sum_{t=1}^{n} \frac{\partial \text{SSR}_t}{\partial \beta_m^{(r)}}, \\ \alpha_{mk}^{(r+1)} &= \alpha_{mk}^{(r)} - \gamma_r \sum_{t=1}^{n} \frac{\partial \text{SSR}_t}{\partial \alpha_{mk}^{(r)}}, \end{aligned} \tag{162}$$

where γ_r is a pre-determined learning rate. From (128), we can derive

$$s_{mi} = \beta_m g^{\top}(\boldsymbol{X}_t \boldsymbol{\alpha}_m)\delta_t, \tag{163}$$

which is known as the back-propagation equations. Using the system of equations defined in (163), updates in (162) can be implemented with the following two-pass algorithm:

1. Forward pass: given current estimates of $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\alpha}}_m$ for $m = 1, \ldots, M$, we compute the predicted value $\hat{f}(\boldsymbol{X}_t)$.

2. Backward pass: use $\hat{f}(\boldsymbol{X}_t)$ to compute $\hat{\delta}_t$ first, then obtain $\hat{s}_{mt}$ by formula (163). Both sets of coefficients are then used to compute the derivatives defined in (128), which finally leads to the gradients for updates of next round $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in (162).

Starting with some initial values, the above algorithm is carried out multiple times until convergence or stop early to avoid overfitting.

The NARX network is good at time series prediction and can be considered as an extension of the simple NN framework we discussed above. In the NARX network, the output y_t is modeled by the following NARX function

$$\begin{aligned} y_t &= f(\boldsymbol{X}_t) + \epsilon_t, \\ \boldsymbol{X}_t &= [y_{t-1}, \ldots, y_{t-L}, \boldsymbol{X}_t^*], \end{aligned}$$

where function $f(\cdot)$ is a unknown nonlinear function, L stands for the maximum number of lags and $\boldsymbol{X}_t^*$ contains all the exogenous variables at time t.

3.11 Support vector machine for regression

In machine learning, support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The theory behind SVM is due to Vapnik and is described in Vapnik (1996). The classic SVM was designed for classification and a version of SVM for regression, later known as support vector regression (SVR), was proposed in by Drucker et al. (1996). The goal of SVR is to find a function $f(\boldsymbol{X}_t)$ that deviates from y_t by a value no greater than a predetermined ϵ for each observations $\boldsymbol{X}_t$, and at the same time is as flat as possible.

In this paper, we first consider the SVR for the linear regression model (SVR$_L$). Following Hastie et al. (2009, Chapter 12),

$$y_t = f(\boldsymbol{X}_t) + \epsilon_t = \boldsymbol{X}_t\boldsymbol{\beta} + \epsilon_t = \beta_0 + \tilde{\boldsymbol{X}}_t\boldsymbol{\beta}_1 + \epsilon_t,$$

where $\boldsymbol{X}_t = [1, \tilde{\boldsymbol{X}}_t]$ and $\boldsymbol{\beta} = [\beta_0, \boldsymbol{\beta}_1^\top]^\top$. We estimate $\boldsymbol{\beta}$ through the minimization of

$$H(\boldsymbol{\beta}) = \sum_{t=1}^{n} V_\varepsilon\big(y_t - f(\boldsymbol{X}_t)\big) + \frac{\lambda}{2}\|\boldsymbol{\beta}_1\|^2, \tag{164}$$

where the loss function

$$V_\varepsilon(r) = \begin{cases} 0 & \text{if } |r| < \varepsilon \\ |r| - \varepsilon & \text{otherwise} \end{cases}$$

is called an ε-insensitive error measure that ignores errors of size less than ε. As a part of the loss function V_ε, the parameter ϵ is usually predetermined. On the other hand, λ is a more traditional regularization parameter, that can be estimated by cross-validation.

Let $\hat{\boldsymbol{\beta}} = \left[\hat{\beta}_0, \hat{\boldsymbol{\beta}}_1^\top\right]^\top$ be the minimizers of function (164). The solution functions follow

$$\begin{aligned} \hat{\boldsymbol{\beta}}_1 &= \sum_{t=1}^{n}(\hat{\alpha}_t^* - \hat{\alpha}_t)\tilde{\boldsymbol{X}}_t^\top, \\ \hat{f}(\boldsymbol{X}) &= \sum_{t=1}^{n}(\hat{\alpha}_t^* - \hat{\alpha}_t)\boldsymbol{X}\boldsymbol{X}_t^\top + \hat{\beta}_0\boldsymbol{\iota}_n, \end{aligned}$$

where $\boldsymbol{\iota}_n$ is an $n \times 1$ vector of ones and the parameters $\hat{\alpha}_t$ and $\hat{\alpha}_t^*$ are the nonnegative multiplier of the following Lagrangian equation

$$\min_{\hat{\alpha}_t, \hat{\alpha}_t^*} \epsilon \sum_{t=1}^{n}(\hat{\alpha}_t^* + \hat{\alpha}_t) - \sum_{t=1}^{n} y_i(\hat{\alpha}_t^* - \hat{\alpha}_t) + \frac{1}{2}\sum_{t=1}^{n}\sum_{t^\top=1}^{n}(\hat{\alpha}_t^* - \hat{\alpha}_t)(\hat{\alpha}_{t^\top}^* - \hat{\alpha}_{t^\top})\boldsymbol{X}_t\boldsymbol{X}_{t^\top}^\top$$

subject to the constraints

$$0 \le \hat{\alpha}_t^*, \hat{\alpha}_t \le 1/\lambda, \quad \sum_{t=1}^{n}(\hat{\alpha}_t^* - \hat{\alpha}_t) = 0, \quad \hat{\alpha}_t\hat{\alpha}_t^* = 0$$

for all $t = 1, \ldots, n$. We usually called the non-zero values of $\hat{\alpha}_t^* - \hat{\alpha}_t$ for $t = 1, \ldots, n$ the support vector.

We now extend the above SVR framework for linear regression to nonlinear regression (SVR$_N$). We approximate the nonlinear regression function $f(\boldsymbol{X}_t)$ in terms of a set of basis function $\{h_m(\tilde{\boldsymbol{X}}_t)\}$ for $m = 1, \ldots, M$:

$$y_t = f(\boldsymbol{X}_t) + \epsilon_t = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(\tilde{\boldsymbol{X}}_t) + \epsilon_t,$$

and we estimate the coefficients $\boldsymbol{\beta} = \left[\beta_0, \beta_1, \ldots, \beta_M\right]^\top$ through the minimization of

$$H(\boldsymbol{\beta}) = \sum_{t=1}^{n} V_\varepsilon\big(y_t - f(\boldsymbol{X}_t)\big) + \frac{\lambda}{2} \sum_{m=1}^{M} \beta_m^2. \tag{165}$$

The solution of (165) has the form

$$\hat{f}(\boldsymbol{X}) = \sum_{t=1}^{n} (\hat{\alpha}_t^* - \hat{\alpha}_t) K(\boldsymbol{X}, \boldsymbol{X}_t) + \hat{\beta}_0 \boldsymbol{\iota}_n,$$

with $\hat{\alpha}_t^*$ and $\hat{\alpha}_t$ being the nonnegative multiplier of the following Lagrangian equation

$$\min_{\hat{\alpha}_t, \hat{\alpha}_t^*} \ \varepsilon \sum_{t=1}^{n} (\hat{\alpha}_t^* + \hat{\alpha}_t) - \sum_{t=1}^{n} y_i (\hat{\alpha}_t^* - \hat{\alpha}_t) + \frac{1}{2} \sum_{t=1}^{n} \sum_{t^\top=1}^{n} (\hat{\alpha}_t^* - \hat{\alpha}_t)(\hat{\alpha}_{t^\top}^* - \hat{\alpha}_{t^\top}) K(\boldsymbol{X}_t, \boldsymbol{X}_{t^\top}),$$

which is similar to the SVR$_L$ case. In the SVR$_N$ case, a kernel function

$$K(\boldsymbol{X}_t, \boldsymbol{X}_{t^\top}) = \sum_{m=1}^{M} h_m(\boldsymbol{X}_t) h_m(\boldsymbol{X}_{t^\top}),$$

is used to replace the inner product of the predictors $\boldsymbol{X}_t \boldsymbol{X}_{t^\top}^\top$ as in the SVR$_L$ case. In our paper, we consider the following kernel functions

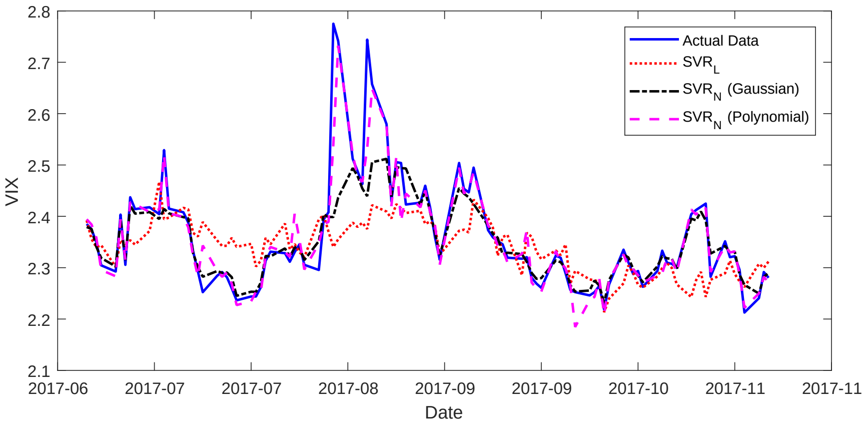
$$K(\boldsymbol{X}_t, \boldsymbol{X}_{t^\top}) = \exp\left(-\|\boldsymbol{X}_t - \boldsymbol{X}_{t^\top}^\top\|^2\right), \tag{166}$$

$$K(\boldsymbol{X}_t, \boldsymbol{X}_{t^\top}) = \left(1 + \boldsymbol{X}_t \boldsymbol{X}_{t^\top}^\top\right)^p \quad \text{with } p \in \{2, 3, \ldots\}. \tag{167}$$

Note that if we set $K(\boldsymbol{X}_t, \boldsymbol{X}_{t^\top}) = \boldsymbol{X}_t \boldsymbol{X}_{t^\top}^\top$, the SVR$_N$ becomes identical to SVR$_L$.

A representation of linear SVR model compared to nonlinear SVR models is depicted in Figure 9. We consider a one-step-ahead VIX forecasting using SVR with various kernels. To keep the figure uncluttered, we only show the results in recent periods from 2017-06-20 to 2017-11-08. The solid line represents the actual exchange rate data. The dotted line, dash-dotted line, and dashed line represent the forecasted results by SVR$_L$, SVR$_N$ by Gaussian kernel, and SVR$_N$ by polynomial kernel with $p = 3$, respectively.

Figure 9: Forecast VIX with the SVR Method



Note: We consider a one-step-ahead forecasting of VIX using SVR$_L$, SVR$_N$ by Gaussian kernel, and SVR$_N$ by polynomial kernel, with results represented by dotted line, dash-dotted line, and dashed line, respectively. To keep the figure uncluttered, we only show the results in recent periods from 2017-06-20 to 2017-11-08.

Both SVR$_L$ and SVR$_N$ by Gaussian kernel have similar performance. They are able capture the direction change of the VIX but are less volatile than the actual data. The SVR$_N$ by polynomial kernel outperforms the others in general. In fact, due to the high order components in the kernel, SVR$_N$ by polynomial kernel is able to capture more volatile movement in the data, for example, the two large spikes around 2017-08.

4 Hybrid methods

However, it is possible to combine the strengths of both methods for cross-fertilization. Most machine learning techniques neglect parameter heterogeneity as they typically rely on local constant models that assume homogeneity in outcomes within individual terminal leaves. This limitation can impact their predictive ability. Presence of heterogeneity can change how the data should be partitioned thereby influencing the forecasting results. On the other hand, conventional econometric methods have provided many effective techniques to deal with heterogeneity. This sets a motivation of the need of hybrid methods.

In this section, we review new strategies for predictive analytics that are contrasted with existing tools from both the econometrics and machine learning literature to first give guidance on how to improve forecast accuracy in applications. These so-called hybrid strategies first use recursive partitioning methods to develop subgroups and then undertake model averaging within these terminal groups to generate forecasts. By allowing for model uncertainty in the subgroups (split-sample, tree leaves, etc.), richer forms of heterogeneity in the relationships between independent variables and outcomes within each subgroup is allowed. We also conduct a simple Monte Carlo simulation to illustrate the benefits of using hybrid tree method.

4.1 Split-sample methods and SPLT$_{PMA}$ methods

Hirano and Wright (2017) proposed a split-sample (SPLT) method to mitigate uncertainty about the choice of predictor variables. They investigate the distributional properties of SPLT in a local asymptotic framework. The core of SPLT is more in the econometric tradition, which consists of splitting the training sample set into two parts, one for model selection via AIC and the other for model estimation. Moreover, the authors show that, adding a bagging step to the plain SPLT substantially improves its prediction performance. The bagging augmented SPLT method can be viewed as a hybrid of econometric and machine learning methods, and is implemented in our simulation exercise.

Liu and Xie (2018) further extended SPLT by replacing the AIC model selection method by the prediction model average (PMA) method developed by Xie (2015), while keeping the bagging procedure. Liu and Xie (2018) denoted this hybrid method by SPLT_{PMA}. In SPLT_{PMA}, after an initial sample split, PMA is applied to the first subsample to obtain a weight structure over all candidate models, and then use the weights to calculate a weighted average model as the prediction model, where each candidate model is estimated on the second subsample. The SPLT_{PMA} algorithm can be summarized as follows:

Algorithm 12 Split-sample Model Averaging Method

1. Draw a random sample *with replacement* from the original training set.
2. Split the sample into two parts by minimizing the total SSR.[20]
3. Apply the PMA method to the first subsample and obtain a weight structure on all candidate model.
4. Estimate each candidate model on the second subsample, and make prediction on the evaluation set.
5. Use the weights in (iii) to calculate the model average forecast using candidate forecasts in (iv).
6. Repeat steps (i) to (v) by B times.
7. The final forecast is the simple average of B model average forecasts in (v).

4.2 Model average tree

To construct forecasts with either (i) regression trees, (ii) bagging, or (iii) random forests, one calculates the predicted value for each leaf l as the value $\bar{y}_{i\in l}$ is actually the fitted value of the following regression model

$$y_i = a + u_i, \quad i \in l, \tag{168}$$

where u_i is the error term and a stands for a constant term with least square estimate $\hat{a} = \bar{y}_{i\in l}$. In other words, after partitioning the dataset into various subgroups, no heterogeneity is assumed within subgroups. From the perspective of the econometrician, this rules out heterogeneity within recursively partitioned subgroups and may appear unsatisfying.

Lehrer and Xie (2018) suggested that for each tree leaf we can construct a sequence of $m = 1, \ldots, M$ linear candidate models, in which regressors of each model m is a subset of the regressors belonging to that tree leaf. The regressors $\boldsymbol{X}^m_{i\in l}$ for each candidate model within each tree leaf is constructed such that the number of regressors $k^m_l \ll n_l$ for all m. Using these candidate models, we perform model averaging estimation and obtain the averaged coefficient

$$\underset{(K\times 1)}{\hat{\boldsymbol{\beta}}_l(\boldsymbol{w})} = \sum_{m=1}^{M} w^m \underset{(K\times 1)}{\tilde{\boldsymbol{\beta}}^m_l}, \tag{169}$$

which is a weighted averaged of the “stretched” estimated coefficient $\tilde{\boldsymbol{\beta}}^m_l$ for each candidate model m. Note that the $K \times 1$ sparse coefficient $\tilde{\boldsymbol{\beta}}^m_l$ is constructed from the $k^m_l \times 1$ OLS coefficient $\hat{\boldsymbol{\beta}}^m_l$ by filling the extra $K - k^m_l$ elements with 0s.

Once the averaged coefficients $\hat{\boldsymbol{\beta}}_l(\boldsymbol{w})$ are constructed for each leaf in a regress tree, we compute the forecasts for all predicting observations:

$$\hat{y}_{t\in l} = \boldsymbol{X}^p_{t\in l}\hat{\boldsymbol{\beta}}_l(\boldsymbol{w}). \tag{170}$$

Note that although the predictors classified in each tree leaf share the common averaged coefficients $\hat{\boldsymbol{\beta}}_l(\boldsymbol{w})$, they generate different forecasts $\hat{y}_{t\in l}$ as the predictors $\boldsymbol{X}^p_{t\in l}$ are also included in the estimation process.

We denote the above method as model averaging regression tree (MART), in which we replace the original leaves (averages of y) of a regression tree with model averaging estimates without altering the original classification process. We apply the same process to each of the B regression trees in bagging. We obtain forecasts from each tree and the equal weight averages of these forecasts is the final bagging forecast value. We denote this method as model averaging bagging (MAB). Applying MART to random forest is essentially the same as MAB with one difference. For random forest, the split of the node is done by the classification of a random sample (without replacement) of k predictors out of the total K predictors. Therefore, when calculating the averaged coefficients $\hat{\boldsymbol{\beta}}_l(\boldsymbol{w})$ for each leaf l in a tree of the random forest, the candidate model set is not constructed from the K regressors as in bagging, but from the k regressors used to split the node contains leaf l. We denote this method as model averaging random forest (MARF).

In a forecast exercise, the predicting observations $\boldsymbol{X}_t^p$ with $t = 1, 2, \ldots, T$ are dropped down the regression tree. For each $\boldsymbol{X}_t^p$, after several steps of classification, we end up with one particular tree leaf l. We denote the predicting observations that are classified in tree leaf l as $\boldsymbol{X}_{t \in l}^p$. If the full sample contains n observations, the tree leaf l contains a subset $n_l < n$ of the full sample of y, denoted as y_i with $i \in l$. Also, the sum of all n_l for each tree leaf equals n. The mean of $y_{i \in l}$ is calculated, denoted as $\bar{y}_{i \in l}$. The value $\bar{y}_{i \in l}$ is the forecast estimate of $\boldsymbol{X}_{t \in l}^p$. It is quite possible that different predicting observations $\boldsymbol{X}_t^p$ and $\boldsymbol{X}_s^p$ with $t \neq s$ will end up with the same tree leaf, therefore, generates identical forecasts.

4.3 A simple illustration of the MART hybrid method

In this section, we replicate the simple illustration of the MART hybrid method in Lehrer et al. (2018). To illustrate the benefits of allowing for heterogeneity due to model uncertainty in each tree leaf in the forest via this two-step hybrid procedure, we simulate data drawn from a non-linear process. Panels (a) and (b) of Figure 10 respectively present the scatterplot and surface plot of training data generated by

$$Y = \sin(X_1) + \cos(X_2) + u,$$

where $X_1 \in [1, 10]$, $X_2 \in [1, 10]$, and u is a Gaussian noise with mean 0 and variance 0.01. Forecasts of Y calculated from RT and MART with the training data are presented in Panels (c) and (d) of Figure 10. Since RT forecasts assume homogeneity within leaves, the surface plot in Panel (c) appears similar to a step-function. In contrast, by allowing for heterogeneity in the forecasts within each leaf, the surface plot from MART in Panel (d) more closely mimics the variation in the joint distribution in the underlying data.

To demonstrate the gains from using MART in place of RT when forecasting Y, we plot the forecast errors from RT and MART against both X_1 and X_2 in panels (e) and (f) of Figure 10. The visualizations in these panels clearly show that the absolute biases from MART are less than half of the biases obtained from RT. These panels illustrate not only the significant benefits from adopting the proposed hybrid approach, but clarify that the gains are achieved by allowing for richer relationships in each tree leaf.

5 Empirical Illustration I: VIX Forecasting

5.1 Data description

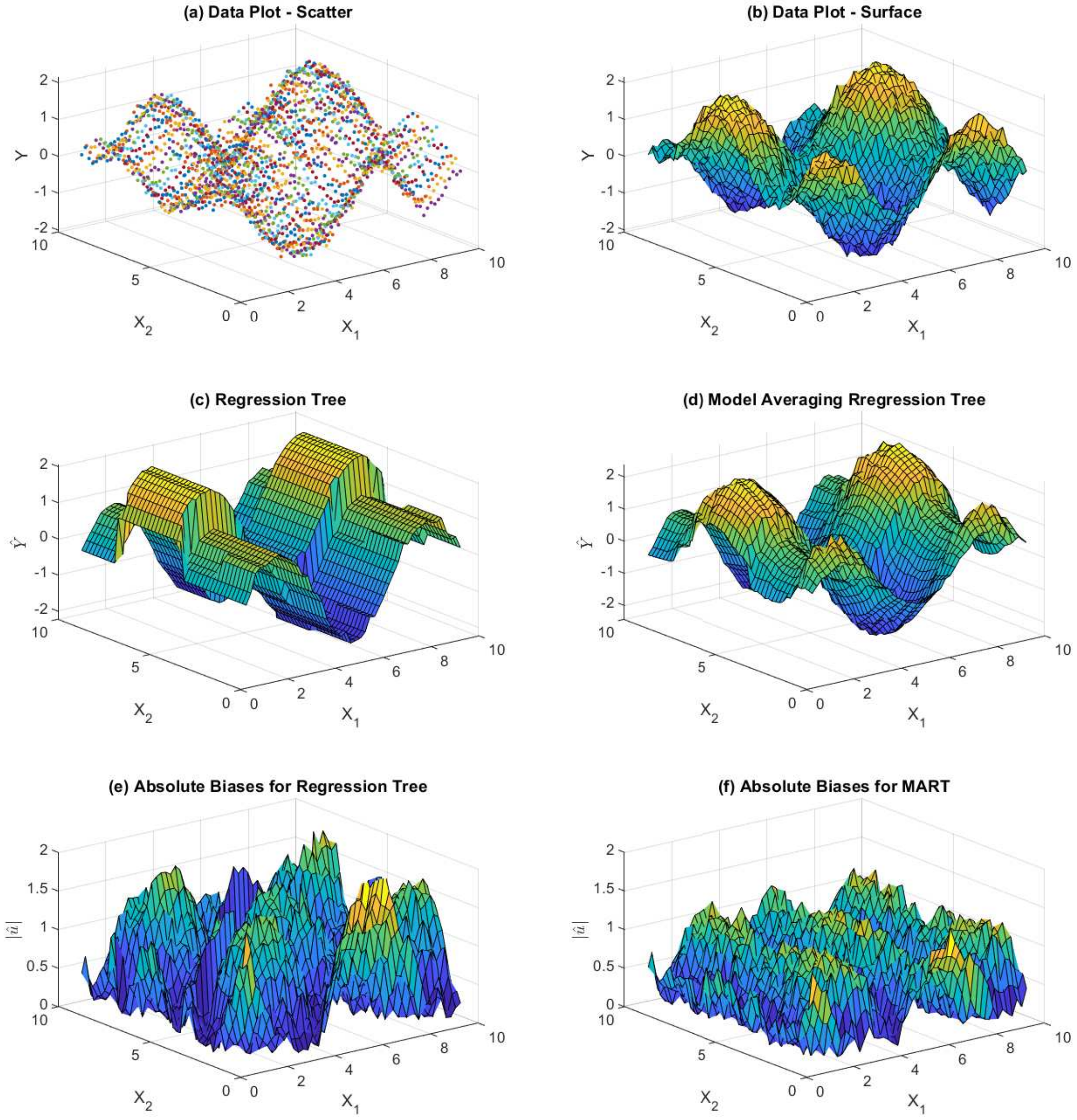
In this study, we use the Chicago Board Options Exchange (CBOE) Volatility Index (VIX) forecasting exercise as an example to demonstrate the pros and cons of each method we introduce and investigate in this proposal. The CBOE VIX is colloquially referred to as the "fear index" or the "fear gauge". We choose to study the VIX not only on the widespread consensus that the VIX is a barometer of the overall market sentiment as to what concerns investors' risk appetite, but also on the fact that there are many trading strategies that rely on the VIX index for hedging and speculative purposes.

The VIX index is a weighted blend of prices for a range of options on the S&P 500 index.[21] The formula that calculates the VIX index is

$$\mathrm{VIX} = 100 \times \sqrt{\frac{2}{T}\sum_{i}\frac{\Delta K_i}{K_i^2}e^{rT}Q(K_i) - \frac{1}{T}\left(\frac{F}{K_0}-1\right)^2}, \tag{171}$$

[21] The VIX is quoted in percentage points and represents the expected range of movement in the S&P 500 index over the next year, at a 68% confidence level (i.e. one standard deviation of the normal probability curve).

Figure 10: Simple Monte Carlo Simulation Results



(a) Data Plot - Scatter

(b) Data Plot - Surface

(c) Regression Tree

(d) Model Averaging Rregression Tree

(e) Absolute Biases for Regression Tree

(f) Absolute Biases for MART

Figure 11: The Daily Log Index of VIX from January 2, 1990 to November 20, 2017

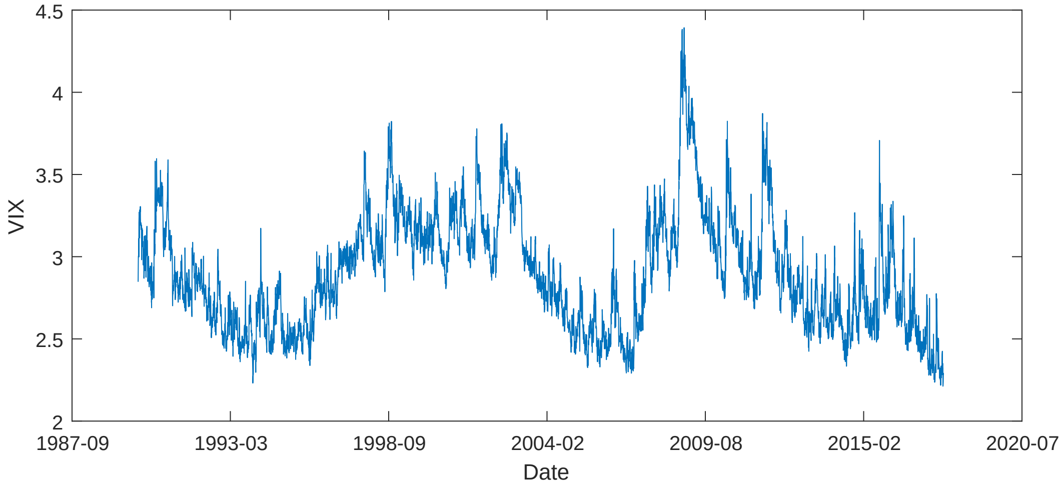


Table 3: List of Input Variables for Forecasting VIX

Variable	General Description
SPX*	the logarithm of the S&P500 index
SPV	the logarithm of the volume of the S&P500 index
OIL	the logarithm of one-month crude oil futures contract
USD*	the logarithm of the foreign exchange value of the U-S dollar index
CS*	the credit spread, which is the excess yield of the Moody's seasoned Baa corporate bond over the Moody's seasoned Aaa corporate bond
TS*	the term spread, which is the difference between the 10-Year and 3-month treasury constant maturity rates
FFD	the difference between the effective and target Federal Funds rates

$*$ These variables are not stationary and their first-order differences are used in the exercises.

where T is time to expiration, F is the forward index level derived from the index options prices, K_i is the strike price of the ith out-of-the-money option, $\Delta K_i = (K_{i+1} - K_{i-1})/2$, K_0 is the first strike below the forward index level, r is the risk-free interest rate to expiration, and $Q(K_1)$ is the mid-quote for the option with strike of K_i.

We collect VIX from 1990-01-02 to 2017-11-20. Figure 11 illustrates the time evolution of the log of VIX index in the full sample period. Following the literature, we also incorporate standard predictors for VIX forecasting, including SPX, SPV, OIL, USD, CS, TS, and FFD. These predictors are listed and described in details in Table 3. Note that variables with * are nonstationary and their first-order differences are used in the exercises.

Table 4: Summary of Statistics

Statistics	VIX	SPX*	SPV	OIL	USD*	CS*	TS*	FFD
Mean	2.9004	0.0003	20.9501	3.6396	0.0000	0.0000	0.0001	0.0109
Median	2.8651	0.0005	21.1178	3.5306	0.0000	0.0000	0.0000	0.0000
Maximum	4.3927	0.1096	23.1618	4.9821	0.0248	0.4100	0.7400	3.6400
Minimum	2.2127	-0.0947	17.8621	2.3721	-0.0411	-0.1500	-0.5600	-1.8100
Std. Dev.	0.3490	0.0111	1.1402	0.6471	0.0044	0.0201	0.0662	0.1784
Skewness	0.6461	-0.2559	-0.4232	0.1640	-0.2170	2.5690	0.3454	2.8856
Kurtosis	3.3367	11.9184	1.9070	1.6666	6.3283	52.5896	13.1333	57.2906
Jarque-Bera	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010
ADF Test	0.0010	0.0010	0.0010	0.0158	0.0010	0.0010	0.0010	0.0010

Note that variables with * are nonstationary and their first-order differences are used in the exercises.

We describe summary statistics of the (log) VIX and its predictors in Table 4. Variables are listed in the first row of Table 4. We document the results of the sample mean, median, minimum, maximum, standard deviation, skewness, and kurtosis for all the variables over the full sample periods. Table 4 also reports the p-values[22] of the Jarque-Bera test for normality and those of the augmented Dickey-Fuller (ADF) test for unit root. The null hypotheses of a normal distribution and a unit root are strongly rejected in all cases, whereas the other statistics disperse over a wide range.

5.2 Empirical results

In this section, we conduct an empirical exercise to extensively examine the out-of-sample performance of the conventional econometric methods, machine learning methods, and the hybrid methods. The selected methods are listed as follows:

1. ARX model: the simple autoregression model AR(22) with standard predictors presented in Table 3;
2. HARX model: the conventional HAR models with lag index $(1,5,22)$ and standard predictors presented in Table 3;
3. LASSO method: the LASSO-HAR method proposed in Audrino and Knaus (2016), where we use the LASSO method to select variables from the HAR model that incorporates all possible lag indices $(1,2,...,22)$ and standard predictors;

[22] In our exercises, we set the lower bound of the p-values of the Jarque-Bera and the ADF tests at 0.001. Values less than 0.001 are truncated at 0.001.

4. RFSV: the rough stochastic volatility model proposed by Gatheral et al. (2018);
5. RT: the regression tree method, in which the input variables are the HAR model that incorporates all possible lag indices $(1, 2, ..., 22)$ and standard predictors;
6. RF: the random forest method, in which the input variables are the HAR model that incorporates all possible lag indices $(1, 2, ..., 22)$ and standard predictors;
7. SVR-L: the support vector regression method with linear kernel, in which the input variables are the HAR model that incorporates all possible lag indices $(1, 2, ..., 22)$ and standard predictors;
8. MARF: the hybrid method that incorporates model averaging estimation at every leaf of every tree from the standard random forest method described in (vi).

We consider both short-horizon and long-horizon forecasts with $h = 1, 5, 10$ and 22. For assessing the out-of-sample performance, we calculate the following five statistics: (i) the mean squared forecast error (MSFE); (ii) the Gaussian quasi-likelihood (QLIKE) measure; (iii) the mean absolute forecast error (MAFE); (iv) the standard deviation of forecast error (SDFE); and (iv) the Mincer-Zarnowitz pseudo-R^2 for each candidate model at each forecast horizon h.

Statistics (i) to (iv) are described as the following:

$$\text{MAFE}(h) = \frac{1}{V}\sum_{j=1}^{V} |e_{T_j,h}|, \tag{172}$$

$$\text{MSFE}(h) = \frac{1}{V}\sum_{j=1}^{V} e^2_{T_j,h}, \tag{173}$$

$$\text{SDFE}(h) = \sqrt{\frac{1}{V-1}\left(e_{T_j,h} - \frac{1}{V}\sum_{j=1}^{V} e_{T_j,h}\right)^2}, \tag{174}$$

$$\text{QLIKE}(h) = \log \hat{y}_{T_j,h} + \frac{y_{T_j,h}}{\hat{y}_{T_j,h}}, \tag{175}$$

where $e_{T_j,h} = y_{T_j,h} - \hat{y}_{T_j,h}$ is the forecast error, $j = 1, 2, \ldots, V$, and $\hat{y}_{T_j,h}$ is the h-day ahead forecast with information up to T_j, where T_j stands for the last observation in each of the V rolling windows. Another widely-adopted method for evaluation is by means of the

R^2-criterion of the Mincer-Zarnowitz regression, given by

$$y_{T_j,h} = a + b\hat{y}_{T_j,h} + u_{T_j}, \text{for } j = 1, 2, \ldots, V. \tag{176}$$

For all of the exercises, we conduct a rolling window out-of-sample exercise. The window length is set at 3000, which is roughly half of the sample. Each of the above methods is applied to the data set, and a series of h days ahead direct forecasts are obtained. Table 5 presents some descriptive results of the out-of-sample evaluation for forecasts 1, 5, 10 and 22 days ahead, presented in Panels A to D, respectively. We report the MSFE, SDFE, MAFE, QLIKE and the pseudo R^2 listed in the first column from the rolling-window exercises for all methods presented in the first row of Table 5.

The machine learning method RT performs the worst consistently in Panels A to D, while the simple linear ARX model has good performance in many cases. In fact, when $h = 1$, the large pseudo R^2s by ARX and HARX imply that the conventional linear regression model is capable of explaining a large fraction of the total variation in the VIX data, which leaves small room for other more complicated methods to improve upon. The HARX method has the best performance among all when $h = 1$, which coincides with the findings in Fernandes et al. (2014).

As h increases, we notice that the criteria MSFE, QLIKE, MAFE, and SDFE increase and the pseudo R^2 decreases, since the forecasting accuracy of all methods decreases as the forecasting horizon increases. For $h = 5, 10$, and 22, the RFSV has the best performance by yielding the smallest forecasting bias by all statistics we considered. This intriguing results emphasize the importance of parsimonious in forecasting exercises.[23] In all panels, The hybrid method MARF is no worse than but also does not dominate the RF method, which suggests that heterogeneity is not a serious concern in the VIX data.

To further examine whether the out-performance is statistically significant, we perform

[23] Note that RFSV has only 3 parameters in our exercises.

Table 5: Out-of-sample forecast comparison of methods for VIX

Statistics	ARX	HARX	LASSO	RFSV	RT	RF	SVR-L	MARF
Panel A: $h = 1$								
MSFE	0.0046	0.0046	0.0145	0.0060	0.0083	0.0052	0.0046	0.0051
QLIKE	0.0003	0.0003	0.0008	0.0003	0.0005	0.0003	0.0003	0.0003
MAFE	0.0488	0.0486	0.0971	0.0572	0.0671	0.0520	0.0483	0.0519
SDFE	0.0679	0.0676	0.1204	0.0776	0.0909	0.0718	0.0675	0.0716
Pseudo R^2	0.9678	0.9681	0.8988	0.9579	0.9422	0.9639	0.9681	0.9641
Panel B: $h = 5$								
MSFE	0.0242	0.0241	0.0329	0.0181	0.0432	0.0262	0.0239	0.0262
QLIKE	0.0014	0.0014	0.0019	0.0011	0.0024	0.0015	0.0014	0.0015
MAFE	0.1174	0.1170	0.1446	0.1015	0.1551	0.1224	0.1144	0.1226
SDFE	0.1555	0.1552	0.1813	0.1346	0.2079	0.1618	0.1546	0.1620
Pseudo R^2	0.8314	0.8321	0.7708	0.8737	0.6986	0.8173	0.8334	0.8169
Panel C: $h = 10$								
MSFE	0.0404	0.0404	0.0477	0.0275	0.0710	0.0447	0.0400	0.0445
QLIKE	0.0023	0.0023	0.0027	0.0016	0.0039	0.0026	0.0023	0.0025
MAFE	0.1522	0.1520	0.1721	0.1259	0.1956	0.1611	0.1465	0.1601
SDFE	0.2011	0.2010	0.2183	0.1657	0.2664	0.2114	0.2000	0.2110
Pseudo R^2	0.7186	0.7187	0.6683	0.8088	0.5058	0.6888	0.7215	0.6900
Panel D: $h = 22$								
MSFE	0.0688	0.0689	0.0724	0.0447	0.1125	0.0756	0.0669	0.0758
QLIKE	0.0038	0.0038	0.0041	0.0025	0.0059	0.0041	0.0037	0.0042
MAFE	0.2007	0.2010	0.2131	0.1605	0.2544	0.2141	0.1883	0.2147
SDFE	0.2624	0.2624	0.2691	0.2114	0.3354	0.2750	0.2586	0.2753
Pseudo R^2	0.5230	0.5229	0.4983	0.6903	0.2209	0.4762	0.5366	0.4748

This table reports the out-of-sample results for predicting h-day future realized variation using the different methods. The results are based on the VIX data spanning from 1990-01-02 to 2017-11-20. We use a rolling window of 3000 observations to estimate the models, and evaluate the out-of-sample forecast performance at four horizons ($h = 1$, $h = 5$, $h = 10$ and $h = 22$). Each panel corresponds to a specific forecast horizon, which ranges from 1 day to 22 days.

the modified Giacomini-White test (Giacomini and White, 2006)[24] of the null hypothesis that the *column method* performs equally well as the *row method* in terms of absolute forecast errors. The corresponding p values are presented in Table 6 for $h = 1, 5, 10$, and 22 in Panels A to D, respectively. We see that the gains in forecast accuracy from the HARX relative to other methods are statistically significant at 5% level when $h = 1$. For other forecasting horizons, the RFSV method significantly outperforms all other methods even at 0.1% level.

[24] Giacomini and White (2006) proposed a framework for out-of-sample predictive ability testing and forecast selection designed for use in the realistic situation in which the forecasting model is possibly misspecified, due to unmodeled dynamics, unmodeled heterogeneity, incorrect functional form, or any combination of these. The null hypothesis of the GW test is that the two models we want to compare are equally accurate on average based on certain criterion.

Table 6: The Giacomini-White test for the mean absolute forecast errors

Method	ARX	HARX	LASSO	RFSV	RT	RF	SVR-L	MARF
Panel A: $h = 1$								
ARX	-	-	-	-	-	-	-	-
HARX	0.0271	-	-	-	-	-	-	-
LASSO	0.0000	0.0000	-	-	-	-	-	-
RFSV	0.0000	0.0000	0.0000	-	-	-	-	-
RT	0.0000	0.0000	0.0000	0.0000	-	-	-	-
RF	0.0000	0.0000	0.0000	0.0000	0.0000	-	-	-
SVR-L	0.0000	0.0089	0.0000	0.0000	0.0000	0.0000	-	-
MARF	0.0000	0.0000	0.0000	0.0000	0.0000	0.7595	0.0000	-
Panel B: $h = 5$								
ARX	-	-	-	-	-	-	-	-
HARX	0.1011	-	-	-	-	-	-	-
LASSO	0.0000	0.0000	-	-	-	-	-	-
RFSV	0.0000	0.0000	0.0000	-	-	-	-	-
RT	0.0000	0.0000	0.0050	0.0000	-	-	-	-
RF	0.0063	0.0034	0.0000	0.0000	0.0000	-	-	-
SVR-L	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-	-
MARF	0.0047	0.0026	0.0000	0.0000	0.0000	0.7062	0.0000	-
Panel C: $h = 10$								
ARX	-	-	-	-	-	-	-	-
HARX	0.5018	-	-	-	-	-	-	-
LASSO	0.0000	0.0000	-	-	-	-	-	-
RFSV	0.0000	0.0000	0.0000	-	-	-	-	-
RT	0.0000	0.0000	0.0002	0.0000	-	-	-	-
RF	0.0115	0.0093	0.0171	0.0000	0.0000	-	-	-
SVR-L	0.0000	0.0001	0.0000	0.0000	0.0000	0.0001	-	-
MARF	0.0236	0.0193	0.0095	0.0000	0.0000	0.2010	0.0003	-
Panel D: $h = 22$								
ARX	-	-	-	-	-	-	-	-
HARX	0.5743	-	-	-	-	-	-	-
LASSO	0.0614	0.0663	-	-	-	-	-	-
RFSV	0.0000	0.0000	0.0000	-	-	-	-	-
RT	0.0000	0.0000	0.0001	0.0000	-	-	-	-
RF	0.0500	0.0570	0.9048	0.0000	0.0000	-	-	-
SVR-L	0.0000	0.0000	0.0007	0.0000	0.0000	0.0004	-	-
MARF	0.0406	0.0466	0.8486	0.0000	0.0000	0.4488	0.0002	-

6 Empirical Illustration II: HICP Forecasting

Macroeconomic forecasting is an important but difficult task. Forecasting performance by conventional econometric methods is usually not quite satisfactory partially due to the restriction of the linear formulation. Recent literature begins to pay attention to more flexible machine learning methods. Jung et al. (2019) forecasted real GDP growth rates for seven countries using machine learning methods. By comparing the forecasting results with benchmark forecasts, Jung et al. (2019) demonstrated the benefits of adopting machine learning methods. Medeiros et al. (2019) explored advances in machine learning methods and the availability of new datasets to forecast U.S. inflation. They showed that machine learning methods with a large number of covariates are systematically more accurate than the benchmarks. In this exercise, we consider utilizing the forward-looking information from a Survey of Professional Forecasters (SPF) to forecast the harmonized index of consumer prices (HICP) for the euro area using both econometric and machine learning techniques.

Coincident with the launch of the euro currency in January 1999, the European Central Bank (ECB) started an SPF as part of its gathering of information and analysis of the euro area macroeconomic outlook. Genre et al. (2013) showed that a simple equally weighted pooling of forecasts performs quite well in practice relative to many other approaches that rely on estimated combination weights. We obtain the data from the SPF official website.[25] The raw data varies from 1999Q1 to 2018Q4 and totals 80 observations. We consider the data on the one-year-ahead prediction of HICP from 119 different forecasters. However, a specific forecaster may or may not submit a survey response throughout the whole period consistently. Therefore, we narrow down to 30 qualified forecasters that submit surveys consistently throughout the sample period.

Let y_t be the target HICP at period t. Denote x_{it} as the prediction by the i^{th} forecaster for period t, which is feasible one year ago. Method recommended by Genre et al. (2013)

[25] http://www.ecb.europa.eu/stats/prices/indic/forecast/html/index.en.html.

can be expressed as

$$y_t = \sum_{i=1}^{30} \beta_i x_{it} + u_t,$$

where β_i is fixed as $1/30$. Therefore, the equally weighted pooling method can be regarded as a restricted least squares estimation. An obvious alternative is the unrestricted OLS estimation.

We then further relax the linearity restriction and assume the following model:

$$y_t = f(\boldsymbol{x}_t) + u_t,$$

where $\boldsymbol{x}_t = [x_{1t}, ..., x_{30t}]^\top$ and the function $f(\cdot)$ maps the collection of forecasters to HICP in a possibly nonparametric manner.

We compare a list of machine learning specification for $f(\cdot)$. We present these specification along with simple averaging and OLS in the following:

1. Simple equal weight (Simple);
2. OLS;
3. Boosting (BOOST);
4. Regression Tree (RT);
5. Bagging (BAG);
6. Random Forest (RF);
7. Support Vector Regression with Linear Kernel (SVR-L);
8. Model Averaging Random Forest (MARF).

We conduct a rolling window forecasting exercise with window length set at 40. We evaluate the forecasting accuracy of the above methods by MSFE, QLIKE, SDFE, MAFE, and Pseudo-R^2. Comparison results are presented in Table 7.

Table 7: Forecast Accuracy Comparison

Statistics	Simple	OLS	BOOST	RT	BAG	RF	SVR-L	MARF
MSFE	0.2603	0.4445	0.4787	0.3182	0.1752	0.1789	0.1834	0.1726
QLIKE	0.2056	0.6695	0.1860	0.3047	0.1349	0.1413	2.0788	0.1389
MAFE	0.4078	0.4693	0.4839	0.4128	0.3543	0.3444	0.3473	0.3341
SDFE	0.5102	0.6667	0.6919	0.5641	0.4185	0.4230	0.4283	0.4155
Pseudo R^2	0.5371	0.2095	0.1487	0.4341	0.6885	0.6818	0.6738	0.6930

This table reports the out-of-sample results for predicting one-year-ahead HICP using the different methods. The results are based on the HICP data varying from 1999Q1 to 2018Q4. We use a rolling window of 40 observations to estimate the forecasts.

Although OLS imposes no restrictions on the coefficients, we note that the forecasting results by OLS are worse than simple averaging by all statistics. Which coincides with the results in Genre et al. (2013). It is also not a surprise that BOOST has overall bad performance due to its in-sample over-fitting. RT yields better results than OLS and BOOST but still worse than simple averaging. On the other hand, BAG, RF, and MARF all yield better forecasting results by all statistics. Moreover, SVR-L beats simple averaging in all statistics except QLIKE. In all statistics, the MARF method yields the best performance.

To examine if the improvement in forecasting accuracy is significant, we perform the Giacomini-White (GW) test of the null hypothesis that the column method performs equally well as the row method in terms of MAFE. The corresponding p-values are presented in Table 8. We pay our attention to the comparison between the benchmark simple averaging method and the rest. We note that only the MARF method significantly beats simple averaging at the 10% level.

Table 8: GW Test Results

Method	Simple	OLS	BOOST	RT	BAG	RF	SVR-L	MARF
Simple	-	-	-	-	-	-	-	-
OLS	0.5424	-	-	-	-	-	-	-
Boost	0.3915	0.8752	-	-	-	-	-	-
RT	0.9490	0.5343	0.2022	-	-	-	-	-
BAG	0.2527	0.1714	0.0961	0.3929	-	-	-	-
RF	0.1743	0.1325	0.0635	0.3224	0.5292	-	-	-
SVR-L	0.3571	0.0458	0.0744	0.3547	0.8723	0.9484	-	-
MARF	0.0898	0.1194	0.0497	0.2573	0.2219	0.3347	0.7698	-

7 Conclusions

This report reviews many techniques that can be used to forecast economic activities. In particular, we focus on two classes of forecasting methods, methods based on econometric models and methods based on machine learning techniques. Within the first class, both univariate models and multivariate models have been reviewed. We explain how to use these models to predict. With the class of multivariate models, both reduced-form models and structural models are reviewed. When reviewing structural models, we pay special attention to how economic theory can restrict relationships among variables. Within the second class, we review several leading machine learning methods, including multivariate adaptive regression splines, regression tree, bootstrap, bagging tree, random forecast, boosting tree, M5' algorithm, neural network, and support vector machine for regression. We also review several variable selection techniques introduced in the machine learning literature.

Deeply rooted in computer science, machine learning techniques aim to find how an output variable is related to input variables intending to produce predictions. They focus on identifying the "best" functional approximation, often in huge samples, for the purpose of predictions. Usually machine learning techniques cannot identify the causality nor take account of restrictions implied by economic theory. In addition, they do not normally care about importance or insight. Moreover, machine learning techniques are not interested in making statistical inference, such as testing a hypothesis that is implied by a certain economic theory. They cannot be used to perform scenario analysis or counterfactual analysis.[26]

Typically econometric methods deal with smaller samples and the focuses are on estimation and statistical inference. Not surprisingly, distributional assumptions, alternative estimation techniques, how to obtain sampling distributions, how to best approximate the

[26] That being said, special attention has been drawn towards identifying treatment effects using machine learning techniques recently. Pioneer studies including Wager and Athey (2018), Chernozhukov et al. (2018), among others are well-received in this burgeoning literature.

sampling distribution are some of the central issues in econometrics. If one wishes to take the economic theory seriously, then econometric models based on structural forms can be used. The use of *Ceteris Paribus* clauses has a long history in econometrics, to isolate the impact of one input variable on the output variable when there are other input variables. Econometric methods are typically based on analytical functions, facilitating scenario analysis and counter-factual analysis.

To take advantage of the strengthens of these two classes of methods, we propose a class of hybrid methods, including the split-sample method, its model averaging extension, and the model averaging tree methods. We show that machine learning presents great opportunities to cross-fertilize the field of the econometric forecast.

Finally, we compare the performance of the alternative methods in two applications based on real data. In the first application, we use eight methods to forecast VIX, including three econometric methods (namely ARX, HARX, RFSV), four machine learning methods (namely LASSO, RT, RF and SVR), and one hybrid method (namely MARF). It is found that when the forecasting horizon is short (one period), the best machine learning method matches the best econometric method. However, as the forecasting horizon increases, the best econometric method tends to outperform the best machine learning method. The dominance of econometric methods over machine learning methods is likely caused by a nearly linear relationship of the present volatility and the past volatilities and, in the meantime, by the lack of a very large sample and rich data in this forecasting exercise such that the advantage of machine learning techniques cannot be fully taken.

We also eight methods to forecast HICP, including two econometric methods (namely Simple, OLS), five machine learning methods (namely BOOST, RT, BAG, RF, SVR-L), and one hybrid method (namely MARF). It is found that the best machine learning method outperforms the best econometric method. This suggests that the actual inflation is related to the predictions of Professional Forecasters in a highly nonlinear way and can even involve interactive effects. Interestingly, the hybrid method always outperforms the best machine learning method.

References

AGUILAR, O. AND M. WEST (2000): "Bayesian dynamic factor models and portfolio allocation," *Journal of Business & Economic Statistics*, 18, 338–357.

AÏT-SAHALIA, Y. AND L. MANCINI (2008): "Out of sample forecasts of quadratic variation," *Journal of Econometrics*, 147, 17 – 33.

AKAIKE, H. (1973): "Information Theory and an Extension of the Maximum Likelihood Principle," *Second International Symposium on Information Theory*, 267–281.

AMEMIYA, T. (1980): "Selection of Regressors," *International Economic Review*, 21, 331–354.

AN, S. AND F. SCHORFHEIDE (2007): "Bayesian analysis of DSGE models," *Econometric reviews*, 26, 113–172.

ANDERSEN, T., T. BOLLERSLEV, F. DIEBOLD, AND H. EBENS (2001a): "The distribution of realized stock return volatility," *Journal of Financial Economics*, 61, 43–76.

ANDERSEN, T. G. AND T. BOLLERSLEV (1998): "Answering the Skeptics: Yes, Standard Volatility Models Do Provide Accurate Forecasts," *International Economic Review*, 39, 885–905.

ANDERSEN, T. G., T. BOLLERSLEV, AND F. X. DIEBOLD (2007): "Roughing It Up: Including Jump Components in the Measurement, Modeling, and Forecasting of Return Volatility," *The Review of Economics and Statistics*, 89, 701–720.

ANDERSEN, T. G., T. BOLLERSLEV, F. X. DIEBOLD, AND P. LABYS (2001b): "The Distribution of Realized Exchange Rate Volatility," *Journal of the American Statistical Association*, 96, 42–55.

——— (2003): "Modeling and Forecasting Realized Volatility," *Econometrica*, 71, 579–625.

ANDREWS, D. W. K. (2003): "Tests for Parameter Instability and Structural Change with Unknown Change Point: A Corrigendum," *Econometrica*, 71, 395–397.

ASAI, M., M. MCALEER, AND J. YU (2006): "Multivariate Stochastic Volatility: A Review," *Econometric Reviews*, 25, 145–175.

ATHEY, S. AND G. W. IMBENS (2019): "Machine Learning Methods Economists Should Know About," *Working Paper*.

AUDRINO, F. AND S. D. KNAUS (2016): "Lassoing the HAR Model: A Model Selection Perspective on Realized Volatility Dynamics," *Econometric Reviews*, 35, 1485–1521.

AUESTAD, B. AND D. TJØSTHEIM (1990): "Identification of nonlinear time series: First order characterization and order determination," *Biometrika*, 77, 669–687.

BAI, J. AND P. WANG (2015): "Identification and bayesian estimation of dynamic factor models," *Journal of Business & Economic Statistics*, 33, 221–240.

BAILLIE, R. T., T. BOLLERSLEV, AND H. O. MIKKELSEN (1996): "Fractionally integrated generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, 74, 3 – 30.

BARNDORFF-NEILSEN, O. E., S. KINNEBROUK, AND N. SHEPHARD (2010): "Measuring Downside Risk: Realised Semivariance," in *Volatility and Time Series Econometrics: Essays in Honor of Robert F. Engle*, ed. by T. Bollerslev, J. Russell, and M. Watson, Oxford University Press, 117–136.

BAUWENS, L., S. LAURENT, AND J. V. K. ROMBOUTS (2006): "Multivariate GARCH models: a survey," *Journal of Applied Econometrics*, 21, 79–109.

BELLONI, A. AND V. CHERNOZHUKOV (2012): "Supplement to 'Least Squares After Model Selection in High-dimensional Sparse Models'," *DOI:10.3150/11-BEJ410SUPP*.

——— (2013): "Least Squares After Model Selection in High-dimensional Sparse Models," *Bernoulli*, 19, 521–547.

BERNANKE, B. S., J. BOIVIN, AND P. ELIASZ (2005): "Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach*," *The Quarterly Journal of Economics*, 120, 387–422.

BIAU, O. AND A. D'ELIA (2010): "Euro Area GDP Forecast Using Large Survey Dataset - A Random Forest Approach," EcoMod2010 259600029, EcoMod.

BICKEL, P. J., Y. RITOV, AND A. B. TSYBAKOV (2009): "Simultaneous Analysis of Lasso and Dantzig Selector," *The Annals of Statistics*, 37, 1705–1732.

BLANCHARD, O. J. AND D. QUAH (1989): "The dynamic effects of aggregate demand and aggregate supply," *The American Economic Review*, 79, 655–73.

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

BOLLERSLEV, T., J. LITVINOVA, AND G. TAUCHEN (2006): "Leverage and Volatility Feedback Effects in High-Frequency Data," *Journal of Financial Econometrics*, 4, 353–384.

BREIMAN, L. (1996): "Bagging Predictors," *Machine Learning*, 26, 123–140.

——— (2001): "Random Forests," *Machine Learning*, 45, 5–32.

BREIMAN, L., J. FRIEDMAN, AND C. J. STONE (1984): *Classification and Regression Trees*, Chapman and Hall/CRC.

BRITTEN-JONES, M. AND A. NEUBERGER (2000): "Option Prices, Implied Price Processes, and Stochastic Volatility," *The Journal of Finance*, 55, 839–866.

CANDES, E. AND T. TAO (2007): "The Dantzig Selector: Statistical Estimation when p is Much Larger than n," *The Annals of Statistics*, 35, 2313–2351.

CESA-BIANCHI, A., L. CESPEDES, AND A. REBUCCI (2015): "Global Liquidity, House Prices, and the Macroeconomy: Evidence from Advanced and Emerging Economies," *Journal of Money, Credit and Banking*, 47, 301–335.

CHAUDHURI, P., W.-D. LO, W.-Y. LOH, AND C.-C. YANG (1995): "Bagging Predictors," *Generalized Regression Trees*, 5, 641–666.

CHEN, Y., W. K. HÄRDLE, AND U. PIGORSCH (2010): "Localized Realized Volatility Modeling," *Journal of the American Statistical Association*, 105, 1376–1393.

CHERNOZHUKOV, V., D. CHETVERIKOV, M. DEMIRER, E. DUFLO, C. HANSEN, W. NEWEY, AND J. ROBINS (2018): "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*, 21, C1–C68.

CHIB, S. AND S. RAMAMURTHY (2010): "Tailored randomized block MCMC methods with application to DSGE models," *Journal of Econometrics*, 155, 19–38.

CHOW, G. C. (1960): "Tests of equality between sets of coefficients in two linear regressions," *Econometrica: Journal of the Econometric Society*, 591–605.

CHRISTENSEN, B. J. AND M. R. NIELSEN (2007): "The Effect of Long Memory in Volatility on Stock Market Fluctuations," *The Review of Economics and Statistics*, 89, 684–700.

CHUKU, C., A. SIMPASA, AND J. ODUOR (2019): "Intelligent forecasting of economic growth for developing economies," *International Economics*, 159, 74 – 93.

COMTE, F. AND E. RENAULT (1996): "Long memory continuous time models," *Journal of Econometrics*, 73, 101 – 149.

——— (1998): "Long memory in continuous-time stochastic volatility models," *Mathematical Finance*, 8, 291–323.

CORSI, F. (2009): "A Simple Approximate Long-Memory Model of Realized Volatility," *Journal of Financial Econometrics*, 7, 174–196.

CORSI, F., F. AUDRINO, AND R. RENÒ (2012): "HAR Modeling for Realized Volatility Forecasting," in *Handbook of Volatility Models and Their Applications*, John Wiley & Sons, Inc., 363–382.

CORSI, F., D. PIRINO, AND R. RENÒ (2010): "Threshold bipower variation and the impact of jumps on volatility forecasting," *Journal of Econometrics*, 159, 276 – 288.

CRAIOVEANU, M. AND E. HILLEBRAND (2012): "Why It Is OK to Use the HAR-RV $(1,5,21)$ Model," *Technical Report*, University of Central Missouri.

DACOROGNA, M. M., U. A. MŰLLER, R. J. NAGLER, R. B. OLSEN, AND O. V. PICTET (1993): "A geographical model for the daily and weekly seasonal volatility in the foreign exchange market," *Journal of International Money and Finance*, 12, 413 – 438.

DANNE, C. (2015): "VARsignR: Estimating VARs using sign restrictions in R," *Working Paper*.

DEL NEGRO, M. AND F. SCHORFHEIDE (2013): "DSGE model-based forecasting," in *Handbook of economic forecasting*, Elsevier, vol. 2, 57–140.

DIEBOLD, F. X. (2006): *Elements of Forecasting*, South-Western College Publishing.

DING, J., V. TAROKH, AND Y. YANG (2019): "Optimal variable selection in regression models," *Working Paper*.

DING, Z., C. W. GRANGER, AND R. F. ENGLE (1993): "A long memory property of stock market returns and a new model," *Journal of Empirical Finance*, 1, 83 – 106.

DRUCKER, H., C. J. C. BURGES, L. KAUFMAN, A. J. SMOLA, AND V. VAPNIK (1996): "Support Vector Regression Machines," in *Advances in Neural Information Processing Systems 9*, ed. by M. C. Mozer, M. I. Jordan, and T. Petsche, MIT Press, 155–161.

EFRON, B. (1979): "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, 7, 1–26.

ENGLE, R. (2002): "Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroskedasticity Models," *Journal of Business & Economic Statistics*, 20, 339–350.

ENGLE, R. F. (1982): "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, 50, 987–1007.

ENGLE, R. F. AND G. M. GALLO (2006): "A Multiple Indicators Model for Volatility Using Intra-daily Data," *Journal of Econometrics*, 131, 3 – 27.

ENGLE, R. F. AND K. F. KRONER (1995): "Multivariate Simultaneous Generalized Arch," *Econometric Theory*, 11, 122–150.

FAN, J. AND R. LI (2001): "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, 96, 1348–1360.

FERNANDES, M., M. C. MEDEIROS, AND M. SCHARTH (2014): "Modeling and Predicting the CBOE Market Volatility Index," *Journal of Banking & Finance*, 40, 1–10.

FERNÁNDEZ-VILLAVERDE, J. (2010): "The econometrics of DSGE models," *SERIEs*, 1, 3–49.

FERNÁNDEZ-VILLAVERDE, J. AND J. F. RUBIO-RAMÍREZ (2005): “Estimating dynamic equilibrium economies: linear versus nonlinear likelihood,” *Journal of Applied Econometrics*, 20, 891–910.

FRAGOSO, T. M., W. BERTOLI, AND F. LOUZADA (2018): “Bayesian model averaging: A systematic review and conceptual classification,” *International Statistical Review*, 86, 1–28.

FREUND, Y. AND R. E. SCHAPIRE (1997): “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Journal of Computer and System Sciences*, 55, 119 – 139.

FRIEDMAN, J. H. (1991): “Multivariate Adaptive Regression Splines,” *The Annals of Statistics*, 19, 1–67.

GATHERAL, J., T. JAISSON, AND M. ROSENBAUM (2018): “Volatility is rough,” *Quantitative Finance*, 18, 933–949.

GENRE, V., G. KENNY, A. MEYLER, AND A. TIMMERMANN (2013): “Combining expert forecasts: Can anything beat the simple average?” *International Journal of Forecasting*, 29, 108 – 121.

GEWEKE, J. (1977): “The dynamic factor analysis of economic time-series models,” in *Latent Variables in Socio-Economic Models*, ed. by D. Aigner and A. Goldberger, North-Holland.

GEWEKE, J. AND S. PORTER-HUDAK (1983): “The Estimation and Application of Long Meomory Time Series Models,” *Journal of Time Series Analysis*, 4, 221–238.

GHYSELS, E. AND M. MARCELLINO (2018): *Applied Economic Forecasting using Time Series Methods*, Oxford University Press.

GIACOMINI, R. AND H. WHITE (2006): “Tests of Conditional Predictive Ability,” *Econometrica*, 74, 1545–1578.

GIANNONE, D., L. REICHLIN, AND L. SALA (2004): “Monetary Policy in Real Time,” *NBER Macroeconomics Annual*, 19, 161–200.

GRANGER, C. W. J. AND R. JOYEUX (1980): “An Introduction to Long-memory Time Series Models and Fractional Differencing,” *Journal of Time Series Analysis*, 1, 15–29.

HANNAN, E. J. AND B. G. QUINN (1979): “The Determination of the Order of an Autoregession,” *Journal of the Royal Statistical Society. Series B (Methodological)*, 41, 190–195.

HANSEN, B. E. (2008): “Least-squares forecast averaging,” *Journal of Econometrics*, 146, 342–350.

——— (2012): "Time Series and Forecasting," *Lecture Notes*.

HANSEN, B. E. AND J. S. RACINE (2012): "Jackknife Model Averaging," *Journal of Econometrics*, 167, 38–46.

HANSEN, P. R., Z. HUANG, AND H. H. SHEK (2012): "Realized GARCH: a joint model for returns and realized measures of volatility," *Journal of Applied Econometrics*, 27, 877–906.

HÄRDLE, W., H. LÜTKEPOHL, AND R. CHEN (1997): "A Review of Nonparametric Time Series Analysis," *International Statistical Review*, 65, 49–72.

HARVEY, A. (1990): *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press.

HARVEY, A., E. RUIZ, AND N. SHEPHARD (1994): "Multivariate Stochastic Variance Models," *Review of Economic Studies*, 61, 247–264.

HASTIE, T., R. TIBSHIRANI, AND J. FRIEDMAN (2009): *The Elements of Statistical Learning:*, Springer Series in Statistics, New York, NY, USA: Springer New York Inc.

HASTIE, T., R. TIBSHIRANI, AND M. WAINWRIGHT (2015): *Statistical learning with sparsity: the lasso and generalizations*, Chapman and Hall/CRC.

HENDRY, D. F. AND B. NIELSEN (2007): *Econometric Modeling: A Likelihood Approach*, , Princeton University Press, chap. 19, 286–301.

HERBST, E. (2010): "Gradient and Hessian-based MCMC for DSGE models,(2010)," *Unpublished manuscript*.

HERBST, E. P. AND F. SCHORFHEIDE (2015): *Bayesian estimation of DSGE models*, Princeton University Press.

HESTON, S. L. (1993): "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options," *Review of Financial Studies*, 6, 327–343.

HIRANO, K. AND J. H. WRIGHT (2017): "Forecasting With Model Uncertainty: Representations and Risk Reduction," *Econometrica*, 85 (2), 617–643.

HOREL, A. (1962): "Applications of Ridge Analysis Toregression Problems," *Chem. Eng. Progress.*, 58, 54–59.

HOSKING, J. R. M. (1981): "Fractional differencing," *Biometrika*, 68, 165–176.

HÄRDLE, W. AND P. VIEU (1992): "Kernel Regression Smoothing of Time Series," *Journal of Time Series Analysis*, 13, 209–232.

HUANG, X. AND G. TAUCHEN (2005): "The Relative Contribution of Jumps to Total Price Variance," *Journal of Financial Econometrics*, 3, 456–499.

HULL, J. AND A. WHITE (1987): "The Pricing of Options on Assets with Stochastic Volatilities," *The Journal of Finance*, 42, 281–300.

HURICH, C. M. AND C.-L. TSAI (1989): "Regression and time series model selection in small samples," *Biometrika*, 76, 297–307.

ING, C.-K. AND C.-Z. WEI (2003): "On same-realization prediction in an infinite-order autoregressive process," *Journal of Multivariate Analysis*, 85, 130 – 155.

JACQUIER, E., N. G. POLSON, AND P. ROSSI (1994): "Bayesian Analysis of Stochastic Volatility Models," *Journal of Business & Economic Statistics*, 12, 371–89.

——— (2004): "Bayesian analysis of stochastic volatility models with fat-tails and correlated errors," *Journal of Econometrics*, 122, 185–212.

JAMES, G., D. WITTEN, T. HASTIE, AND R. TIBSHIRANI (2013): *An introduction to statistical learning*, vol. 112, Springer.

JIANG, G. J. AND Y. S. TIAN (2005): "The Model-Free Implied Volatility and Its Information Content," *Review of Financial Studies*, 18, 1305–1342.

JIANG, L., X. WANG, AND J. YU (2019): "In-fill asymptotic theory for structural break point in autoregression: A unified theory," *Working Paper*.

JUNG, J.-K., M. PATNAM, AND A. TER-MARTIROSYAN (2019): "An Algorithmic Crystal Ball: Forecasts-based on Machine Learning," *IMF Working Paper*.

KALMAN, R. E. (1960): "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, 82, 35–45.

KEMP, G. C. (1997): "Linear Combinations of Stationary Processes—Solution," *Econometric Theory*, 13, 897–898.

KIM, H. AND W.-Y. LOH (2003): "Classification Trees With Bivariate Linear Discriminant Node Models," *Journal of Computational and Graphical Statistics*, 12, 512–530.

KIM, S., N. SHEPHARD, AND S. CHIB (1998): "Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models," *The Review of Economic Studies*, 65, 361–393.

KREISS, J.-P. AND S. N. LAHIRI (2012): "Bootstrap Methods for Time Series," in *Time Series Analysis: Methods and Applications, Volume 30*, ed. by T. S. Rao, S. S. Rao, and C. Rao, North Holland, chap. 1, 3–26.

KUERSTEINER, G. AND R. OKUI (2010): "Constructing Optimal Instruments by First-Stage Prediction Averaging," *Econometrica*, 78, 697–718.

KULPERGER, R. J. AND B. L. S. PRAKASA RAO (1989): "Bootstrapping a Finite State Markov Chain," *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 51, 178–191.

KÜNSCH, H. R. (1989): "The Jackknife and the Bootstrap for General Stationary Observations," *The Annals of Statistics*, 17, 1217–1241.

LEHRER, S. F. AND T. XIE (2017): "Box Office Buzz: Does Social Media Data Steal the Show from Model Uncertainty When Forecasting for Hollywood?" *The Review of Economics and Statistics*, 99, 749–755.

——— (2018): "The Bigger Picture: Combining Econometrics with Analytics Improve Forecasts of Movie Success," *Working Paper*.

LEHRER, S. F., T. XIE, AND X. ZHANG (2018): "Twits versus Tweets: Does Adding Social Media Wisdom Trump Admitting Ignorance when Forecasting the CBOE VIX?" *Working Paper*.

LI, Y., T. ZENG, AND J. YU (2019): "Deviance Information Criterion for Model Selection: Justification and Variation," *Working Paper*.

LIN, Y.-N. (2007): "Pricing VIX futures: Evidence from integrated physical and risk-neutral probability measures," *Journal of Futures Markets*, 27, 1175–1217.

LIU, Y. AND T. XIE (2018): "Machine Learning Versus Econometrics: Prediction of Box Office," *Applied Economics Letter*, Forthcoming.

LO, A. W. (1991): "Long-Term Memory in Stock Market Prices," *Econometrica*, 59, 1279–1313.

LUBIK, T. AND F. SCHORFHEIDE (2005): "A Bayesian look at new open economy macroeconomics," *NBER macroeconomics annual*, 20, 313–366.

——— (2006): "Do central banks respond to exchange rate fluctuation–a structural investigation," *Journal of Monetary Economics*, 313–366.

MALLOWS, C. L. (1973): "Some Comments on C_p," *Technometrics*, 15, 661–675.

MCALEER, M. AND M. C. MEDEIROS (2008): "A multiple regime smooth transition Heterogeneous Autoregressive model for long memory and asymmetries," *Journal of Econometrics*, 147, 104–119.

MCCULLOCH, W. S. AND W. PITTS (1943): "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, 5, 115–133.

MEDEIROS, M. C., G. F. R. VASCONCELOS, ÁLVARO VEIGA, AND E. ZILBERMAN (2019): "Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods," *Journal of Business & Economic Statistics*, Forthcoming.

Müller, U. A., M. M. Dacorogna, R. D. Davé, O. V. Pictet, R. B. Olsen, and J. Ward (1993): “Fractals and Intrinsic Time - a Challenge to Econometricians,” Tech. rep.

Neal, R. M. (1996): *Bayesian Learning for Neural Networks*, Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Nelson, D. B. (1991): “Conditional Heteroskedasticity in Asset Returns: A New Approach,” *Econometrica*, 59, 347–370.

Nelson, D. B. and C. Q. Cao (1992): “Inequality Constraints in the Univariate GARCH Model,” *Journal of Business & Economic Statistics*, 10, 229–235.

Patton, A. J. and K. Sheppard (2015): “Good Volatility, Bad Volatility: Signed Jumps and The Persistence of Volatility,” *The Review of Economics and Statistics*, 97, 683–697.

Pesaran, M. H. and A. Pick (2011): “Forecast Combination across Estimation Windows,” *Journal of Business & Economic Statistics*, 29, 307–318.

Pesaran, M. H., A. Pick, and M. Pranovich (2013): “Optimal forecasts in the presence of structural breaks,” *Journal of Econometrics*, 177, 134–152.

Pesaran, M. H. and A. Timmermann (2007): “Selection of estimation window in the presence of breaks,” *Journal of Econometrics*, 137, 134–161.

Quinlan, J. R. (1992): “Learning With Continuous Classes,” World Scientific, 343–348.

Robinson, P. (2001): “The memory of stochastic volatility models,” *Journal of Econometrics*, 101, 195 – 218.

Robinson, P. M. (1983): “Nonparametric Estimators for Time Series,” *Journal of Time Series Analysis*, 4, 185–207.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986): “Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1,” Cambridge, MA, USA: MIT Press, chap. Learning Internal Representations by Error Propagation, 318–362.

Sargent, T. and C. Sims (1977): “Business cycle modeling without pretending to have too much a priori economic theory,” Working Papers 55, Federal Reserve Bank of Minneapolis.

Scharth, M. and M. Medeiros (2009): “Asymmetric effects and long memory in the volatility of Dow Jones stocks,” *International Journal of Forecasting*, 25, 304–327.

Schorfheide, F. (2000): “Loss function-based evaluation of DSGE models,” *Journal of Applied Econometrics*, 15, 645–670.

SCHWARZ, G. (1978): “Estimating the Dimension of a Model,” *The Annals of Statistics*, 6, 461–464.

SENTANA, E. (1995): “Quadratic ARCH Models,” *The Review of Economic Studies*, 62, 639–661.

SHAO, J. (1997): “An Asymptotic Theory for Linear Model Selction,” *Statistica Sinica*, 7, 221–242.

SHEPHARD, N., ed. (2005): *Stochastic Volatility: Selected Readings*, Oxford University Press.

SOWELL, F. (1992): “Maximum likelihood estimation of stationary univariate fractionally integrated time series models,” *Journal of Econometrics*, 53, 165 – 188.

SPIEGELHALTER, D. J., N. G. BEST, B. P. CARLIN, AND A. VAN DER LINDE (2002): “Bayesian measures of model complexity and fit,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 583–639.

STOCK, J. H. AND M. W. WATSON (2002): “Forecasting Using Principal Components from a Large Number of Predictors,” *Journal of the American Statistical Association*, 97, 1167–1179.

——— (2011): “Dynamic Factor Models,” in *The Oxford Handbook of Economic Forecasting*, ed. by M. P. Clements and D. F. Hendry, Oxford University Press.

STRID, I., P. GIORDANI, AND R. KOHN (2010): “Adaptive hybrid Metropolis-Hastings samplers for DSGE models,” Tech. rep., SSE/EFI Working Paper Series in Economics and Finance.

TIBSHIRANI, R. (1996): “Regression Shrinkage and Selection Via the Lasso,” *Journal of the Royal Statistical Society, Series B*, 58, 267–288.

TIFFIN, A. J. (2016): “Seeing in the Dark; A Machine-Learning Approach to Nowcasting in Lebanon,” IMF Working Papers 16/56, International Monetary Fund.

TONG, H. AND K. LIM (1980): “Threshold autoregression, limit cycles and cyclical data-with discussion,” *Journal of the Royal Statistical Society. Series B*, 42.

TSAY, R. AND R. CHEN (2018): *Nonlinear Time Series Analysis*, Wiley Series in Probability and Statistics, Wiley.

UHLIG, H. (2005): “What are the effects of monetary policy on output? Results from an agnostic identification procedure,” *Journal of Monetary Economics*, 52, 381–419.

VAPNIK, V. N. (1996): *The Nature of Statistical Learning Theory*, New York, NY, USA: Springer-Verlag New York, Inc.

VENS, C. AND H. BLOCKEEL (2006): "A Simple Regression Based Heuristic for Learning Model Trees," *Intell. Data Anal.*, 10, 215–236.

VORTELINOS, D. I. (2017): "Forecasting realized volatility: HAR against Principal Components Combining, neural networks and GARCH," *Research in International Business and Finance*, 39, 824 – 839.

WAGER, S. AND S. ATHEY (2018): "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests," *Journal of the American Statistical Association*, 113, 1228–1242.

WANG, T., Y. SHEN, Y. JIANG, AND Z. HUANG (2017): "Pricing the CBOE VIX Futures with the Heston–Nandi GARCH Model," *Journal of Futures Markets*, 37, 641–659.

WANG, X., W. XIAO, AND J. YU (2019): "Estimation and Inference of Fractional Continuous-Time Model with Discrete-Sampled Data," *Working Paper*.

WANG, Y., F. MA, Y. WEI, AND C. WU (2016): "Forecasting Realized Volatility in A Changing World: A Dynamic Model Averaging Approach," *Journal of Banking & Finance*, 64, 136–149.

WANG, Y. AND I. H. WITTEN (1997): "Inducing Model Trees for Continuous Classes," in *In Proc. of the 9th European Conf. on Machine Learning Poster Papers*, 128–137.

WHALEY, R. E. (2000): "The Investor Fear Gauge," *The Journal of Portfolio Management*, 26, 12–17.

WOLD, H. O. A. (1966): "Estimation of principal components and related models by iterative least squares," in *Multivariate analysis*, ed. by P. R. Krishnajah, NewYork: Academic Press.

XIE, T. (2015): "Prediction Model Averaging Estimator," *Economics Letters*, 131, 5–8.

——— (2017): "Heteroscedasticity-robust Model Screening: A Useful Toolkit for Model Averaging in Big Data Analytics," *Economics Letters*, accepted.

YU, J. (2002): "Forecasting volatility in the New Zealand stock market," *Applied Financial Economics*, 12, 193–202.

——— (2005): "On leverage in a stochastic volatility model," *Journal of Econometrics*, 127, 165–178.

——— (2012): "A semiparametric stochastic volatility model," *Journal of Econometrics*, 167, 473–482.

YU, J. AND R. MEYER (2006): "Multivariate Stochastic Volatility Models: Bayesian Estimation and Model Comparison," *Econometric Reviews*, 25, 361–384.

ZHANG, J. E. AND Y. ZHU (2006): "VIX Futures," *Journal of Futures Markets*, 26, 521–531.

ZHANG, X., A. T. WAN, AND G. ZOU (2013): "Model Averaging by Jackknife Criterion in Models with Dependent Data," *Journal of Econometrics*, 174, 82–94.

ZHANG, X., G. ZOU, AND R. J. CARROLL (2015): "Model Averaging Based on Kullback-Leibler Distance," *Statistica Sinica*, 25, 1583–1598.

ZHU, S.-P. AND G.-H. LIAN (2012): "An Analytical Formula for VIX Futures and Its Applications," *Journal of Futures Markets*, 32, 166–190.

ZOU, H. (2006): "The adaptive lasso and its oracle properties," *Journal of the American statistical association*, 101, 1418–1429.

ZOU, H. AND T. HASTIE (2005): "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67, 301–320.

ZOU, H. AND H. H. ZHANG (2009): "On the adaptive elastic-net with a diverging number of parameters," *Annals of statistics*, 37, 1733.