Econometrics of Machine Learning Methods in Economic Forecasting*

Andrii Babii [†] Eric Ghysels [‡] Jonas Striaukas [§]

December 31, 2023

Abstract

We review the recent methodological advances in machine learning for economic forecasting and nowcasting. We consider the high-dimensional regularized regressions for individual time series and panel data, paying special attention to how time series lags and cross-validation should be used in practice. We also discuss how to do inference and tests such as the Granger causality test with high-dimensional regularized regressions. Lastly, we review the practical implementation of tree-based methods (boosting and random forests) and (deep) neural networks. We refer the reader to the Python and R libraries that can be used to compute the reviewed methods whenever possible.

Keywords: Machine learning, economic forecasting and nowcasting, panel data, MIDAS regressions, boosted trees, deep learning.

*We thank the Editors for their valuable feedback on a first draft of this chapter.

[†]University of North Carolina at Chapel Hill - Gardner Hall, CB 3305 Chapel Hill, NC 27599-3305. Email: babii.andrii@gmail.com.

[‡]Department of Economics and Kenan-Flagler Business School, University of North Carolina–Chapel Hill and CEPR. Email: eghysels@unc.edu.

[§]Department of Finance, Copenhagen Business School, Frederiksberg, Denmark. Email: jonas.striaukas@gmail.com.

1 Introduction

Economic forecasting has traditionally relied on models estimated with the maximum likelihood (MLE) approach. The limitations of the MLE are well known as eloquently described in Bradley and Trevor (2021):

"Arguably the 20th century's most influential piece of applied mathematics, maximum likelihood continues to be a prime method of choice in the statistician's toolkit. Roughly speaking, maximum likelihood provides nearly unbiased estimates of nearly minimum variance, and does so in an automatic way. That being said, maximum likelihood estimation has shown itself to be an inadequate and dangerous tool in many 21st century applications. Again speaking roughly, unbiased can be an unavoidable luxury when there are hundreds or thousands of parameters to estimate at the same time."

James and Stein (1961) made this point dramatically in a much simpler setting showing that the maximum likelihood estimator is inadmissible when the dimension is greater or equal to 3. In particular, biased shrinkage estimators and more generally regularized estimators outperform the conventional unbiased maximum likelihood estimator. The machine learning (ML) methods developed over roughly the past 60 years have revolutionized decision-making across various fields. At its core, ML involves formulating a loss or cost function for forecasting rules. In this context, a forecasting rule, denoted as $f(x_t)$, predicts the value of a target variable, y_{t+h}, at a future horizon, h, based on information available at time t. The loss function, $\ell(y_{t+h}, f(x_t))$, quantifies the error incurred by the forecasted value compared to the actual outcome.

The central goal is to approximate the optimal decision rule, f^*, which minimizes the expected loss, $\mathbb{E}[\ell(y_{t+h}, f(x_t))]$. This approach has its roots in the decision theory, see Wald (1949), and is adopted in statistical learning, see Vapnik (1999), and economic forecasting, see Granger and Pesaran (2000). For instance, when employing a quadratic loss function, $\ell(y_{t+h}, f(x_t)) = (y_{t+h} - f(x_t))^2$, the optimal decision rule corresponds to the (non-linear) regression, $f^*(x_t) = \mathbb{E}[y_{t+h}|x_t]$ with respect to $f(x_t)$.[1]

The data-driven decision rules lead to the bias-variance trade-off in the forecasting performance. Flexible nonparametric techniques offer a solution by reducing bias at the cost of increasing variance, leading to potential overfitting issues. At the same time, regularization and dimensionality reduction introduce some bias to reduce the variance. Machine learning offers a wide array of nonparametric and high-dimensional tools, enabling flexible and accurate approximations of the optimal

[1]Equivalently, we could consider the regression model, $y_{t+h} = f^*(x_t) + \varepsilon_{t+h}$ with $\mathbb{E}[\varepsilon_{t+h}|x_t] = 0$.

decision rules, adapting to the bias-variance trade-off, and optimizing the forecasting performance.

Many of the widely used ML tools relate to known and well-established statistical methods. For example, deep learning can be understood as a regression model with nonlinearities generated by a multi-layer neural network; see Hornik, Stinchcombe, and White (1990) and Chen (2007).[2] Random forests and gradient boosting which can be understood as a new generation of regression and classification trees; see Breiman, Friedman, Stone, and Olshen (1984). The penalized regression can be traced back to the idea of shrinkage, see James and Stein (1961), regularization of ill-posed inverse problems, see Tikhonov (1963), and the ridge regression, see Hoerl and Kennard (1970a,b).[3]

While the development of ML methods has a long history, the remarkable recent success and wide adoption are mostly due to the increasing availability of new high-dimensional data, cheap computational power, and scalable statistical packages.[4] Economists also rely increasingly on high-dimensional datasets such as textual and image data, credit card spending, or Google Trends. Consequently, ML methods are gaining appreciation and are becoming ubiquitous in economics and finance.

In this chapter, we aim to review some of the recent developments in the machine learning literature for economic forecasting, focusing on the appropriate treatment of time series lags, panel and tensor data, nowcasting, high-dimensional Granger causality tests, time series cross-validations, and classification. We also review the nonlinear tree-based methods and (deep) neural networks. Hence, this chapter is focused on topics of interest to forecasters and we refer the reader to other existing surveys and introductions to the ML methods for a more general review of the subject.[5]

[2]Various forms of neural networks have achieved a remarkable performance recently with perceptual data like text, images, speech, or videos; see also Farrell, Liang, and Misra (2021) and Gu, Kelly, and Xiu (2020) for applications with tabular data.

[3]See also Carrasco, Florens, and Renault (2007) and Babii and Florens (2017).

[4]One may quote the remarkable success of ML methods in the prediction contests with substantial monetary prizes, such as Kaggle or Makridakis Competitions; see Makridakis, Spiliotis, and Assimakopoulos (2020, 2022).

[5]See James, Witten, Hastie, and Tibshirani (2013), Hastie, Tibshirani, Friedman, and Friedman (2009), and Breiman (2001b) for general introductions. See also Mullainathan and Spiess (2017), Athey and Imbens (2019), and Varian (2014) for economics surveys. Lastly, see Coulombe, Leroux, Stevanovic, and Surprenant (2022) and Masini, Medeiros, and Mendes (2023) for time series reviews.

2 High-Dimensional Linear Projections

2.1 Time Series Forecasting

The empirical analysis of time series data entails several notable challenges. Firstly, in a data-rich time series environment the objective is often to forecast a low-frequency variable (e.g. quarterly GDP growth or inflation) while the information set may contain predictors measured at a higher frequency (e.g. monthly or daily). Additionally, certain economic effects tend to persist over time. This brings us to the question of how to combine the high (or same) frequency time series lags in regression equations.

Secondly, the prevalence of high-dimensional datasets further compounds the complexity. In addition to traditional macroeconomic and financial indicators, modern empirical research increasingly relies on non-standard data sources like textual data, credit card spending records, traffic and satellite data, among others. Consequently, the task of selecting an accurate forecasting model from this vast array of predictors becomes a significant challenge. Shrinkage methods like ridge regression or LASSO effectively mitigate multicollinearity and overfitting issues, making them particularly suited for handling large predictor sets.

To address the aforementioned challenges, Babii, Ghysels, and Striaukas (2022) introduce high-dimensional regularized projections for time series data inspired by the mixed-frequency data sampling, i.e. MIDAS regression or the distributed lag econometric literature (see Ghysels, Santa-Clara, and Valkanov (2006)). Let $(y_t)_{t\in[T]}$ be a target time series, e.g. quarterly GDP growth or inflation (where we put $[T] = \{1, 2, \ldots, T\}$ for a positive integer T). The covariates consist of K time-varying predictors measured potentially at higher frequencies, e.g. quarterly, monthly, or daily,

$$\left\{x_{t-j/n_k^H,k} : , t \in [T], j = 0, \ldots, n_k^L n_k^H - 1, k \in [K]\right\},$$

where n_k^H is the number of high-frequency observations for the k^{th} covariate in a low-frequency time t, and n_k^L is the number of low-frequency periods used as lags. For instance, $n_k^L = 1$ corresponds to a single quarter of high-frequency lags used as covariates and $n_k^H = 3$ corresponds to 3 months of data available per quarter.

The mixed frequency time series regression equation for forecasting a low -frequency target y_{t+h} at a horizon $h \geq 1$ is

$$y_{t+h} = \alpha + \sum_{j=0}^{J} \rho_j y_{t-j} + \sum_{k=1}^{K} \psi(L^{1/n_k^H}; \beta_k) x_{t,k} + u_{t+h},$$

where we use the lag polynomial notation $\psi(L^{1/n_k^H};\beta_k)x_{t,k} = \frac{1}{m_k}\sum_{j=0}^{m_k-1}\beta_{j,k}x_{t-j/n_k^H,k}$, where $m_k = n_k^L n_k^H$ is the total number of all lags. The resulting projection model has a large number of parameters and is prone to overfitting.

Babii, Ghysels, and Striaukas (2022) propose to parameterize the lag coefficients using a MIDAS weighting function ω described by a low-dimensional parameter $\beta_k \in \mathbf{R}^L$ $\psi(L^{1/n_k^H};\beta_k)x_{t,k} = \frac{1}{m_k}\sum_{j=0}^{m_k-1}\omega\left(\frac{j}{n_k^H};\beta_k\right)x_{t-j/n_k,k}$, where $\omega(s;\beta_k) = \sum_{l=0}^{L-1}\beta_{l,k}w_l(s)$ and $(w_l)_{l\geq 0}$ is a collection of approximating functions, called *dictionary*. The default choice for the dictionary could be a set of Legendre polynomials shifted to $[0, n_k^L]$ interval.[6] Given this choice, the forecasting equation is mapped to the linear regression model, where covariates are weighted by a matrix generated from the weight function. Importantly, the time series lags define the sparse-group structure, where a group of coefficients $\beta_k \in \mathbf{R}^L$ corresponding to the k^{th} covariate, is approximately sparse.

Next, Babii, Ghysels, and Striaukas (2022) propose to use the sparse-group LASSO (sg-LASSO) estimator of Simon, Friedman, Hastie, and Tibshirani (2013), namely:

$$\min_{b\in\mathbf{R}^p}\|\mathbf{y}-\mathbf{X}b\|_T^2 + \lambda\Omega(b),$$

where $\|.\|_T = |.|_2/\sqrt{T}$ is the empirical norm, $\lambda \geq 0$ is a tuning parameter, and Ω is the sg-LASSO regularizing functional:

$$\Omega(b) = \gamma|b|_1 + (1-\gamma)\|b\|_{2,1},$$

is a penalty function.[7] The sg-LASSO penalty is a linear combination of the LASSO (ℓ_1 norm), see Tibshirani (1996), and the group LASSO, see Yuan and Lin (2006) ($\|\beta\|_{2,1} = \sum_{k=1}^K |\beta_k|_2$). The group LASSO penalty selects covariates while the standard LASSO penalty selects the shape of the MIDAS weight function. The sparse-group LASSO estimator is an example of a shrinkage estimator, where the coefficients are shrunken towards zero in the norm Ω. The shrinkage reduces the variance at costs of introducing the bias and the appropriate choice of the tuning parameter λ allows us to achieve the desired combinations of the bias-variance trade-off.

The estimator nests the standard LASSO ($\gamma = 1$) and the group LASSO ($\gamma = 0$) as special cases. Babii, Ghysels, and Striaukas (2022) establish the non-asymptotic theoretical properties of the estimator for heavy-tailed τ-mixing processes which are

[6] Other possibilities include splines, trigonometric polynomials, or wavelets.

[7] We use $|z|_q = (\sum_{i=1}^p z_i^q)^{1/q}$ to denote the ℓ_q norm of $z \in \mathbf{R}^p$.

general enough for macroeconomic and financial time series.[8] The properties rely on the Fuk-Nagaev concentration inequality obtained in Babii, Ghysels, and Striaukas (2024).

The literature on the applications of penalized regressions to time series data is vast and we can only mention some of the interesting developments. Mogliani and Simoni (2021) propose a Bayesian approach to the high-dimensional MIDAS regressions based on the group LASSO. They find good forecasting performance in forecasting US economic activity. Beyhum and Striaukas (2023) extend the work of Babii, Ghysels, and Striaukas (2022) proposing a factor augmented sg-LASSO-MIDAS regression. They find that factor augmentation yields improvements in nowcast accuracy during the COVID period. Hecq, Ternes, and Wilms (2023) consider the extension of the sparse-group LASSO, called the hierarchical LASSO, where the groups can be arranged on a multi-level tree.

Some of the penalized methods have been used for a long time. For example, the HP filter is essentially a penalized regression; see Mei, Phillips, and Shi (2022) and Phillips and Shi (2021) for recent contributions. Chen and Maung (2023) propose a nonparametric estimator of time-varying forecast combination weights and develop corresponding asymptotic theory. They apply the LASSO-type estimator for kernel regression to estimate the forecast combination weights. Application to inflation and unemployment shows the benefits of the method compared to alternative techniques used in forecast combination literature such as Complete Subset Regressions proposed by Elliott, Gargano, and Timmermann (2013), partially egalitarian LASSO approach of Diebold and Shin (2019).

There are also a number of applications of penalized regressions to asset pricing; see Gu, Kelly, and Xiu (2020), Freyberger, Neuhierl, and Weber (2020), Feng, Giglio, and Xiu (2020), Bryzgalova (2015), and the review paper Giglio, Kelly, and Xiu (2022). Li, Plagborg-Møller, and Wolf (2022) conduct a comprehensive simulation study and conclude that the shrinkage and penalization can be attractive when estimating structural impulse response functions.

On the theory side, Kock (2016) and Medeiros and Mendes (2016, 2017) establish the model selection consistency and derive convergence rates for the adaptive LASSO with time series data. Kock and Callot (2015) and Masini, Medeiros, and Mendes (2022) derive convergence rates for high-dimensional VAR models; see also Wong, Li, and Tewari (2020), Chernozhukov, Härdle, Huang, and Wang (2021), Adamek, Smeekes, and Wilms (2023) for convergence rates of LASSO under β-mixing, physical

[8]This class of processes is large enough to cover the α-mixing processes as well as infinite linear transformations of β-mixing processes.

dependence, and near-epoch dependence.

2.2 Panel Data

It is often the case that the objective is to forecast or nowcast a large number of long time series of size T, e.g. N regional growth indices or price/earnings ratios for N firms observed at T quarters. In the latter case, the predictors cover the firm-specific accounting and textual data as well as the aggregate macroeconomic and financial indicators. While the time series methods described in the previous section can be applied series-by-series, this approach ignores the cross-sectional variation in the panel.

Babii, Ball, Ghysels, and Striaukas (2023, 2024) focus on the high-dimensional panel data regressions

$$y_{i,t+h} = \alpha_i + \sum_{k=1}^{K} \psi(L^{1/n_k^H}; \beta_k) x_{i,t,k} + u_{i,t|\tau}, \qquad i \in [N], t \in [T]$$

where the index i denotes the cross-sectional dimension, e.g. a region or a firm. The corresponding regularized fixed effects estimator solves

$$\min_{(a,b)\in\mathbf{R}^{N+p}} \|\mathbf{y} - Ba - \mathbf{X}b\|_{NT}^2 + 2\lambda\Omega(b),$$

where $B = I_N \otimes \iota$ and $\iota \in \mathbf{R}^T$ is an "all ones" vector and the panel data observations are stacked in $(\mathbf{y}, \mathbf{X})$. An attractive feature of the estimator is that it captures the heterogeneity of time series intercepts α_i. The disadvantage is that estimating N additional parameters leads to the precision loss.[9] In some cases, these costs outweigh the benefits and simpler pooled panel data regressions

$$\min_{(a,b)\in\mathbf{R}^{1+p}} \|\mathbf{y} - \iota a - \mathbf{X}b\|_{NT}^2 + 2\lambda\Omega(b),$$

where the intercepts are $\alpha_1 = \cdots = \alpha_N = a$.

Some of the recent empirical work using machine learning, panel data, and nowcasting includes Van Binsbergen, Han, and Lopez-Lira (2023) (firm earnings), Ghysels, Grigoris, and Özkan (2022) (government earnings and expenditures), Fosten and Greenaway-McGrevy (2022) (state-level GDP growth). On the methodology side, Carrasco and Rossi (2016) consider general regularization based on spectral

[9] Babii, Ball, Ghysels, and Striaukas (2023) quantify the price of estimating additional parameters depending on how persistent and fat-tailed the data are.

decomposition covering the ridge regression as a special case. Carvalho, Masini, and Medeiros (2018) apply the LASSO to predict controls for causal inference, generalizing the method of synthetic controls of Abadie, Diamond, and Hainmueller (2010).[10]

2.3 Nowcasting, real-time data flow, and textual data

The term nowcasting is a contraction of now and forecasting. It is defined as the prediction of the present, the very near future, or the very recent past, using the real-time data flow reflecting the evolving economic conditions and data revisions; see Bańbura, Giannone, Modugno, and Reichlin (2013) and Giannone, Reichlin, and Small (2008). Nowcasting a target variable y_t at low-frequency (e.g. quarterly) often involves vintage data, defined as a sequence of information sets, denoted

$$I_{t_r} = \left\{ x_{k,\lceil t_r \rceil - j/n_k^H | r} : \ k \in [K_r], j = \underline{j}_{r,k}, \ldots, n_k^H n_k^L - 1 \right\}$$

where $t_1 \leq t_2 \leq \cdots \leq t_R$ are times when the information set is updated.[11] The updates at a given time t_r appear for two reasons: 1) new data is *released*; 2) old data is *revised*. The revisions are especially common for the macroeconomic data and it is crucial to forecast using the vintage data available at a particular point of time to avoid the look ahead biases (see Ghysels, Horan, and Moench (2018) for further discussion).

Babii, Ghysels, and Striaukas (2022) consider the problem of nowcasting the quarterly US GDP growth using higher frequency macroeconomic and financial. They find that the machine learning nowcasts are either superior or at par with those posted by the New York Federal Reserve Bank. Additional gains are achieved using the data coming from the textual analysis of economic news; see Bybee, Kelly, Manela, and Xiu (2020). Ellingsen, Larsen, and Thorsrud (2022) also report that the textual news data add value to the traditionally used FRED-MD data. In a related work, Babii, Ball, Ghysels, and Striaukas (2023, 2024) consider the problem of nowcasting the price-earnings ratios with firm-specific accounting information as well as the aggregate macroeconomic, financial, and textual news information and report. They find that the machine learning panel data models perform favorably, cf. Ball and Ghysels (2018).

The high-dimensional MIDAS regression models are implemented in the package **midasml** available on CRAN. In the package, there are multiple functionalities,

[10] The literature on using LASSO to predict the counterfactuals is fast; see Belloni, Chernozhukov, and Hansen (2014), Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins (2018), and references therein.

[11] For a real number a, we use $\lceil a \rceil$ to denote the smallest integer larger than a.

including functions that deal with the high frequency lags construction, estimation of time series and panel data regression models, tuning parameter selection methods such as (time-series) cross-validation and information criteria, and precision matrix estimation used to compute the debiased estimates for conducting Granger causality test (see the subsequent section).

Borup, Rapach, and Schütte (2023) study the weekly unemployment insurance initial claims using unrestricted MIDAS specification utilizing Google trends data; see also Ferrara and Simoni (2022). They find that the ensemble (or combinations) of linear and nonlinear ML methods perform the best and that the daily Google Trends data were particularly relevant during the COVID-19 crisis. Jardet and Meunier (2022) also find that nowcasting performance improves during crises period. On the other hand, Jardet and Meunier (2022) find that nowcasting performance improves during crises period when weekly data is used in prediction models.

Barbaglia, Manzan, and Tosetti (2023) develop a Fine-Grained Aspect-based Sentiment analysis method to compute sentiments from news articles about the state of the economy. They find that economic news sentiments extracted from a large pool of news articles track economic cycles and help accurately nowcast economic activity. Lastly, Cimadomo, Giannone, Lenza, Monti, and Sokol (2022) consider large Bayesian Vector Autoregressive (BVAR) models to nowcast the US economic activity.

2.4 Granger Causality Tests

The time series models are often misspecified. In this case, the regression has only a projection interpretation and the regression errors are serially correlated. In addition to that the LASSO estimator has a complicated sampling distribution due to a significant shrinkage bias. Let $\hat{\beta}_G = (\hat{\beta}_j)_{j\in G}$ be a subset of projection coefficients fitted with the LASSO (or sg-LASSO) indexed by $G \subset [p]$, where p is the number of regressors. Babii, Ghysels, and Striaukas (2024) show that for the heavy-tailed τ-mixing time series, we have

$$\sqrt{T}(\hat{\beta}_G + B_G - \beta_G) \xrightarrow{d} N(0, \Xi_G),$$

where B_G is a bias correction term, $\Xi_G = \lim_{T\to\infty} \mathrm{Var}\left(\frac{1}{\sqrt{T}}\sum_{t=1}^{T} u_t \Theta_G x_t\right)$ is the long-run variance, and Θ is the precision matrix.[12]

[12]See also Chernozhukov, Härdle, Huang, and Wang (2021) for the physically dependent processes and Adamek, Smeekes, and Wilms (2023) for the near-epoch dependent processes.

The long-run variance can be estimated using the standard HAC estimator, see Newey and West (1987) and Andrews (1991)

$$\hat{\Xi}_G \triangleq \sum_{|k|<T} K\left(\frac{k}{M_T}\right)\hat{\Gamma}_k,$$

where $K : \mathbf{R} \to [-1, 1]$ is the kernel weight function, M_T is the lag truncation parameter, and $\hat{\Gamma}_k = \hat{\Theta}_G\left(\frac{1}{T}\sum_{t=1}^{T-k}\hat{u}_t\hat{u}_{t+k}x_t x_{t+k}^\top\right)\hat{\Theta}_G^\top$, are the autocovariances for fitted residuals $\hat{u}_t$. Babii, Ghysels, and Striaukas (2024) characterize the MSE convergence rate of the HAC estimator based on the sg-LASSO residuals. Their result leads to the following "rule of thumb" choice of the bandwidth parameter

$$M_T = \begin{cases} 1.3\left(\frac{T}{\log p}\right)^{\frac{1}{1+\varsigma}}, & \text{sub-Gaussian data} \\ 1.3\left(\frac{T^{2-2/q}}{p^{2/q}}\right)^{\frac{1}{1+\varsigma}}, & \text{heavy-tailed data,} \end{cases}$$

where $\varsigma = 2$ for the Quadratic spectral and Parzen kernels, and $q > 2$ is the number of finite moments in the data.

For forecasting problems, we can use these results to test Granger causality which is a formal statistical way to evaluate whether a particular time series marginally adds to the projection of a target variable on a set of predictors. Interestingly, in his original paper, Granger (1969a) defined causality in terms of high-dimensional time series data which he referred to as "all the information available in the universe at time t". To test whether a series $(w_t)_{t\in\mathbf{Z}}$ Granger causes another series $(y_t)_{t\in\mathbf{Z}}$ at a horizon h, consider

$$y_{t+h} = c + \sum_{j\geq 1} z_{t,j}\gamma_j + \mathbf{w}_{t-1}^\top\alpha + u_{t+h},$$

where $(z_{t,j})_{j\geq 1}$ is a high-dimensional set of controls $\alpha \in \mathbf{R}^K$ and $\mathbf{w}_{t-1} \in \mathbf{R}^K$ is a vector lags of $(w_t)_{t\in\mathbf{Z}}$.

Babii, Ghysels, and Striaukas (2024) show that under the null hypothesis, the bias-corrected Wald statistics follows a chi-squared distribution

$$W_T := T\left[(\hat{\alpha} + A - \alpha)\right]^\top \hat{\Xi}_\alpha^{-1}\left[(\hat{\alpha} + A - \alpha)\right] \xrightarrow{d} \chi^2_K,$$

where A is the bias correction term and $\hat{\Xi}_\alpha$ is the HAC estimator. The practical implementation of the test is as follows:

1. Estimate $\alpha \in \mathbf{R}^K$ using the LASSO (or sg-LASSO) and compute the HAC estimator using the LASSO residuals.

2. Compute the bias-corrected Wald statistics.

3. Reject the Granger non-causality if $W_T > q_{1-\alpha}$, where $q_{1-\alpha}$ is the $1-\alpha$ quantile of χ^2_K and do not reject otherwise.

Alternatively, one could use a likelihood ratio test or an LM test; see Hecq, Margaritella, and Smeekes (2023) for the latter.

As an illustrative example, we provide a stylized Monte Carlo example simulating a model

$$y_{t+1} = z_t\gamma_0 + \mathbf{w}_{t-1}^\top\alpha_0 + u_{t+1},$$

where $\gamma_0 \in \mathbf{R}$ and $\alpha_0 \in \mathbf{R}^p$. The time series are generated as $(z_t, \mathbf{w}_{t-1}^\top) \sim AR(1)$ and $u_{t+1} \sim AR(1)$ with AR coefficient set to 0.6 in all cases. We use a Toeplitz covariance with a coefficient 0.6 for the covariance matrix of $(z_t, \mathbf{w}_{t-1}^\top)$ to introduce the dependence among covariates. The regression coefficients are $\gamma_0 = 1$ and the first four entries of α_0 are equal to 1 while the remaining entries are zero. We plot the squared ratio of bias/standard error, where the standard errors are computed with the HAC estimator as in Babii, Ghysels, and Striaukas (2024). We report the values for the γ_0 where i) estimated using debiased LASSO regressing y_{t+1} on z_t and $\mathbf{w}_t$ (LASSO) ii) $\hat{\gamma}$ is estimated with OLS by regressing y_{t+1} on z_t (OLS-low) and iii) $\hat{\gamma}$ is estimated with OLS by regressing y_{t+1} on z_t and $\mathbf{w}_t$ (OLS-high). We increase the dimension $p = \{20, 40, \ldots, 100\}$. Results are plotted in Figure 1.

The plot reveals that the LASSO performs well regardless of the number of controls. On the one hand, the OLS without controls has a larger bias compared to the LASSO due to the exclusion of relevant controls. Including all controls is only feasible for low to moderate dimensions of the control vector. For a larger number of controls the errors grow dramatically.

2.5 Time Series Cross-Validation

The practical implementation of ML methods requires specifying one or several tuning parameters. For i.i.d. data, a common practice is to rely on the K-fold cross-validation. which may or may not be appropriate for time series data. Bergmeir, Hyndman, and Koo (2018) show that for autoregressive models with i.i.d. errors the standard K-fold cross-validation remains valid. However, with correlated errors – for example, when the regression has only projection interpretation due to misspecification – the standard cross-validation fails due to the correlation between the training and the test samples.

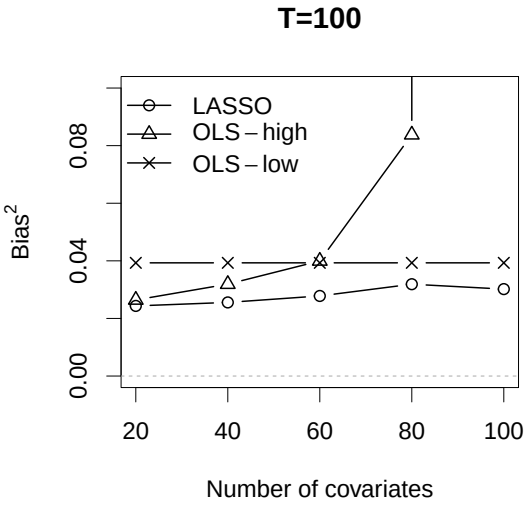


Figure 1: The squared ratio of bias/standard error for LASSO, OLS without high-dimensional controls (OLS-low) and OLS with high-dimensional controls (OLS-high).

One could rely on the following leave-one-out cross-validation with a gap procedure that decorrelates the training and the test samples, see also Chu and Marron (1991). Let $\hat{f}_\lambda(x_t)$ be a prediction rule of a machine learning model with tuning parameters $\lambda = (\lambda_1, \ldots, \lambda_M)$.[13]. For some $l \in \mathbf{N}$ and each $t = 1, \ldots, T$:

1. If $t > l+1$ and $t < T-l$, use observations $I_{t,l} = \{1, \ldots, t-l-1, t+l+1, \ldots, T\}$ to fit the machine prediction, denoted $\hat{f}_{\lambda,-t,l}(x_t)$. For $t = 1, \ldots, l+1$, use $I_{t,l} = \{t+l+1, \ldots, T\}$ as the training sample. Similarly, for $t = T-l, \ldots, T$, use $I_{t,l} = \{1, \ldots, T-l-1\}$ as the training sample.

2. Use the left-out observations to test the model

$$CV(\lambda) = \frac{1}{T}\sum_{t=1}^{T} \ell(y_t - \hat{f}_{\lambda,-t,l}(x_t)),$$

   where ℓ is the loss function, e.g. the MSE or quadratic loss, $\ell(u) = u^2$.

3. Minimize $CV(\lambda)$ with respect to λ.

[13]For example, sg-LASSO corresponds to the linear prediction rule $\hat{f}_\lambda(x_t) = x_t^\top \hat{\beta}_\lambda$ with $M = 2$

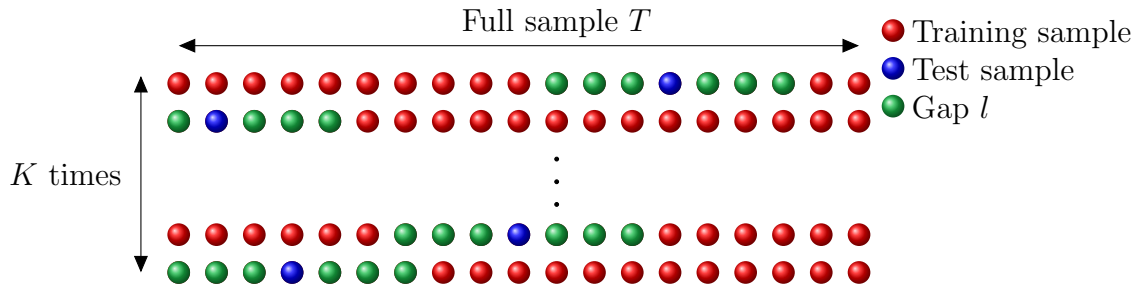


Figure 2: Time series cross-validation scheme with $l = 3$

For $l = 0$ the procedure is the usual leave-one-out cross-validation while for $l \geq 1$, there is a gap of l observations separating the test and the training samples. Since the procedure is computationally demanding, Babii, Ghysels, and Striaukas (2024) draw randomly a sub-sample $I \subset [T]$ of size K and minimize

$$CV_K(\lambda) = \frac{1}{K} \sum_{t \in I} \ell(y_t - \hat{f}_{\lambda,-t,l}(x_t))$$

instead. Figure 2 illustrates it for a gap of $l = 3$ observations. The red training data are separated from the blue test data with a gap of 3 green left-out observations on each side.

3 Nonlinear Machine Learning Methods

The regularized linear projection methods discussed in the previous section can be viewed as linear approximations, $x_t^\top \beta_h$, to the potentially nonlinear conditional mean function $f_h^*(x_t) = \mathbb{E}[y_{t+h}|x_t]$, or more generally to the optimal decision rule f_h^* for the loss function $\ell(y_{t+h}, f_h(x_t))$. Nonlinear machine learning methods, such as regression trees, random forests, boosting, and (deep) neural networks, aim to achieve more accurate approximations to f_h^* capturing nonlinearities and higher-order interactions between covariates.

It is essential to note that the distinction between linear and nonlinear methods is not rigid, as one can always expand the predictor space with quadratic, interaction terms, and higher-order counterparts to apply techniques like LASSO or other shrinkage methods.[14] From a practical standpoint, the flexibility of nonlinear

[14]This is justified since functions have potentially infinite series expansions in various bases, such as orthogonal polynomials, splines, or wavelets.

methods comes at the cost of fitting a larger number of parameters, which can be challenging in low signal-to-noise environments where evidence for nonlinearities is often weak. In this section, we review the tree-based methods and (deep) neural networks. Since the time series lags and the cross-validation can be used in the same way as for linear methods, in what follows we will skip these topics.

3.1 Tree-based Methods

3.1.1 Regression and Classification Trees

Regression and classification trees are sequential methods for building forecasting models. The predictor space $\mathcal{X} \subset \mathbf{R}^p$ is greedily decomposed into a partition $\mathcal{X} = R_1 \cup R_2 \cup \cdots \cup R_J$ with $R_j \cap R_k = \emptyset, \forall j \neq k$. The fitted regression function is piecewise constant on the partition elements:

$$\hat{f}(x) = \sum_{j=1}^{J} \hat{c}_j \mathbb{1}_{R_j}(x),$$

where $\hat{c}_j$ is the sample mean of all y_{t+h} such that $x_t \in R_j$ and we put $\mathbb{1}_{R_j}(x) = 1$ if $x \in R_j$ and $\mathbb{1}_{R_j}(x) = 0$ otherwise.[15] The partition usually consists of hyperrectangles with sides parallel to coordinate axes. The process unfolds sequentially by selecting a predictor $k = 1, \ldots, p$ and a split location $s \in \mathbf{R}$ that yields the largest reduction in mean-squared error. It is easiest to understand the process graphically.

Consider a very simple example of predicting inflation with an unemployment rate. Figure 3, panel (a) shows a tree with five partition elements (called leaves) and four splits at various levels of the unemployment rate. The partition is obtained by splitting the space of unemployment at 3.65 first, then the interval with values ≥ 3.65 is splitted at 9.35 and the process continues with 2 more splits. The number at the bottom are the predicted values obtained as sample means of inflation in each leave. Panel (a) displays the corresponding piecewise constant regression function.

The regression tree might not be the most captivating tool when only one predictor is available; more visually appealing estimates can be derived from nonparametric smoothing techniques. However, its greedy nature and ease of visualization become advantageous with larger predictor sets. Each split location is determined by the most influential predictor, yielding the greatest model fit improvement. Figure 3, panel (b), displays a tree with splits along two predictors (unemployment rate and

[15] Piecewise linear or polynomial functions can also be considered; see Friedberg, Tibshirani, Athey, and Wager (2020).

industrial production), and panel (d) displays the resulting partition with predicted inflation values. It is generally recommended to grow deeper trees with more splits and then prune them using an appropriate cross-validation procedure.

Classification trees share this methodology, with the distinction that the outcome is a discrete variable. These trees employ the so-called Gini or cross-entropy measures instead of mean-squared error to guide the splitting process.

The key tuning parameter for regression and classification trees is the depth of the tree or the optimal level of pruning, reflecting the total number of leaves. Deeper trees fit the data with lower bias and higher variance, while shallow trees have more observations in each leave, reducing the variance at the cost of potentially missing important nonlinearities.

3.1.2 Random Forests

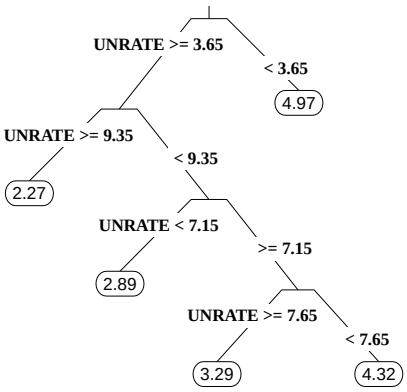
While regression trees are interpretable, a single tree might not be the most powerful predictive model. Random forests enhance the performance by bootstrapping the original data B times and fitting a regression tree, denoted as $\hat{f}_b$, on each bootstrap sample. The final regression estimate is obtained through the sample average:

$$\hat{f}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x). \tag{1}$$

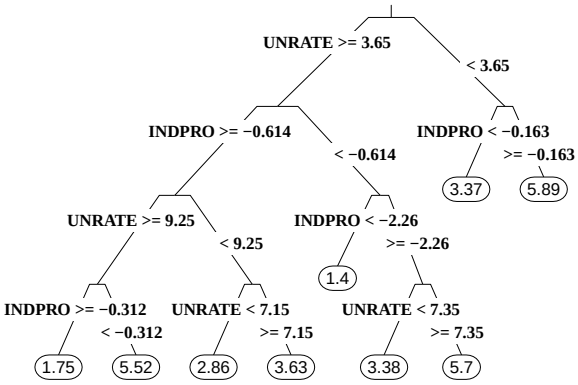
This technique, known as bagging (**b**ootstrap **agg**regation), aims to reduce the variance associated with a single tree. A formal analysis can be found in Friedman and Hall (2007), with an early economic application discussed in Inoue and Kilian (2008). It is essential to exercise caution when bootstrapping time series data, as naive sampling with replacements can disrupt time-series dependence. One remedy is to sample data blocks $(y_{t+h}, x_t)_{t=1}^{T}$; see Politis and Romano (1994) for details.

Random forests can be conceptualized as a form of bagging, where, at each split, only a subset of randomly selected m predictors is considered; see Breiman (2001a). A common rule-of-thumb is to set m as the integer part of $\sqrt{p}$. The rationale behind this choice lies in the lower variance of the sample mean in equation (1) compared to the variance of a single tree $\hat{f}_b(x)$, provided that the trees are not correlated. However, if there is a dominant predictor, all bagged trees may make their first split along this predictor. Randomly selecting a subset of predictors for each split helps decorrelate the trees.

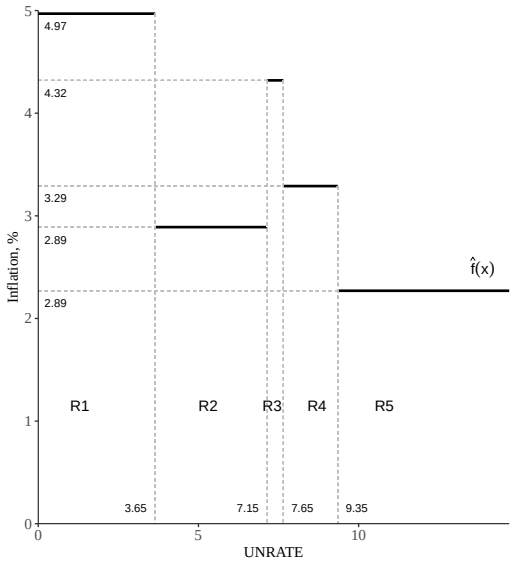
The asymptotic theory of regression trees and random forests is not as well-established as that for nonparametric kernel or series estimators, even in the i.i.d.



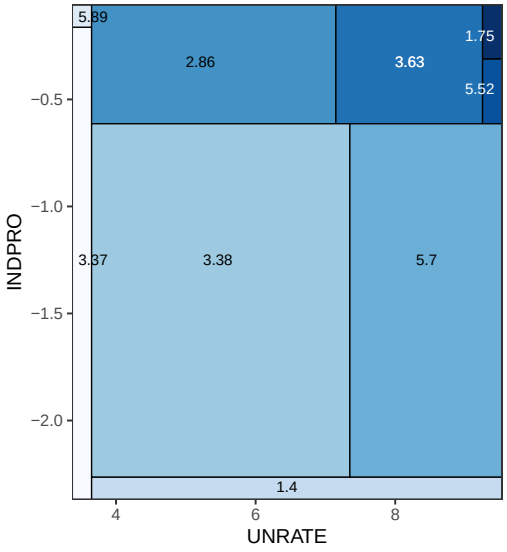
(a) Unemployment.



(b) Unemployment and industrial production.



(c) Regression corresponding to (a).



(d) Partition corresponding to (b).

Figure 3: Fitted regression trees for the year-on-year inflation using unemployment rate as a predictor (Figure 3a-3c) and unemployment rate with industrial production as predictor (Figure 3b-3d). Sample period January, 1980, to November, 2023.

case. Key contributions are found in Cattaneo, Chandak, and Klusowski (2022), Chi, Vossler, Fan, and Lv (2022), Syrgkanis and Zampetakis (2020), Athey and Imbens (2019), Scornet, Biau, and Vert (2015), and Nobel (1996).

3.1.3 Boosting

Boosting progressively constructs a forecasting model by iteratively refining the fit using shallow regression trees to refit residuals. In the initial step, a shallow tree, denoted as $\hat{f}_1$, is obtained to fit (y_{t+h}, x_t), updating residuals as $\hat{u}_{t+h}^{(1)} = y_{t+h} - \lambda \hat{f}_1(x_t)$ for some $\lambda > 0$. For subsequent iterations $k = 2, 3, \ldots, K$, the next tree, $\hat{f}_k$, is fitted using $(\hat{u}_{t+h}^{(k-1)}, x_t)$, with residuals updated as $\hat{u}_{t+h}^{(k)} = \hat{u}_{t+h}^{(k-1)} - \lambda \hat{f}_k(x_t)$. The regression estimate is then given by:

$$\hat{f}(x) = \sum_{k=1}^{K} \lambda \hat{f}_k(x)$$

The critical aspect of this process is the early stopping at K which is pivotal for managing the bias-variance trade-off. Halting too soon results in substantial bias, while overly large choices of K lead to heightened variance. Other key tuning parameters include the tree depth and the learning rate λ. Smaller values of λ facilitate slower learning at the expense of requiring larger iterations K to achieve a good fit. **XGBoost**, a popular Python and R implementation of boosting, introduces several additional regularizations and tuning parameters. Proper selection of these parameters is crucial, and the cross-validation with appropriate time series adjustments can be used as discussed in the previous section.

It's worth noting that a common practice in time series analysis involves plotting/regressing residuals against various covariates. Boosting automates this process by sequentially extracting small pieces of predictive information from residuals. Additionally, boosting can be viewed as a form of functional gradient descent with an early stopping rule, known for its regularization effect; see Biau and Cadre (2021), Blanchard, Lugosi, and Vayatis (2003), and Friedman (2001). While kernel and series estimators excel on standard smoothness spaces and are impossible to outperform, see Stone (1982), the exceptional performance of tree-based methods in practice remains not entirely understood. This efficacy is likely attributable to their adaptability to sparsity and heterogeneous smoothness, aspects conventional implementations of linear nonparametric methods like kernels or splines may struggle to achieve.

To highlight some of the applications of tree-based methods we refer to Bryzgalova, Pelger, and Zhu (2024), Gu, Kelly, and Xiu (2020), and Rossi and Timmermann (2015) who use tree-based methods in asset pricing. Medeiros, Vasconcelos, Veiga, and Zilberman (2021), Coulombe (2024), Lahiri and Yang (2022), and Bai and Ng (2009) use random forests and boosting for macroeconomic forecasting. Lastly, Babii, Chen, Ghysels, and Kumar (2021) and Kleinberg, Lakkaraju, Leskovec, Lud-

wig, and Mullainathan (2018) focus on predicting recidivism at the stage of pre-trial detention.

3.2 Neural Networks and Deep Learning

Neural network regressions can be conceptualized as M-estimators, solving a minimization problem over a sequence of approximating spaces $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \ldots$

$$\min_{f \in \mathcal{F}_k} \sum_{t=1}^{T} \ell(y_{t+h}, f(x_t)) + \lambda \Omega(f), \tag{2}$$

where ℓ represents a quadratic (or alternative) loss function and Ω is a regularizing functional; see Chen (2007) for an early review. A single-layer neural network is characterized as

$$\mathcal{F}_k = \left\{ x \mapsto \sum_{j=1}^{p_1} w_{1,j} \sigma(w_{0,j}^\top x + v_{1,j}) : \ \theta = (v_{1,1}, w_{0,1}^\top, w_{1,1} \ldots, v_{1,p_1}, w_{0,p_1}^\top, w_{1,p_1}^\top)^\top \right\}, \tag{3}$$

where σ is a known nonlinear activation function and $\theta \in \mathbf{R}^d$. Popular choices include the sigmoid function, $\sigma(u) = 1/(1+e^{-u})$, or the rectified linear unit (ReLU), $\sigma(u) = \max(u, 0)$. Solving the problem in equation (2) effectively minimizes over $\theta \in \mathbf{R}^d$, which is known as a nonlinear least-squares in the case of quadratic loss. The width of the network p_1 serves as a crucial tuning parameter, controlling the bias-variance trade-off. A sufficiently wide single-layer neural network with large p_1 can approximate a continuous function akin to algebraic polynomials or splines; see Hornik, Stinchcombe, and White (1990) or Mhaskar (1996).

Recent breakthroughs in computer vision and natural language processing have propelled neural networks into the spotlight. Networks applied to text, images, speech, and videos often feature hundreds of layers obtained recursively, a methodology commonly referred to as deep learning. To describe a multilayer neural network, note first that a single layer neural network in equation (2) can be shortly denoted as $x \mapsto W_1 \sigma_{v_1} \circ W_0 x$, where W_j are $p_{j+1} \times p_j$ matrices and $\sigma_v \circ (y_1, \ldots, y_p)^\top = (\sigma(y_1 - v_1), \ldots, \sigma(y_p - v_p))^\top$ with $v \in \mathbf{R}^p$. A neural network with L layers can be described recursively as

$$\mathcal{F}_k = \left\{ x \mapsto W_L \sigma_{v_L} \circ W_{L-1} \sigma_{v_{L-1}} \ldots W_1 \sigma_{v_1} \circ W_0 x : \ \theta = (W_0, W_1, \ldots, W_L) \right\},$$

where the unknown parameters are the weight matrices $(W_j)_{j=0}^L$; see Figure 4 for an example.

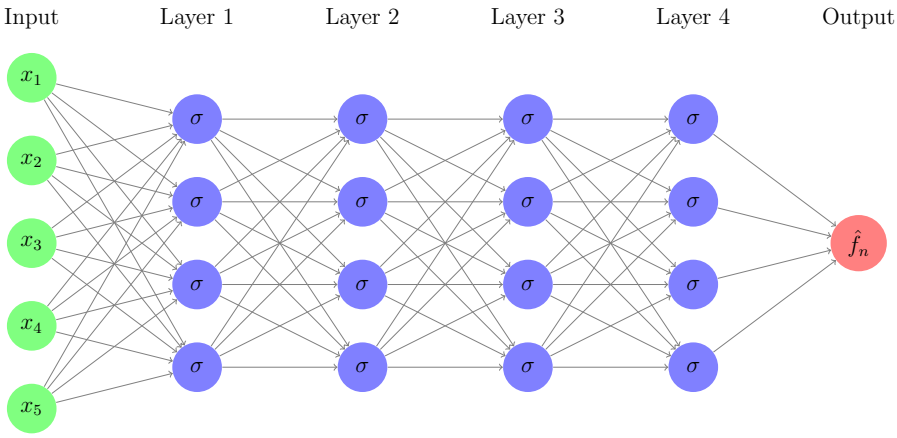


Figure 4: Directed graph of a deep neural network with $p_0 = 5$ covariates, and $L = 4$ hidden layers of width 4 neuron.

It is noteworthy that for tabular economic and financial data, the optimal performance of neural networks typically occurs with 1-5 layers. To the best of our knowledge, there is no evidence supporting the idea that very deep neural networks outperform simpler, shallow multi-layer networks. Neural networks can be employed, for example, via the **TensorFlow** Python library. Beyond considerations of depth, width, and activation function, the practical implementation involves various tuning parameters. These include the number of epochs (the times gradient descent attempts to solve the non-convex optimization problem), the size of the subsample used to compute the gradient, the gradient step size, among others. It is crucial to carefully select these tuning parameters, and randomness should be controlled using multiple seed numbers to ensure reproducibility.

On the theoretical side, while the approximating properties of deep neural networks as $L \to \infty$ resemble those of shallow networks, they sometimes exhibit better expressive power; see Yarotsky (2017). Furthermore, neural networks can adapt to special structures, such as linear or additive structures and manifolds; see Bach (2017), Bartlett, Montanari, and Rakhlin (2021), and Berner, Grohs, Kutyniok, and Petersen (2021) for recent mathematical reviews. For applications in economics, refer to Xu, Wang, Jiang, and Liu (2023), Bredahl Kock and Teräsvirta (2016), and Gu, Kelly, and Xiu (2020). Numerous fascinating topics are associated with neural networks, including sparsity, benign overfitting (or the double descent curve), and convolutional and recurrent neural networks. For an in-depth exploration of these topics, we recommend the aforementioned review papers.

4 Classification for Economists

Forecasting binary variables is a prominent problem, also known as the classification or screening in the computer science literature; see Lahiri and Yang (2013) for a review. The classification rules build a foundation for the automated data-driven algorithm based on vast data inputs that are increasingly used for various life-changing decisions, including job hiring, pre-trial release from jail, medical testing and treatment. They are also used for various routine tasks such as loan approval, fraud detection, or spam filtering.

Babii, Chen, Ghysels, and Kumar (2021) highlight that the downside risk and upside gains of many economic decisions are not symmetric. The importance of asymmetries in prediction problems arising in economics has been recognized for a long time; see Granger (1969b), Manski and Thompson (1989), Granger and Pesaran (2000), Elliott and Lieli (2013), and the textbook treatment in Elliott and Timmermann (2016), among many others. At the same time, the standard logistic regression and machine learning algorithm often ignore the asymmetric cost and benefit considerations. Consider, for example, the problem of forecasting recession. The prediction is $f(x_t) \in \{-1, 1\}$ (1 if recession) and outcome is $y_t \in \{1, -1\}$ (1 if recession).

prediction \ outcome	Recession, $y_t = 1$	No Recession, $y_t = -1$
Recession, $f(x_t) = 1$	$\ell_{1,1}(z_t)$	$\ell_{1,-1}(z_t)$
No Recession, $f(x_t) = -1$	$\ell_{-1,1}(z_t)$	$\ell_{-1,-1}(z_t)$

Table 1: Classification under asymmetric losses

If the recession is falsely predicted (a false positive mistake), we suffer a loss $\ell_{1,-1}$ while if we fail to predict the recession (a false negative mistake), we suffer a loss $\ell_{-1,1}$. The decision maker may have preferences such that failing to predict a recession is costlier than the false alarm of a recession, in which case $\ell_{-1,1} > \ell_{1,-1}$. Additionally, there may be some benefits for correct predictions encoded in $\ell_{1,1} \leq 0$ and $\ell_{-1,-1} \leq 0$.[16] Note also that the quartet of loss function in Table 1 may be driven by some economic factors encoded in z_t.[17] Babii, Chen, Ghysels, and Kumar (2021) show that the economic costs and benefits can be accommodated by reweighing the

[16] Another example of an asymmetric decision problem could be the case of a policymaker deciding between two policies under good or bad economic conditions or a risk-loving investor deciding between going short or long on an asset.

[17] Compare this with the standard binary classification (e.g. logistic regression) which optimizes the following loss function $\ell(f, y, x) = \mathbb{1}\{f(x) \neq y\}$.

logistic regression (or ML methods) by the asymmetries of the loss function. For instance, in the case of the logistic regression with a LASSO penalty, it is enough to solve

$$\hat{\theta} = \underset{\theta \in \mathbf{R}^p}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \omega(y_i, x_i) \log\left(1 + e^{-y_i x_i^\top \theta}\right) + \lambda |\theta|_1, \tag{4}$$

where the individual likelihoods are weighted by $\omega(y_i, x_i) := y_i a(x_i) + b(x_i)$ with

$$\begin{aligned} a(x) &= \ell_{-1,1}(x) - \ell_{1,1}(x) + \ell_{-1,-1}(x) - \ell_{1,-1}(x), \\ b(x) &= \ell_{-1,1}(x) - \ell_{1,1}(x) + \ell_{1,-1}(x) - \ell_{-1,-1}(x). \end{aligned}$$

The data decision rule is then $\hat{f}(x_i) = 1$ if $x_i^\top \hat{\theta} \geq 0$ and $\hat{f}(x_i) = -1$ if $x_i^\top \hat{\theta} < 0$. Note that the problem in Eq. (4) is a convex optimization problem that can be easily solved using the standard optimization methods. This bypasses the need to solve a non-convex problem using the mixed-integer optimization; see Elliott and Lieli (2013) and Florios and Skouras (2008).[18] In addition to the (high-dimensional) logistic regression, the approach of Babii, Chen, Ghysels, and Kumar (2021) can also be applied to suitably reweighted support vector machines (SVM), boosting, and deep learning.[19]

Some recent methodological developments and applications related to classification include Barbaglia, Manzan, and Tosetti (2023), Kitagawa, Sakaguchi, and Tetenov (2021), and Christensen, Moon, and Schorfheide (2020), while the fairness issues are also discussed in Rambachan, Kleinberg, Mullainathan, and Ludwig (2020) and Viviano and Bradic (2023).

5 Tensor Factor Models

The datasets available in modern empirical applications often have a multi-dimensional panel structure. For example, in the regional macroeconomic datasets, $y_{i,j,t}$ is the macroeconomic indicator i for region j measured at time t, so the data is the 3-dimensional panel. Another example is the network data, where $y_{i,j,t}$ is the outcome for the nodes (i, j) at time t, e.g. the exchange rates for a pair of currencies (i, j). In asset pricing, $y_{i,j,t}$ is the excess return of j^{th} quantile sorted on anomaly i at time

[18] See also Pellatt and Sun (2013) for a PAC-Bayesian perspective.

[19] Most of the popular machine learning packages classification packages in Python and R conveniently allow to specify weights including the **XGBoost**, **scikit-learn**, and **TensorFLow** Python libraries.

t; see Lettau and Pelger (2020). Adding the international dimension, we obtain the 4-dimensional panel.

While the two-dimensional panel data are represented by matrices, the multi-dimensional panel data lead to their higher-order counterparts, called tensors. A d-dimensional tensor is described by enumerating all the entries along the d dimensions: $\mathbf{Y} = \{y_{i_1,i_2,\ldots,i_d},\ 1 \le i_j \le N_j,\ 1 \le j \le d\}$; see Figure 5.

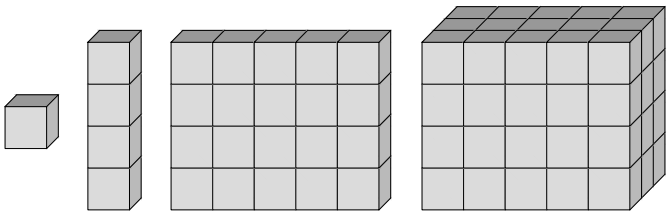


Figure 5: A scalar, 1st order, 2nd order, and 3rd order tensors

Tensor datasets are often characterized by complex dependencies between entries ignoring which may not be appropriate. For instance, the regional macroeconomic indicators are correlated with each other, as well as over space and time. Babii, Ghysels, and Pan (2023) consider the tensor factor model to capture such dependencies. For a tensor $\mathbf{Y} \in \mathbf{R}^{N\times J\times T}$, the model with R latent factors is[20]

$$\mathbf{Y} = \sum_{r=1}^{R} \lambda_r \otimes \mu_r \otimes f_r + \mathbf{U}, \qquad \mathbb{E}\mathbf{U} = 0,$$

where $\mathbf{U} \in \mathbf{R}^{N\times J\times T}$ is an idiosyncratic noise tensor, $f_r \in \mathbf{R}^T$ are time series factors, $\lambda_r \in \mathbf{R}^N$ and $\mu_r \in \mathbf{R}^J$ are loadings in different dimensions, and we use the tensor product notation so that $(\lambda_r \otimes \mu_r \otimes f_r)_{i,j,t} = \lambda_{r,i}\mu_{r,j}f_{r,t}$. This means that the entry (i,j,t) in $\mathbf{Y}$ is modeled as

$$y_{i,j,t} = \sum_{r=1}^{R} \lambda_{r,i}\mu_{r,j}f_{r,t} + u_{i,j,t}.$$

[20]Note that while one could consider a three-dimensional tensor as a collection of matrices and apply the standard factor model, this approach has significant limitations. It ignores the tensor structure and leads to the overparametrized model compared to the tensor factor approach.

In the asset pricing application described above, the loading $\lambda_{r,i}$ would be the exposure of an anomaly i to the common factor $f_{t,r}$ while the loading $\mu_{r,j}$ would be the exposure of j^{th} quantile to $f_{t,r}$.

The conventional approach is to ignore the tensor factor structure fitting a model with pooled loadings

$$y_{i,j,t} = \sum_{r=1}^{R} \nu_{r,i,j} f_{r,t} + u_{i,j,t}.$$

Babii, Ghysels, and Pan (2023) discuss that such an approach is not optimal as it involves $(NJ+T)\times R$ parameters instead of $(N+J+T)R$ parameters in the tensor factor approach.

More generally, for a d-dimensional tensor $\mathbf{Y} \in \mathbf{R}^{N_1\times\cdots\times N_d}$, the tensor factor model is

$$\mathbf{Y} = \sum_{r=1}^{R} \sigma_r \bigotimes_{j=1}^{d} m_{j,r} + \mathbf{U}, \qquad \mathbb{E}\mathbf{U} = 0,$$

where $m_{j,r}$ are the unit norm loadings/factors and σ_r are the scale components.

Babii, Ghysels, and Pan (2023) study the PCA estimators for the tensor factor model based on tensor matricizations along each of its dimensions. The PCA estimators have a closed-form expression in contrast to the conventionally used alternating least-squares algorithm for tensor decomposition; see Kolda and Bader (2009). To describe the algorithm, we need to matricize tensors which is an operation generalizing the matrix vectorization:

Example 5.1. *Let* $\mathbf{Y}$ *be a* $3\times 4\times 2$ *dimensional tensor of the following two slices:*

$$\mathbf{Y}_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \quad \mathbf{Y}_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}.$$

*Then the mode-*1*,* 2 *and* 3 *matricizations of* $\mathbf{Y}$ *are respectively:*

$$\mathbf{Y}_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}, \qquad \mathbf{Y}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix},$$

$$\mathbf{Y}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & \cdots & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & \cdots & 21 & 22 & 23 & 24 \end{bmatrix}$$

This leads to the following tensor PCA algorithm:

1. Matricize the tensor $\mathbf{Y}$ into matrices $\mathbf{Y}_{(1)}, \mathbf{Y}_{(2)}, \ldots, \mathbf{Y}_{(d)}$ along each of its dimensions.

2. Estimate the unit norm factors and loadings as $(\hat{m}_{j,1}, \ldots, \hat{m}_{j,R})$ via PCA in each of the d dimensions, i.e. the first R eigenvectors of $N_j \times N_j$ matrix $\mathbf{Y}_{(j)}\mathbf{Y}_{(j)}^\top$.

3. Estimate $(\hat{\sigma}^2_{r,j})_{r=1}^R$ as the R largest eigenvalues of $\mathbf{Y}_{(j)}\mathbf{Y}_{(j)}^\top$.

To determine the number of factors in a tensor factor model, Babii, Ghysels, and Pan (2023) consider the eigenvalue ratio test, cf. Onatski (2009). They show that the null hypothesis that there are at most k factors, we have for every matricization $j = 1, 2, \ldots, d$,

$$S_j := \max_{k<r\leq K} \frac{\hat{\sigma}^2_{r,j} - \hat{\sigma}^2_{r+1,j}}{\hat{\sigma}^2_{r+1,j} - \hat{\sigma}^2_{r+2,j}} \xrightarrow{d} \max_{0<r\leq K-k} \frac{\xi_r - \xi_{r+1}}{\xi_{r+1} - \xi_{r+2}} =: Z,$$

where $(\xi_1, \ldots, \xi_{K-k+2})$ follow the joint type-1 Tracy-Widom distribution; see Karoui (2003). On the other hand, under the alternative hypothesis that the number of factors is $> k$ but $\leq K$, the statistics diverges to infinity. The testing procedure is

1. Let $(Z_i)_{i=1}^m$ be m independent random variables drawn from the same distribution as Z. To approximate the distribution of $(\xi_1, \xi_2, \ldots)$, we can use the first eigenvalues of a symmetric $N_j \times N_j$ Gaussian matrix matrix with entries $\zeta_{i,j} \sim_{\text{i.i.d.}} N(0, \tau_{i,j}), i \leq j$, where $\tau_{i,j} = 1$ if $i < j$ and $\tau_{i,j} = 2$ if $i = j$.

2. Compute the p-value $p_j = 1 - F_m(S_j)$ for each $1 \leq j \leq d$, where $F_m(x) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}_{Z_i \leq z}$.

3. Combine the p-values from the individual matricizations as $p_{\text{mean}} = \frac{2}{d}\sum_{j=1}^d p_j$; see Vovk and Wang (2020).

The tensor PCA algorithm and the test can be computed using the **TensorPCA** package.[21]

There also exist tensor factor models related to the Tucker decomposition that allows for a different number of factors in different dimensions, see Chen, Yang, and Zhang (2022) and Han, Chen, and Zhang (2022). While these models are more general they feature a larger number of parameters and may involve non-trivial identification issues. To the best of our knowledge, there are no formal statistical tests that can be used to decide which model is appropriate for a particular application.

[21] See `https://github.com/junsupan/TensorPCA`.

6 Conclusions

Machine learning methods are attracting significant attention in economics and finance. The success of these methods stems from their ability to provide flexible regularized approximations to the theoretically optimal decision rules in data-rich environments. The empirical application of machine learning in economics and finance involves several methodological challenges. In this survey, we cover some of the interesting recent developments that address these challenges. This is an exciting and rapidly growing area of research and we foresee many interesting methodological developments and applications in the future.

Lastly, we referred to the statistical packages **midasml** (R) and **TensorPCA** (Python) that can fit the high-dimensional regressions and tensor factor models. Many of the standard machine learning routines (LASSO, ridge, elastic net, trees, random forests, etc) are also available in the **scikit-learn** Python library, though special care should be taken to avoid pitfalls with time series data described in the chapter.

References

Abadie, A., A. Diamond, and J. Hainmueller (2010): "Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program," *Journal of the American Statistical Association*, 105(490), 493–505.

Adamek, R., S. Smeekes, and I. Wilms (2023): "LASSO inference for high-dimensional time series," *Journal of Econometrics*, 235(2), 1114–1143.

Andrews, D. W. (1991): "Heteroskedasticity and autocorrelation consistent covariance matrix estimation," *Econometrica*, 59(3), 817–858.

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

Babii, A., R. T. Ball, E. Ghysels, and J. Striaukas (2023): "Machine learning panel data regressions with heavy-tailed dependent data: Theory and application," *Journal of Econometrics* (forthcoming).

——— (2024): "Panel data nowcasting: The case of price-earnings ratios," *Journal of Applied Econometrics* (forthcoming).

Babii, A., X. Chen, E. Ghysels, and R. Kumar (2021): "Binary choice with asymmetric loss in a data-rich environment: Theory and an application to racial justice," *arXiv preprint arXiv:2010.08463v5*.

Babii, A., and J.-P. Florens (2017): "Is completeness necessary? Estimation in nonidentified linear models," *arXiv preprint arXiv:1709.03473*.

Babii, A., E. Ghysels, and J. Pan (2023): "Tensor Principal Component Analysis," *arXiv preprint arXiv:2212.12981*.

Babii, A., E. Ghysels, and J. Striaukas (2022): "Machine learning time series regressions with an application to nowcasting," *Journal of Business and Economic Statistics*, 40(3), 1094–1106.

——— (2024): "High-dimensional Granger causality tests with an application to VIX and news," *Journal of Financial Econometrics*, (forthcoming).

Bach, F. (2017): "Breaking the curse of dimensionality with convex neural networks," *The Journal of Machine Learning Research*, 18(1), 629–681.

Bai, J., and S. Ng (2009): "Boosting diffusion indices," *Journal of Applied Econometrics*, 24(4), 607–629.

Ball, R. T., and E. Ghysels (2018): "Automated earnings forecasts: Beat analysts or combine and conquer?," *Management Science*, 64(10), 4936–4952.

Bańbura, M., D. Giannone, M. Modugno, and L. Reichlin (2013): "Nowcasting and the real-time data flow," in *Handbook of Economic Forecasting - Volume 2 Part A*, ed. by G. Elliott, and A. Timmermann, pp. 195–237. Elsevier.

Barbaglia, L., S. Manzan, and E. Tosetti (2023): "Forecasting loan default in Europe with machine learning," *Journal of Financial Econometrics*, 21(2), 569–596.

Bartlett, P. L., A. Montanari, and A. Rakhlin (2021): "Deep learning: a statistical viewpoint," *Acta numerica*, 30, 87–201.

Belloni, A., V. Chernozhukov, and C. Hansen (2014): "Inference on treatment effects after selection among high-dimensional controls," *Review of Economic Studies*, 81(2), 608–650.

Bergmeir, C., R. J. Hyndman, and B. Koo (2018): "A note on the validity of cross-validation for evaluating autoregressive time series prediction," *Computational Statistics and Data Analysis*, 120, 70–83.

Berner, J., P. Grohs, G. Kutyniok, and P. Petersen (2021): "The modern mathematics of deep learning," *arXiv preprint arXiv:2105.04026*, pp. 86–114.

Beyhum, J., and J. Striaukas (2023): "Sparse plus dense MIDAS regressions and nowcasting during the COVID pandemic," *arXiv preprint arXiv:2306.13362*.

Biau, G., and B. Cadre (2021): "Optimization by gradient boosting," in *Advances in Contemporary Statistics and Econometrics: Festschrift in Honor of Christine Thomas-Agnan*, pp. 23–44. Springer.

Blanchard, G., G. Lugosi, and N. Vayatis (2003): "On the rate of convergence of regularized boosting classifiers," *Journal of Machine Learning Research*, 4(Oct), 861–894.

Borup, D., D. E. Rapach, and E. C. M. Schütte (2023): "Mixed-frequency machine learning: Nowcasting and backcasting weekly initial claims with daily internet search volume data," *International Journal of Forecasting*, 39(3), 1122–1144.

Bradley, E., and H. Trevor (2021): *Computer age statistical inference: Algorithms, evidence, and data science*. Cambridge University Press.

Bredahl Kock, A., and T. Teräsvirta (2016): "Forecasting macroeconomic variables using neural network models and three automated model selection techniques," *Econometric Reviews*, 35(8-10), 1753–1779.

Breiman, L. (2001a): "Random forests," *Machine learning*, 45, 5–32.

——— (2001b): "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," *Statistical Science*, 16(3), 199–231.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984): *Classification and regression trees*. CRC press.

Bryzgalova, S. (2015): "Spurious factors in linear asset pricing models," LSE Discussion Paper.

Bryzgalova, S., M. Pelger, and J. Zhu (2024): "Forest through the trees: Building cross-sections of stock returns," *Journal of Finance*, (forthcoming).

Bybee, L., B. T. Kelly, A. Manela, and D. Xiu (2020): "The structure of economic news," Discussion paper, National Bureau of Economic Research.

Carrasco, M., J.-P. Florens, and E. Renault (2007): "Linear inverse problems in structural econometrics estimation based on spectral decomposition and regularization," in *Handbook of Econometrics - Volume 6, Part B*, ed. by J. Heckman, and E. Leamer, pp. 5633–5751. Elsevier.

Carrasco, M., and B. Rossi (2016): "In-sample inference and forecasting in misspecified factor models," *Journal of Business & Economic Statistics*, 34(3), 313–338.

Carvalho, C., R. Masini, and M. C. Medeiros (2018): "ArCo: An artificial counterfactual approach for high-dimensional panel time-series data," *Journal of Econometrics*, 207(2), 352–380.

Cattaneo, M. D., R. Chandak, and J. M. Klusowski (2022): "Convergence rates of oblique regression trees for flexible function libraries," *arXiv preprint arXiv:2210.14429*.

Chen, B., and K. Maung (2023): "Time-varying forecast combination for high-dimensional data," *Journal of Econometrics* (forthcoming).

Chen, R., D. Yang, and C.-H. Zhang (2022): "Factor models for high-dimensional tensor time series," *Journal of the American Statistical Association*, 117(537), 94–116.

Chen, X. (2007): "Large sample sieve estimation of semi-nonparametric models," in *Handbook of Econometrics - Volume 6, Part B*, ed. by J. Heckman, and E. Leamer, pp. 5549–5632. Elsevier.

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: Double/debiased machine learning," *The Econometrics Journal*, 21(1), C1–C68.

Chernozhukov, V., W. Härdle, C. Huang, and W. Wang (2021): "Lasso-driven inference in time and space," *Annals of Statistics*, 49(3), 1702–1735.

Chi, C.-M., P. Vossler, Y. Fan, and J. Lv (2022): "Asymptotic properties of high-dimensional random forests," *Annals of Statistics*, 50(6), 3415–3438.

Christensen, T., H. R. Moon, and F. Schorfheide (2020): "Robust forecasting," *arXiv preprint arXiv:2011.03153*.

Chu, C.-K., and J. S. Marron (1991): "Comparison of two bandwidth selectors with dependent errors," *Annals of Statistics*, 19(4), 1906–1918.

Cimadomo, J., D. Giannone, M. Lenza, F. Monti, and A. Sokol (2022): "Nowcasting with large Bayesian vector autoregressions," *Journal of Econometrics*, 231(2), 500–519.

Coulombe, P. G. (2024): "The macroeconomy as a random forest," *Journal of Applied Econometrics (forthcoming)*.

Coulombe, P. G., M. Leroux, D. Stevanovic, and S. Surprenant (2022): "How is machine learning useful for macroeconomic forecasting?," *Journal of Applied Econometrics*, 37(5), 920–964.

Diebold, F. X., and M. Shin (2019): "Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives," *International Journal of Forecasting*, 35(4), 1679–1691.

Ellingsen, J., V. H. Larsen, and L. A. Thorsrud (2022): "News media versus FRED-MD for macroeconomic forecasting," *Journal of Applied Econometrics*, 37(1), 63–81.

Elliott, G., A. Gargano, and A. Timmermann (2013): "Complete subset regressions," *Journal of Econometrics*, 177(2), 357–373.

Elliott, G., and R. P. Lieli (2013): "Predicting binary outcomes," *Journal of Econometrics*, 174(1), 15–26.

Elliott, G., and A. Timmermann (2016): *Economic Forecasting*. Princeton University Press.

Farrell, M. H., T. Liang, and S. Misra (2021): "Deep neural networks for estimation and inference," *Econometrica*, 89(1), 181–213.

Feng, G., S. Giglio, and D. Xiu (2020): "Taming the factor zoo: A test of new factors," *Journal of Finance*, 75(3), 1327–1370.

Ferrara, L., and A. Simoni (2022): "When are Google data useful to nowcast GDP? An approach via preselection and shrinkage," *Journal of Business and Economic Statistics*, 41(4), 1188–1202.

Florios, K., and S. Skouras (2008): "Exact computation of max weighted score estimators," *Journal of Econometrics*, 146(1), 86–91.

Fosten, J., and R. Greenaway-McGrevy (2022): "Panel data nowcasting," *Econometric Reviews*, 41(7), 675–696.

Freyberger, J., A. Neuhierl, and M. Weber (2020): "Dissecting characteristics nonparametrically," *Review of Financial Studies*, 33(5), 2326–2377.

Friedberg, R., J. Tibshirani, S. Athey, and S. Wager (2020): "Local linear forests," *Journal of Computational and Graphical Statistics*, 30(2), 503–517.

Friedman, J. H. (2001): "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232.

Friedman, J. H., and P. Hall (2007): "On bagging and nonlinear estimation," *Journal of statistical planning and inference*, 137(3), 669–683.

Ghysels, E., F. Grigoris, and N. Özkan (2022): "Real-time forecasts of state and local government budgets with an application to the COVID-19 pandemic," *National Tax Journal*, 75(4), 731–763.

Ghysels, E., C. Horan, and E. Moench (2018): "Forecasting through the rearview mirror: Data revisions and bond return predictability," *Review of Financial Studies*, 31(2), 678–714.

Ghysels, E., P. Santa-Clara, and R. Valkanov (2006): "Predicting volatility: getting the most out of return data sampled at different frequencies," *Journal of Econometrics*, 131(1-2), 59–95.

Giannone, D., L. Reichlin, and D. Small (2008): "Nowcasting: The real-time informational content of macroeconomic data," *Journal of Monetary Economics*, 55(4), 665–676.

Giglio, S., B. Kelly, and D. Xiu (2022): "Factor models, machine learning, and asset pricing," *Annual Review of Financial Economics*, 14, 337–368.

Granger, C. W. (1969a): "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, 39(3), 424–438.

——— (1969b): "Prediction with a generalized cost of error function," *Journal of the Operational Research Society*, 20(2), 199–207.

Granger, C. W., and M. H. Pesaran (2000): "Economic and statistical measures of forecast accuracy," *Journal of Forecasting*, 19(7), 537–560.

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

Han, Y., R. Chen, and C.-H. Zhang (2022): "Rank determination in tensor factor model," *Electronic Journal of Statistics*, 16(1), 1726–1803.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009): *The elements of statistical learning: data mining, inference, and prediction*. Springer.

Hecq, A., L. Margaritella, and S. Smeekes (2023): "Granger causality testing in high-dimensional VARs: a post-double-selection procedure," *Journal of Financial Econometrics*, 21(3), 915–958.

Hecq, A., M. Ternes, and I. Wilms (2023): "Hierarchical regularizers for reverse unrestricted mixed data sampling regressions," *arXiv preprint arXiv:2301.10592*.

Hoerl, A. E., and R. W. Kennard (1970a): "Ridge regression: applications to nonorthogonal problems," *Technometrics*, 12(1), 69–82.

——— (1970b): "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, 12(1), 55–67.

Hornik, K., M. Stinchcombe, and H. White (1990): "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural networks*, 3(5), 551–560.

Inoue, A., and L. Kilian (2008): "How useful is bagging in forecasting economic time series? A case study of US consumer price inflation," *Journal of the American Statistical Association*, pp. 511–522.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013): *An introduction to statistical learning*. Springer.

James, W., and C. Stein (1961): "Estimation with quadratic loss," in *Proceedings Fourth Berkeley Symposium on Mathematical Statistics and Probability, 1961*.

Jardet, C., and B. Meunier (2022): "Nowcasting world GDP growth with high-frequency data," *Journal of Forecasting*, 41(6), 1181–1200.

Karoui, N. E. (2003): "On the largest eigenvalue of Wishart matrices with identity covariance when n, p and p/n tend to infinity," *arXiv preprint math/0309355*.

Kitagawa, T., S. Sakaguchi, and A. Tetenov (2021): "Constrained classification and policy learning," *arXiv preprint arXiv:2106.12886*.

Kleinberg, J., H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan (2018): "Human decisions and machine predictions," *The quarterly journal of economics*, 133(1), 237–293.

Kock, A. B. (2016): "Consistent and conservative model selection with the adaptive lasso in stationary and nonstationary autoregressions," *Econometric Theory*, 32(1), 243–259.

Kock, A. B., and L. Callot (2015): "Oracle inequalities for high dimensional vector autoregressions," *Journal of Econometrics*, 186(2), 325–344.

Kolda, T. G., and B. W. Bader (2009): "Tensor decompositions and applications," *SIAM Review*, 51(3), 455–500.

Lahiri, K., and C. Yang (2022): "Boosting tax revenues with mixed-frequency data in the aftermath of COVID-19: The case of New York," *International Journal of Forecasting*, 38(2), 545–566.

Lahiri, K., and L. Yang (2013): "Forecasting binary outcomes," in *Handbook of Economic Forecasting - Volume 2, Part B*, ed. by G. Elliott, and A. Timmermann, pp. 1025–1106. Elsevier.

Lettau, M., and M. Pelger (2020): "Factors that fit the time series and cross-section of stock returns," *Review of Financial Studies*, 33(5), 2274–2325.

Li, D., M. Plagborg-Møller, and C. K. Wolf (2022): "Local projections vs. VARs: Lessons from thousands of DGPs," Discussion paper, National Bureau of Economic Research.

Makridakis, S., E. Spiliotis, and V. Assimakopoulos (2020): "The M4 Competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, 36(1), 54–74.

——— (2022): "M5 accuracy competition: Results, findings, and conclusions," *International Journal of Forecasting*, 38(4), 1346–1364.

Manski, C. F., and T. S. Thompson (1989): "Estimation of best predictors of binary response," *Journal of Econometrics*, 40(1), 97–123.

Masini, R. P., M. C. Medeiros, and E. F. Mendes (2022): "Regularized estimation of high-dimensional vector autoregressions with weakly dependent innovations," *Journal of Time Series Analysis*, 43(4), 532–557.

——— (2023): "Machine learning advances for time series forecasting," *Journal of Economic Surveys*, 37(1), 76–111.

Medeiros, M. C., and E. F. Mendes (2016): "ℓ_1-regularization of high-dimensional time-series models with non-Gaussian and heteroskedastic errors," *Journal of Econometrics*, 191(1), 255–271.

——— (2017): "Adaptive LASSO estimation for ARDL models with GARCH innovations," *Econometric Reviews*, 36(6-9), 622–637.

Medeiros, M. C., G. F. Vasconcelos, Á. Veiga, and E. Zilberman (2021): "Forecasting inflation in a data-rich environment: the benefits of machine learning methods," *Journal of Business and Economic Statistics*, 39(1), 98–119.

Mei, Z., P. C. Phillips, and Z. Shi (2022): "The boosted HP filter is more general than you might think," *arXiv preprint arXiv:2209.09810*.

Mhaskar, H. N. (1996): "Neural networks for optimal approximation of smooth and analytic functions," *Neural computation*, 8(1), 164–177.

Mogliani, M., and A. Simoni (2021): "Bayesian MIDAS penalized regressions: estimation, selection, and prediction," *Journal of Econometrics*, 222(1), 833–860.

Mullainathan, S., and J. Spiess (2017): "Machine learning: an applied econometric approach," *Journal of Economic Perspectives*, 31(2), 87–106.

Newey, W. K., and K. D. West (1987): "A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix," *Econometrica*, 55(3), 703–708.

Nobel, A. (1996): "Histogram regression estimation using data-dependent partitions," *Annals of Statistics*, 24(3), 1084–1105.

Onatski, A. (2009): "Testing hypotheses about the number of factors in large factor models," *Econometrica*, 77(5), 1447–1479.

Pellatt, D. F., and Y. Sun (2013): "Binary forecast and decision rules via PAC bayesian model aggregation," *UCSD Working Paper, Economics Department*.

Phillips, P. C., and Z. Shi (2021): "Boosting: Why you can use the HP filter," *International Economic Review*, 62(2), 521–570.

Politis, D. N., and J. P. Romano (1994): "The stationary bootstrap," *Journal of the American Statistical Association*, 89(428), 1303–1313.

Rambachan, A., J. Kleinberg, S. Mullainathan, and J. Ludwig (2020): "An economic approach to regulating algorithms," Discussion paper, National Bureau of Economic Research.

Rossi, A. G., and A. Timmermann (2015): "Modeling covariance risk in Merton's ICAPM," *Review of Financial Studies*, 28(5), 1428–1461.

Scornet, E., G. Biau, and J.-P. Vert (2015): "Consistency of random forests," *Annals of Statistics*, 43(4), 1716–1741.

Simon, N., J. Friedman, T. Hastie, and R. Tibshirani (2013): "A sparse-group LASSO," *Journal of Computational and Graphical Statistics*, 22(2), 231–245.

Stone, C. J. (1982): "Optimal global rates of convergence for nonparametric regression," *Annals of Statistics*, pp. 1040–1053.

Syrgkanis, V., and M. Zampetakis (2020): "Estimation and inference with trees and forests in high dimensions," in *Conference on Learning Theory*, pp. 3453–3454. PMLR.

Tibshirani, R. (1996): "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), 267–288.

Tikhonov, A. N. (1963): "On the solution of ill-posed problems and the method of regularization," in *Doklady akademii nauk*, vol. 151(3), pp. 501–504. Russian Academy of Sciences.

Van Binsbergen, J. H., X. Han, and A. Lopez-Lira (2023): "Man versus machine learning: The term structure of earnings expectations and conditional biases," *Review of Financial Studies*, 36(6), 2361–2396.

Vapnik, V. (1999): *The nature of statistical learning theory*. Springer.

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

Viviano, D., and J. Bradic (2023): "Fair policy targeting," *Journal of the American Statistical Association*, (forthcoming).

Vovk, V., and R. Wang (2020): "Combining p-values via averaging," *Biometrika*, 107(4), 791–808.

Wald, A. (1949): "Statistical decision functions," *The Annals of Mathematical Statistics*, 20(2), 165–205.

Wong, K. C., Z. Li, and A. Tewari (2020): "Lasso guarantees for β-mixing heavy-tailed time series," *Annals of Statistics*, 48(2), 1124–1142.

Xu, Q., Z. Wang, C. Jiang, and Y. Liu (2023): "Deep learning on mixed frequency data," *Journal of Forecasting*, 42(8), 2099–2120.

Yarotsky, D. (2017): "Error bounds for approximations with deep ReLU networks," *Neural Networks*, 94, 103–114.

Yuan, M., and Y. Lin (2006): "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1), 49–67.