Prediction and Variable Selection with Multicollinear, High-Dimensional Macroeconomic Data

Evan Munro, Nuffield College

May 2018

Word Count: 20,000

Submitted in partial fulfilment of the requirements for the degree of Master of Philosophy in Economics.

Abstract

Lasso is increasingly found in the economics literature, but boosting, which is a simple and flexible high-dimensional estimation procedure that has been used successfully in genetics, computer science, and other fields, is not familiar to most economists. I describe the close theoretical ties between a linear varient of general gradient boosting, L_2-Boosting, and lasso and the conditions required for each to guarantee prediction and model selection consistency. For the first time in the economics literature, I compare the performance of boosting and lasso for both variable selection and prediction accuracy. Furthermore, I address the specific issues that arise under block-correlation typically found in macroeconomic datasets. In simulations, I find that lasso selects a more parsimonious model that is closer to the truth while maintaining prediction accuracy. In an application to forecasting series in the FRED-MD dataset, I find that the forecasting performance of L_2-boost and lasso are close to equivalent at 1 month forecast horizons and significantly better than the AR baseline, with mixed results at the 6 month horizon. There are some indications that a non-linear form of gradient boosting has the best performance for longer time horizons. Since I show that lasso and boosting are not stable under correlated data and lack of sparsity, I describe how for macroeconomic data the variable selection output can be interpreted more robustly by aggregating variables in groupings.

Contents

List of Figures

List of Tables

1 Introduction

Estimation of reduced form models has a variety of applications in macroeconomics; two primary ones are model selection and forecasting. New challenges arise in the estimation of these models when the predictor set is very large, especially in situations where the number of predictor variables is larger than the number of observations when OLS is not tractable. With a large number of potential predictor variables, many of which are highly correlated in high-dimensional macroeconomic datasets, it is difficult to distinguish between true zero and non-zero variables; furthermore, the variance introduced by adding too many variables can result in poor out-of-sample forecasting performance if not managed appropriately. Some traditional methods have been applied to select models in high-dimensional settings: for example, statistical methods like Autometrics (Doornik & Hendry, 2015). Common factor models have also been used with success to forecast macroeconomic series using predictor sets that are highly correlated (Stock & Watson, 2002a), (Stock & Watson, 2002b). However, statistical methods like Autometrics are computationally intensive, and factor models, while providing good forecasting performance, don't provide direct variable selection that is often useful in interpreting the results from the estimation of reduced form models.

Economists working with high-dimensional data have increasingly used machine learning methods that simultaneously provide variable selection and estimation, such as lasso regression, ridge regression, and boosting. These methods can handle very large datasets with computational efficiency. Lasso and boosting generally have been shown to have comparable or lower mean-square forecast error (MSFE) in prediction tasks compared to factor and simple linear methods, but also provide directly interpretable output (Li & Chen, 2014), (Wohlrabe & Buchen, 2014). There has not been, however, a systematic comparison of lasso and boosting in the literature that also examines the limits of their performance under collinearity and lack of sparsity that occurs in high-dimensional macroeconomic data.

In this paper, I study the performance of lasso-type measures and a linear form of boosting, L_2-Boost, in a macroeconomic prediction context where the accuracy of variable selection also matters. Ng (2013) reviews criterion-based, regularization, and dimension reduction methods of selecting predictors in a high-dimensional context using simulations and describes the unresolvable tradeoff between prediction accuracy and consistent model determination. It has been theoretically proven that it is not possible to select one criterion (Yang, 2005) nor one regularization parameter for lasso (Meinshausen & Bühlmann, 2006) that is optimal for both prediction accuracy and variable selection. Though it has not yet been shown explicitly, boosting likely has the same optimality trade-off for the stopping parameter. It is still interesting, however, to determine which methods have the ability to perform both tasks well, even if optimal performance for both is not possible.

Both lasso and L_2-Boost are known to be consistent for prediction under a sparsity condition. Results are available for model selection consistency for lasso but only under strict restrictions on the correlation of the predictors that are not likely to hold in macroeconomic data. The LARS algorithm of Efron *et al.* (2004) united lasso and forward stagewise regression, a variant of L_2-Boosting. Freund *et al.* (2017) and Hastie *et al.* (2007) have separately showed that there are strong theoretical links between linear boosting and lasso; the former shows each are the solution to a problem by subgradient optimization and the latter shows each are differential equations that are optimal in terms of a local optimization procedure. These authors show that a) there are versions of lasso that provide the same solution as L_2-Boost under restrictions on the path of the lasso coefficients as the regularization parameter varies and b) there are restricted versions of linear boosting that provide the same solution as lasso. Understanding these theorems provides motivation for studying the variable selection and prediction performance of lasso and boosting together when dealing with difficulties in high-dimensional macroeconomic data, such as collinearity and lack of sparsity.

Lasso is known to have issues with stability of coefficients and model selection consistency under lack of sparsity and collinearity; this is shown in simulations and applications to dense macroeconomic data by Giannone *et al.* (2017) and by Li & Chen (2014). With slight perturbations of the data or changes in time window, the predictors selected by lasso can change dramatically. This issue can be mitigated by using grouped lasso or elastic net, but grouped lasso forces a manually determined structure on the lasso penalty term that may not be desirable in a prediction context (Callot & Kock, 2014), and elastic net still has stability issues as the penalty term is varied, which I will describe using Monte Carlo simulations. Furthermore, alternative estimators for high-dimensional datasets such as dynamic factor models (Stock & Watson, 2002a), do not provide output that is easily interpretable for economic data. Boosting, on the other hand, has been studied less in the economics literature, but has been used with great success for prediction of categorical variables in the computer science literature. Given its close theoretical links to lasso, one would expect that boosting would have some of the same difficulties in dense and collinear economic applications. However, one of the key differences between lasso and boosting is that boosting has a monotone coefficient path as the regularization parameter varies. I examine the nuances of this theoretical difference in both simulations and applications to macroeconomic forecasting. Along with L_2-Boosting and lasso, I also examine the performance of a non-linear form of boosting, tree boosting, and elastic net, which is a variant of lasso that was designed to better handle groups of correlated variables.

In simulations, I find that the performance of L_2-Boosting is even more sensitive than lasso or elastic net to density in a block-correlated data generating process. The variable selection results of all methods worsen under increased density and correlation. Though significant issues arise in all methods, lasso performs the best in both prediction and model selection when variables are highly correlated and the data generating process isn't sparse. However, under sparsity and reasonable collinearity L_2-boosting performance is equivalent to regularized regression. I find that, in applications to forecasting 4 macroeconomic series (real production, unemployment, price and interest rate series) from the FRED-MD database (McCracken & Ng, 2016) 1, 3, and 6 months ahead, the linear high-dimensional methods generally perform better than no-change or AR baselines, especially at the 1 month horizon, and that L_2-Boosting, elastic net, and lasso have similar MSFEs, with no method consistently beating the others. The results indicate that linear models in general beat more flexible non-linear alternatives such as boosted regression trees at short time horizons, but that the non-linear boosting method does not overfit substantially. For the 3 month horizon, where linear methods do not perform better than the AR model, tree boosting shows marginal improvement. In selecting variables for the 1 month ahead forecasts, the results are nearly identical for lasso and boosting, which, before examining the theory behind the methods, would be surprising given the very different formulation of the estimation algorithms for lasso-type methods and boosting.

Aggregating variables by the groups defined in the Fred-MD appendix results in clear interpretation that is less likely to be muddled by issues with the correlation of the individual series, since the blocks have low correlation between them. For example, the results indicate that increases in output, employment and decreases in financing costs are most closely associated with 1 month ahead increases in industrial production, which closely follows economic intuiton. I posit, though I do not prove, that lasso and L_2-Boost are block-consistent, meaning they correctly select variables at a block-level. I also conduct robustness checks on the variable selection results using OLS and factor model alternatives. I find that the top variables selected by lasso and L_2-Boost are highly significant in an OLS regression. Using simple PCA-based methods to select group-specific factors, an alternative to aggregating results in groups from high-dimensional estimation methods, does not perform well.

Section 2 reviews related work. Section 3 describes gradient boosting and penalized regression and interprets the methods from the perspective of various kinds of general optimization methods. Section 4 provides an overview of the theoretical results for prediction and model selection consistency and for the relationship between boosting and lasso which motivate the applied work. Section 5 provides Monte Carlo simulations that illustrate the issues that arise in collinear high-dimensional time series settings. Section 6 provides an application of boosting and lasso methods to prediction and variable selection for forecasting four U.S. macroeconomic series.

2 Related Work

In this section, I briefly describe the literature on boosting and lasso for economic forecasting, and other approaches that have been suggested when dealing with block-correlated datasets.

Boosting

In the economics literature, boosting has been used for financial time series forecasting, but the literature on boosting for macroeconomic forecasting is rather sparse. For binary dependent variables, Ng (2014) used AdaBoost with decision stumps on a database of 132 U.S. financial and real series to identify important predictors for U.S. recessions 3m, 6m, and 12m ahead and Dopke *et al.* (2017) uses boosted decision trees to predict German recessions with lower out of sample performance than probit approaches. Both papers look at which variables boosting selects as important, using Friedman's importance coefficient for boosting with decision trees, and finding that term spreads, as expected, are the most important predictors of recessions.

For continuous dependent variables, Bai & Ng (2009) uses boosting to select predictors in factor-augmented regressions and finds that prediction performance is improved by the boosting selection method compared to criterion-based techniques. Wohlrabe & Buchen (2014) tests the forecasting performance of boosting for U.S., Euro area and German data and Buchen & Wohlrabe (2011) evaluates boosting compared to dynamic factor models and model averaging methods for forecasting U.S. industrial production. Lehmann &

Wohlrabe (2017) uses boosting to forecast regional German economic indicators and finds that boosting outperforms the benchmark for regional economic forecasting. Robinzonov *et al.* (2012) uses boosting with nonlinear base learners in a high-dimensional time series setting to estimate nonlinear lag functions. Taieb *et al.* (2014) proposes a boosting autoregression procedure and evaluates performance in two time series forecasting competitions. Few of the papers in the boosting forecasting literature attempt to evaluate the prediction models in terms of variable selection, apart from Lehmann & Wohlrabe (2016) which looks at counts of how often a variable is selected to forecast German industrial production at different time horizons to determine which are important.

Lasso

Li & Chen (2014) evaluate the performance of several lasso-based approaches, including regular lasso, grouped lasso, and elastic net, compared to dynamic factor models for twenty U.S. macroeconomic variables. They find that lasso approaches are better than dynamic factor models in out of sample forecasting exercise and that combining lasso and dynamic factor model forecasts are better than either method individually. They also suggest manually grouping predictors into economically meaningful blocks and using group lasso or elastic net to improve the interpretability and stability of such models. Callot & Kock (2014) evaluate the forecasting accuracy and variable selection of lasso and some of its variants, adaptive and adaptive group lasso on a large U.S. macroeconomic dataset. They analyze the performance of the methods for different groups and find that lasso performs best, but the adaptive versions perform similarly to factor models. Kim & Swanson (2011) uses recursive estimation to test the predictive accuracy of a variety of models based on principal components or shrinkage methods, including lasso, boosting, elastic net, factor models, and various model combination methods. They find that factor-augmented models constructed with shrinkage methods, such as those introduced in Bai & Ng (2009), have the lowest out of sample error when predicting eleven macroeconomic variables at various time horizons.

Models for Block-Correlated Data

Variants of lasso have been proposed to vary the penalty term to better account for blocks of related variables in data, such as elastic net, grouped lasso and the adaptive grouped lasso (Zou & Hastie, 2005), (Yuan & Lin, 2006), (Wang & Leng, 2008). I include elastic net in the simulation and application results. Factor models taking into account the block-structure of economic data have also been introduced. For example, Moench *et al.* (2013) introduces a hierarchical model that includes block-specific factors within economically-meaningful blocks along with common factors to increase interpretability of dynamic factor models. I use a simplified version of the model to test the robustness of the variable selection results in Section 6. Bai & Ng (2009) derives Block Boosting from modifying the typical linear boosting procedure in a factor augmented regression setting to take into account the relationship between a variable and its lags.

Comparing Approaches

In the biostatistics literature, Hepp *et al.* (2016) compare the performance of variable selection stability and forecasting for L_2-boosting and lasso in a variety of simulated settings, varying collinearity, true sparsity, and signal-to-noise ratio, and finds that results are similar for both methods. Ng (2013) studies model selection and prediction together in a high-dimensional simulation using monte-carlo methods and finds that factor methods perform more accurately at both tasks when the data generating process is dense, and that regularization methods perform better when the data generating process is sparse.

3 Model and Methods

In this section, I introduce the model and notation used in Sections 5 and 6. I also describe the estimation procedures in detail, which is necessary for understanding the theory in Section 4 and the results in Section 5 and 6.

3.1 Notation

Let x be a vector in $\mathbb{R}^n$. $||x||_2 = \sqrt{\sum_{i=1}^{n}(x_i)^2}$, $||x||_2^2 = \sum_{i=1}^{n}(x_i)^2$, $||x||_1 = \sum_{i=1}^{n}|x_i|$, $||x||_\infty = \max_i |x_i|$.

3.2 Model

For the forecasting simulations and predictions below I used restricted versions of the below predictive model, which is common in the forecasting literature. y_t from $t = 1, \ldots, T$ is the stationary-transformed, continuous-valued, target variable. There are V variables x_{it} from $t = 1, \ldots, T$ available that can potentially explain y_t, along with their lags and lags of y_t itself, up to a maximum of K lags.

$$y_{t+h} = \beta_{00} + \sum_{k=0}^{K} \alpha_k y_{t-k} + \sum_{i=1}^{V}\sum_{k=0}^{K} \beta_{ik} x_{i,(t-k)} + \epsilon_{t+h}, \qquad t = 1, \ldots, T \tag{3.1}$$

To simplify the notation in x_t denote the $p = (V * (k+1) + (k+1) + 1)$ row vector of RHS variables for y_{t+h} at time t and let β denote the possibly sparse vector of RHS parameters relating the predictors to y_{t+h}. It is possible that $p >> T$ and that x_t contains many collinear predictors. The above equation is summarized as

$$y_{t+h} = x_t\beta + \epsilon_{t+h}, \qquad t = 1, \ldots, T \tag{3.2}$$

3.3 Estimation Methods

In the machine learning literature that derived lasso and boosting, procedures for estimating predictive functions for a dependent variable from a set of possible predictors are known as learning algorithms when the performance of the method improves with additional data. A simple learning algorithm is linear regression, for example. A learning algorithm takes as input a labeled sequence of training examples $(x_1, y_1), \ldots, (x_T, y_T)$ and uses these to construct a function $\phi(x_t)$ that will classify new instances x_t. y_t may be categorical, binary, or real-valued. Each of the below models is a learning algorithm for continuous y_t that has applicability to high-dimensional problems, where the dimension of x_t, p, is larger than T.

3.3.1 Lasso-Type Methods

Given observations on y_t and each of p observed predictors $x_t = (x_{t1}, ..., x_{tp})$ for $t = 1, \ldots, T$. Under $p >> T$, OLS is not defined, since it requires the $X'X/T$ to be positive definite. Furthermore, in high-dimensional collinear settings, even when $p < T$, OLS does not perform well for prediction or interpretation. Tibshirani (1996) proposed L_1 penalized regression, which performs variable selection and model estimation simultaneously, and shows improved performance in prediction and model parsimony compared to OLS due to reduction in variance introduced by the penalty term, λ, at the cost of the introduction of bias.

$$\hat{\beta}^{(lasso)} = \arg\min_{\beta} \sum_{t=1}^{T} (y_t - x_t\beta)^2 + \sum_{j=1}^{p} \lambda|\beta_j| \tag{3.3}$$

The resulting lasso function is $\hat{\phi}^{(lasso)}(x_t) = x_t\hat{\beta}^{(lasso)}$

Elastic Net

Zou & Hastie (2005) propose elastic net to address some of the issues that lasso has when predictors are high-dimensional and correlated. I don't provide theoretical results for elastic net since it is considered as part of the lasso family; however, given it has been proposed to improve prediction performance in the context of collinear predictors I include it the simulation and application result. Lasso tends to select only one of a group of correlated predictors and switches between them with small changes to the regularization parameter (see Section 5). Elastic net, on the other hand, tends to select correlated variables in groups. The elastic net estimator is defined by the below minimization function. α is the weight on the l_1 penalization and $1 - \alpha$ is the weight on the l_2 penalization and λ is the penalization term on the size of the coefficients.

$$\hat{\beta}^{(enet)} = \arg\min_{\beta} \sum_{t=1}^{T} (y_t - x_t'\beta)^2 + \alpha\lambda \sum_{j=1}^{p} |\beta_j| + (1-\alpha)\lambda \sum_{j=1}^{p} (\beta_j)^2 \tag{3.4}$$

The elastic net function is $\hat{\phi}^{(enet)}(x_t) = x_t'\hat{\beta}^{(enet)}$

3.3.2 Gradient Boosting

My paper is mainly concerned with determining how boosting compares theoretically and in an applied sense to lasso and its variants, since boosting is not yet very familiar to economists and has performed very well in prediction in other disciplines. Boosting makes a prediction by efficiently combining the predictions of many simple models, known in the machine learning literature as weak learners. For modeling a continous dependent variable, there are a variety of weak learners available, including single variable linear regressions, and non-linear learners such as k-splines and regression trees. Boosting is very modular and can be used to model a variety of dependent variable types; for binary variables, for example, boosting combines classifiers like single variable logistic regressions or shallow decision trees.

The first form of boosting was AdaBoost, which has been used for binary classification succesfully and minimizes a form of exponential loss, and is due to Freund *et al.*

(1996). After the applied success of AdaBoost, a significant amount of work went into understanding the properties of boosting in a game theoretic and online learning context (see Schapire & Freund (2012) for a good overview), as well as generalizing the algorithm to different loss functions and to categorical and continuous dependent variables. The general form of gradient boosting presented below for modeling functions of continuous variables is due to Friedman (2001). I also clarify at each step the choices that will give L_2-Boost, which is the linear form of boosting that I focus on in this paper.

Given observations on y_t and each of p observed predictors $x_t = (x_{t1}, ..., x_{tp})$ for $t = 1, \ldots, T$, let $\phi(x_t)$ be a function on $\mathbb{R}^p$ and $C(y_t, \phi(x_t))$ be the loss function that penalizes the deviation of $\phi(x_t)$ from y_t. For L_2-Boost, choose the error function $C(y_t, \phi(x_t)) = \frac{1}{2}(y_t - \phi(x_t))^2$, which is the quadratic loss function. The following steps give the solution to the gradient boosting algorithm.

1. $\hat{\phi}_0(x_t) = \bar{y}$

2. For $m = 1, \ldots, M$

   - For $t = 1, \ldots, T$, compute the negative gradient vector

$$u_t^{(m)} = \frac{-\delta C(y_t, \phi)}{\delta \phi}|_{\phi = \hat{\phi}_{m-1}(x_t)}. \tag{3.5}$$

   Under the quadratic loss function $u_t^{(m)} = y_t - \hat{\phi}_{m-1}(x_t)$;

   - Fit a base learner to the gradient vector to yield the update for $\hat{\phi}_m$. For L_2-Boost, the base learner is a single variable regression. Calculate

$$\hat{\beta}_{j_m} = \frac{\sum\limits_{t=1}^{T} x_{j_m,t} u_t^{(m)}}{\sum\limits_{t=1}^{T} x_{j_m,t}^2},$$

a single variable regression coefficient, where

$$j_m = \arg\min_{1\leq j\leq p} \sum_{t=1}^{T} (u_t^{(m)} - \hat{\beta}_j x_{jt})^2$$

and corresponds to the index of the single variable that is most correlated to the current residuals $u_t^{(m)}$. For L_2-Boost, $g_m(x_t) = x_{jt}\hat{\beta}_{j_m}$

3. Update $\hat{\phi}_m(x_t) = \hat{\phi}_{m-1}(x_t) + vg_m(x_t)$, where $0 \leq v \leq 1$ is the step length.

Under quadratic loss function and with single variable regression as the base learner, the algorithm is known as L_2-Boost. Forward stagewise linear regression (FSLR), which is closely related to L_2-Boost and will be discussed when evaluating the connections between lasso and boosting, is formed from the same procedure except that $g_m(x_t) = sign(\hat{\beta}_j)x_{jt}$. For FSLR, the coefficient update on x_{jt} is made in the direction of the coefficient $\hat{\beta}_j$ but always at a constant size of v. For L_2-Boost, it is made in the direction of the coefficient but at a variable size of $v\hat{\beta}_j$.

Under L_2-Boost, the final classifier can be expressed as the linear function $\hat{\phi}^{(l2boost)}(x_t) = x_t\hat{\beta}^{(l_2boost)}$, where $\hat{\beta}^{(l2boost)} = [\hat{\beta}_1, \ldots, \hat{\beta}_p]$ and for $i = 1, \ldots, p$

$$\hat{\beta}_i^{(l_2boost)} = \sum_{m=1}^{M} \hat{\beta}_{j_m} \mathbb{1}(i = j_m) \tag{3.6}$$

L_2-Boost can be intepreted as a cautious version of Forward Stepwise regression, which is another well-known method that is a variant of L_2-Boost with stepsize $v = 1$. A model estimated with Forward Stepwise regression is built sequentially by adding one variable at a time that is most correlated with the current residual.

Regression trees can also be used as base learners, $g_m(x_t)$, in the gradient boosting algorithm instead of single variable regressions as for L_2-Boost. A rigorous description of regression trees is too extensive for the scope of this work; instead, a basic overview follows[1]. A regression tree for boosting at step m in the boosting algorithm takes as

[1] See Athey & Imbens (2015) for a more extensive description of regression trees in an economic context

input a target variable, which are the residuals $u_t^{(m)}$, and each of p observed predictors $x_t = (x_{t1}, ..., x_{tp})$ for $t = 1, \ldots, T$. Boosted regression trees create a non-linear estimator of $u_t^{(m)}$ that allow for interaction terms between predictor variables and other, more complex linearities. If the true generating process is not linear, then regression trees may outperform single variable regressions as a base learner. A regression tree is made up a series of nodes, with splits defined as thresholds on the predictors. The splits eventually lead to a terminal node, which is a node with no splits following it, defined as a leaf, which assigns a value to an observation that reaches it. To estimate using a regression tree, an observation starts at the initial node and follows the splits until it is assigned the value for the dependent variable at the first leaf reached. The maximum number of splits from the top of the tree to the leaves of the tree is the depth of the tree. Figure 1 shows an example tree of depth two. The initial parent node splits into two child nodes based on the observation's value for x_{3t}. Then, the tree either assigns a value, or splits again on x_{1t}, depending on which branch was followed from the split on x_{3t}. Each of the leaves give the mean of the target variable for the partition of the training sample that reaches that leaf of the tree.

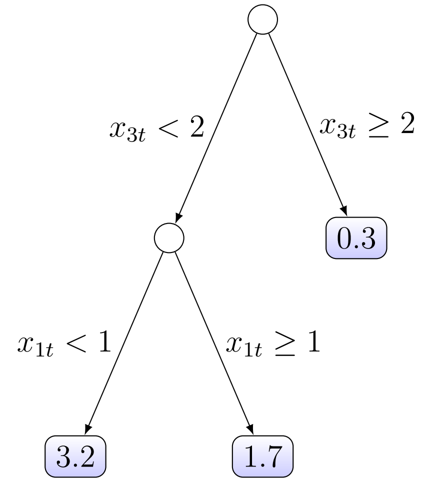


Figure 1: Sample Regression Tree

Let *leaves*(B) be the set of terminal nodes of a tree B. Let t_c be the set of indices corresponding to observations that are assigned to leaf c based on the splits defined for B.

Note that a regression tree partitions the set of training examples so that each training example only reaches one leaf of the tree. The sum of squared errors for a tree B is

$$S = \sum_{c \in leaves(B)} \sum_{j \in t_c} (u_j^{(m)} - m_c)^2$$

where $m_c = \frac{1}{n_c} \sum\limits_{j \in t_c} u_j^{(m)}$ which is the prediction for leaf c and n_c is the number of indices in t_c. The standard regression tree growing algorithm with maximum depth D in a recursive formulation (adapted from Shalizi (2006)) is:

1. Start with single node containing all points.

2. For the node, calculate m_c and S.

3. If all points have the same value for the dependent variables, or if the node is at the maximum depth of the tree D, stop. Otherwise search over all single variable binary splits (of type $x_{it} \geq a$, $x_{it} < a$) to find the variable and the split that reduces S the most. If it is less than some threshold δ or if one of the resulting two nodes contains less than predefined q points then stop. Otherwise, split, creating two new nodes.

4. For each new node, return to step 2.

3.3.3 K-fold Cross-Validation

K-fold cross validation is common in machine learning and is used in non-parametric regression; however, it is not familiar to all econometricians so I give a brief introduction here. Leave one-out cross validation, where the number of subsets K in the process described below is equal to the sample size, has been shown to be equivalent to AIC when the model is estimated by maximum likelihood (Stone, 1977). Model-based criteria generally work better if the model is fully and correctly specified, but cross-validation generally works better in practice due to its flexible and non-parametric form. 10-fold cross-validation is used since it is the standard in the machine learning community and has been shown to provide better results than more computationally expensive methods such as leave-one-out cross validation (Kohavi *et al.* , 1995). I use 10-fold cross-validation to select the regularization parameters for lasso and for boosting for all of the simulation and application results. In cross-validation, the regularization parameter chosen is the one that minimizes the estimated prediction error in the cross-validation task. The method proceeds as follows:

- Split the dataset into K subsets, with the members of each subset chosen randomly
- For each regularization parameter in a reasonable prespecified range :
  - For each of the K subsets, estimate the model on the other $K-1$ subsets and calculate the mean squared error on the subset that is held out from the model estimation. Average the out of sample error across the K subsets.
- The regularization parameter chosen is the one that minimizes the average out of sample error across the K folds.

3.4 Computation

Programming for this paper has been done in R. The estimation of elastic net and lasso models and cross validation are done using the R package glmnet (Friedman *et al.* , 2010), the estimation of L_2-boosting is done using the R package mboost (Hothorn *et al.* , 2012) and tree boosting with gbm (Ridgeway, 2007). Computation of a single boosting, lasso, or elastic net model is efficient under the scenarios considered in the simulation and application, where T ranges from 200 to 700 and N ranges from 120 to 1000. However, under the forecasting simulations, hundreds of cross-validations are performed, and many tens of thousands of boosting, lasso, and elastic net models have to be estimated. This process is infeasible on a laptop computer.

To improve computation speed I have deployed R on a c4.2xlarge Amazon EC2 instance, which is a mid-tier Amazon cloud server optimized for computation with 8 cores. I parallelized the computation of the mean forecast errors across those 8 cores using R package doParallel (Weston, 2014). This reduced computation time for calculating Mean Square Forecast Error vs. the baseline model for cross-validated high dimensional models on one macroeconomic series at one time horizon to a few hours from many days.

The code for the simulation and application results is available upon request.

4 Theory

In this section I describe the basic properties of boosting and penalized regression for variable selection and prediction, as well as the links between the different methods. For the following section, the notation and results are presented for cross-section data. It is left to future work to confirm that all of the results presented can be adapted to a high-dimensional time series setting; existing work like Basu *et al.* (2015) and Kock & Callot (2015) suggest that related results can be derived for consistency in high-dimensional time series settings. Furthermore, for clarity, I focus on the basic forms of lasso and L_2-Boost, rather than also discussing variants of lasso such as elastic net or nonlinear forms of boosting.

4.1 Consistency

Both lasso and boosting are consistent for prediction in high-dimensional models, under a potentially reasonable assumption of sparsity in the true coefficients. However, consistency for model selection requires assumptions that are much more strict for both boosting and lasso, and are not likely to hold in most real-world situations.

A brief discussion of consistency is required before proceeding to the results, adapted from Zhao & Yu (2006). Let $f(X;\beta) = X\beta + \epsilon$ be a linear regression model parameterized by β. X is $n \times p$. An estimation procedure giving an estimator $\hat{\beta}$ is **consistent for prediction** if

$$f(X;\hat{\beta}) - f(X;\beta) \rightarrow_p 0, \text{ as } n \rightarrow \infty$$

An estimator $\hat{\beta}$ with true parameter β is **estimation consistent** if

$$\hat{\beta} - \beta \rightarrow_p 0, \text{ as } n \rightarrow \infty$$

A set of estimates is **consistent for model selection** if

$$P(\{i : \hat{\beta}_i \neq 0\} = \{i : \beta_i \neq 0\}) \rightarrow 1, \text{ as } n \rightarrow \infty$$

None of these definitions imply the other. A model estimation method can be prediction consistent, but not consistent in terms of model selection or parameter estimation, for example by substituting predictors outside the true model for predictors in the true model that are correlated with those outside the true model. A model estimation method that is consistent in terms of model selection may not be prediction consistent if the correct coefficients are selected but all with a constant bias, for example.

Our discussion below focuses on prediction consistency and a slightly stronger form of model selection consistency, sign consistency, but does not describe the properties of lasso and boosting in terms of parameter estimation consistency, since our results focus on the tradeoff between prediction and variable selection in high-dimensional macroeconomic forecasting and are not concerned with recovering the exact values of all parameters.

A note on the notation for sign consistency in the following sections:

$$\hat{\beta} =_s \beta$$

if and only if

$$sign(\hat{\beta}) - sign(\beta) \rightarrow_p 0, \text{ as } n \rightarrow \infty$$

where *sign* for a vector in $\mathbb{R}^p$ is a function returning a p-length vector containing the sign for each element of the input vector. This will be used to define a slightly stronger former of model selection consistency; rather than the estimated parameter $\hat{\beta}$ just setting the correct variables to zero, for the predictors in the true model, an estimator that is sign consistent also estimates the correct sign asymptotically.

4.1.1 Lasso Prediction Consistency

Bickel *et al.* (2009) derives a bound for lasso's prediction risk in high dimensional settings for non-random X.

Consider the linear model

$$y_i = X_i\beta + \epsilon_i, \qquad i = 1, \ldots, n \tag{4.1}$$

Let X be the $n \times p$ matrix of predetermined covariates, where $p > n$. Defining some additional notation is required. Let $M(\beta)$ be the sparsity (the number of non-zero coefficients) in a vector of coefficients β, and s be an upper bound on $M(\beta)$. Let J_0 be the indices of the nonzero coefficients of the true parameter β in the model.

Definition 4.1. The **Restricted Eigenvalue Condition** holds for $1 \leq s \leq p$ and a positive number c_0 for $\delta = \hat{\beta} - \beta$ when:

$$\kappa(s, c_0) = \min_{J_0 \subset 1,\ldots,p, |J_0| \leq s} \quad \min_{\delta \neq 0, ||\delta_{J_0^c}||_1 \leq c_0 ||\delta_{J_0}||_1} \frac{||X\delta||}{\sqrt{n}||\delta_{J_0}||_2} > 0 \tag{4.2}$$

This condition roughly means that the columns of X cannot be too correlated (Tibshirani, 2015).

Theorem 1 (Bickel *et al.* (2009))**.** *Let ϵ_i be i.i.d. $N(0, \sigma^2)$ random variables. Let the* *diagonal elements of $X'X/n$ be equal to 1, and let $M(\beta) \leq s$, where $1 \leq s \leq p$, $n \geq 1$, $p \geq 2$. Let the restricted eigenvalue condition be satisfied for $c_0 = 3$. Consider the lasso estimator $\hat{\beta}^{(Lasso)}$ with*

$$\lambda = A\sigma\sqrt{\frac{\log(p)}{n}}$$

and $A > 2\sqrt{2}$. Then, with probability at least $1 - p^{1-A^2/8}$,

$$||X(\hat{\beta} - \beta)||_2^2 \leq \frac{16A^2}{\kappa^2(s,3)}\sigma s \log p \tag{4.3}$$

This result gives the finite sample result for the prediction risk. Dividing the result by n, it is clear that, as long as s, the upper bound on the number of non-zero coefficients in β, grows sufficiently slowly with n, then lasso is consistent for prediction. Tibshirani (2015) unites this specific result and the related work by Greenshtein & Ritov (2004) and others in a basic asymptotic sense. For

$$\lambda = A\sigma\sqrt{\frac{\log(p)}{n}}$$

and under the assumption that the norm of the column vectors of X, $||X_j||_2^2 = n$, for $j = 1 \ldots p$ (which is trivial and can be achieved by normalizing X):

$$||X(\hat{\beta} - \beta)||_2^2/n = O_P\left(\sigma\sqrt{\frac{\log p}{n}}||\beta||_1\right) \tag{4.4}$$

As long as the l_1 norm of the true coefficients $||\beta||_1$ grows slower than $\sqrt{n/\log(p)}$ then lasso is consistent for prediction.

This result and the links to the various oracle inequalities derived by statisticians under different but related assumptions are presented in more detail in Bühlmann & van de Geer (2011).

4.1.2 Lasso Model Selection Consistency

Zhao & Yu (2006) show that lasso selects the correct model under restrictions on the sample covariance matrix and regression coefficients, known as the Irrepresentable Condition (IC). Meinshausen & Bühlmann (2006) have a similar result for random regressors under a related condition, known as the neighborhood stability condition. I describe Zhao & Yu (2006)'s results here. The IC requires that predictors that are not in the true model can't be represented by predictors that are in the true model.

Consider the linear model

$$y_i = X_i\beta + \epsilon_i, \qquad i = 1, \ldots, n \tag{4.5}$$

where ϵ_i are i.i.d. with mean 0 and variance σ^2. X_i is a p_n-dimensional vector of explanatory variables. β is a p_n dimensional vector of coefficients. p_n can grow as the sample size grows. Let $\mathbf{X}$ be the $n \times p_n$ matrix of explanatory variables.

Definition 4.2. Lasso is **Strongly Sign Consistent** if there is $\lambda_n = f(n)$ such that

$$\lim_{n \to \infty} P(\hat{\beta}(\lambda_n) =_s \beta) = 1$$

To define the Strong Irrepresentable Condition, which is a necessary condition for strong sign consistency for lasso, some additional notation is required. Let $\beta = (\beta_1, \ldots, \beta_{q_n}, \beta_{q_n+1}, \ldots, \beta_{p_n}$ where $\beta_j \neq 0$ for $j = 1, \ldots, q_n$ and $\beta_j = 0$ for $j = q_n + 1, \ldots, p_n$. Let $\mathbf{X}(1)$ and $\mathbf{X}(2)$ be the first q_n and the last $p_n - q_n$ columns of $\mathbf{X}$. Let $C_{11} = \frac{1}{n}\mathbf{X}(1)'\mathbf{X}(1)$, $C_{22} = \frac{1}{n}\mathbf{X}(2)'\mathbf{X}(2)$, $C_{12} = \frac{1}{n}\mathbf{X}(1)'\mathbf{X}(2)$, and $C_{21} = \frac{1}{n}\mathbf{X}(2)'\mathbf{X}(1)$.

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Definition 4.3. For the **Strong Irrepresentable Condition** to hold, there must exist a positive vector η such that

$$|C_{21}(C_{11})^{-1}sign(\beta_{(1)})| \leq \mathbf{1} - \eta$$

where $\mathbf{1}$ is a $p_n - q_n$ vector of 1's. The Strong Irrepresentable Condition is not something that can be verified in practice given it relies on knowing which covariates are in the true model; it can be interpreted, however, as a constraint on the regression coefficients of the irrelevant covariates when regressed on the relevant covariates. Zhao & Yu (2006) describe some constraints on the correlation structure of the covariates in a series of five corollaries that are sufficient for the Strong Irrepresentable Condition to hold. I present one below and leave the complete results for the reader to consider in the original paper.

Corollary 1 (Zhao & Yu (2006))**.** *Suppose β has q_n nonzero entries. C has 1s on the diagonal and the covariates have bounded correlation $|r_{ij}| \leq \frac{c}{2q-1}$ for $0 \leq c < 1$, then the Strong Irrepresentable Condition holds.*

So, for IC to hold, the correlation between covariates must be bounded, and this bound decreases as q_n, the density of the true model, increases. For large values of q_n the bound may be too small to be feasible in practice; so both sparsity and lack of collinearity are necessary for IC to hold, which is in turn necessary for lasso to be sign consistent for model selection. This result is relevant to the block-correlated correlation matrices that I will explore later in the applied section. In the simulations described, the theoretical tradeoff between q and the maximum correlation r will be made explicit.

There are 4 further assumptions necessary for the result. Assume there exists $0 \leq c_1 \leq c_2 \leq 1$ and $M_1, M_2, M_3, M_4 > 0$ so that:

1. $\frac{1}{n}(X_i)'(X_i) \leq M_1 \forall i$, which is trival since normalizing covariates can always achieve this.

2. $\alpha' C_{11} \alpha \geq M_2$, for $||\alpha||_2^2 = 1$, which is a condition on the eigenvalues for relevant covariates that ensures the inverse of C_{11} is well behaved.

3. $q_n = O(n^{c1})$, which is a sparseness condition.

4. $n^{\frac{1-c_2}{2}} \min_{i=1,\ldots,q_n} |\beta_i| \geq M_3$, a beta-min condition.

Theorem 2 (Zhao & Yu (2006))**.** *Consider the model (4.7) satisfying assumptions (1)-(4). Assume ϵ_i have a finite 2k'th moment $\mathbb{E}(\epsilon_i)^{2k} < \infty$. The Strong Irrepresentable Condition implies that lasso has strong sign consistency for $p_n = o(n^{(c_2-c_1)k})$. More specifically, for any λ that satisfies $\frac{\lambda}{\sqrt{n}} = o(n^{\frac{(c_2-c_1)}{2}})$ and $\frac{1}{p_n}(\frac{\lambda}{\sqrt{n}})^{2k} \to \infty$, then:*

$$P(\hat{\beta}(\lambda) =_s \beta) \geq 1 - O\left(\frac{p_n n^k}{\lambda^{2k}}\right) \to 1, \text{ as } n \to \infty$$

Under a beta-min condition, a restriction on correlation between the covariates (the strong IC), and some additional technical assumptions, lasso is strongly sign consistent. However, these are highly restrictive assumptions in practice for economic data. As the applied section will show, for large macroeconomic datasets, where data generating processes for individual series may be dense rather than sparse (see Giannone *et al.* (2017), and where covariates are block-correlated, lasso is not likely to be strongly sign consistent.

4.1.3 Boosting Prediction Consistency

Bühlmann (2006) proves that boosting is consistent for prediction in high-dimensional settings when X is predetermined. Let X_i be a p_n-dimensional vector for $i = 1, \ldots, n$

Consider the linear model

$$y_i = X_i\beta_n + \epsilon_i, \qquad i = 1, \ldots, n \tag{4.6}$$

where X is $n \times p$, $X_1, \ldots, X_n$ are i.i.d. with $E|X_j|^2 = 1$ for all $j = 1, \ldots, p_n$ and $E(\epsilon' X) = 0$ and $E(\epsilon) = 0$. The number of predictors p_n is allowed to grow with sample size n. The following assumptions are made:

1. The dimension of the predictor set satisfies $p_n = O(\exp(Cn^{1-\psi}))$, for $n \to \infty$, for

some $0 < \psi < 1, 0 < C < \infty$. This allows the predictor dimension to be large and grow with the sample size.

2. $sup_{n \in N} \sum_{j=1}^{p_n} |\beta_{j,n}| < \infty$. This is a sparseness condition. There can be many predictors that are relevant, but if so most must contribute with only small magnitudes.

3. $sup_{1 \leq j \leq p_n, n \in N} ||X_j||_\infty < \infty$, where $||X||_\infty = sup_{\omega \in \Omega} |X(\omega)|$ and Ω denotes the underlying probability space of the covariates.

4. $E|\epsilon|^s < \infty$ for some $s > 4/\psi$ with ψ from (1).

Theorem 3 (Bühlmann (2006))**.** *Consider the model (4.5) satisfying assumptions (1)-(4). Then, the boosting estimate $\hat{\phi}^{(m)}(\cdot)$ with the componentwise L_2-boost procedure from Section 3.3.2 satisfies; for some sequence $(m_n)_{n \in N}$ with $m_n \to \infty$ as $n \to \infty$ sufficiently slowly,*

$$\mathbb{E}_X |\hat{\phi}^{(m)}(X) - f_n(X)|^2 = o_p(1), \text{ as } n \to \infty \tag{4.7}$$

where X denotes a new predictor variable, independent of and with the same distribution as the X-component of the data $(X_i, y_i), i = 1, \ldots, n$.

4.1.4 Boosting Model Selection Consistency

It is still an open question whether or not L_2-Boost is consistent for model selection (Bühlmann & Hothorn, 2007). Ing & Lai (2011) study the properties of a variation of L_2-Boost, which they call the orthogonal greedy algorithm (OGA). This algorithm, similar to L_2-Boost, is a stepwise process that chooses the variable that is most correlated to the residual at each stage. Unlike L_2-Boost, to update the prediction function, rather than using a simple single variable regression, OGA sequentially orthogonalizes the selected variables. They prove that, under a beta-min condition, with probability approaching 1, the variables chosen by OGA contain all the relevant variables.

It is left to future work to determine if a similar proof is possible for L_2-Boost. The results of Ing & Lai (2011) suggest that some sort of beta-min condition would be necessary. However, it is unclear whether the strict limits on the correlation of the variables inside and outside the true generating process required for lasso to be consistent are also necessary for unmodified L_2-Boosting. The next section of theory describes the close relation that boosting has with L_1 penalization methods, which suggests that boosting also has some difficulties with variable selection in high-dimensional, collinear settings.

4.1.5 Discussion

I present results for the prediction risk for lasso in finite sample and asymptotically, while for boosting I just describe the asymptotic prediction risk. The expected prediction error of lasso in finite samples worsens under correlation of the covariates and decreasing sparsity. However, asymptotically, the prediction consistency of lasso depends only on the l_1 norm of the coefficients growing sufficiently slowly; this is the same as the main condition for L_2-Boost to be consistent for prediction. So, under large samples, I expect that L_2-Boost and lasso-family methods would perform similarly well, even in block-correlated and potentially dense macroeconomic data. Under finite samples, it is unclear whether or not boosting will have the same limitations as lasso. This motivates the next section that examines the theoretical connections between boosting and lasso to determine that similar limitations are likely for boosting.

For model selection consistency, the first issue is that in general a single regularization parameter cannot both be optimal for prediction and for model selection. The second, which was described more fully in section 4.1.3, is that lasso requires strong conditions on the correlation of covariates which are not likely to apply in block-correlated macroeconomic datasets. In the simulation and application sections, I will study more closely what kinds of mistakes lasso and boosting make in a block-correlated scenario and provide some suggestions for how some interpretation of the results is still possible. Another issue is that for the theorems presented, the proofs are based on non-random covariates X, whereas for economic data the covariates are likely to be random. However, the theoretical results for non-random X still provide insight into why there are limitations on boosting and lasso in certain real world scenarios.

4.2 Relationship between Lasso and L_2-Boost

In interpreting the results in the applied sections of the paper, it is informative to understand the similarities between lasso and L_2-Boost, and where these two methods diverge. In the previous section, I explored the consistency for prediction and model selection of lasso and boosting. Boosting does not have results available for finite sample prediction error or model selection consistency. Describing the theoretical links between the two methods can help clarify why performance differences in the two methods might exist in practice and also what can be expected for model selection consistency of boosting, where specific results are not available. I begin by describing LARS, which unified forward stagewise regression and lasso in the same framework, and then proceed to some more recent work unifying the method under the framework of subgradient optimization (Freund *et al.* , 2017) and as a solution to differential equations (Hastie *et al.* , 2007). For this section, the following algorithm is required, which was briefly introduced in Section 3.

Forward Stagewise Regression takes as input observations y_t and each of p observed predictors $x_t = (x_{t1}, ..., x_{tp})'$ for $t = 1, \ldots, T$. Let X be the $T \times p$ matrix of covariates.

1. $\hat{\phi}_0(x_t) = \bar{y}$
2. For $m = 1, \ldots, M$
    - For $t = 1, \ldots, T$, $u_t = y_t - \hat{\phi}_{m-1}(x_t)$; $\mathbf{u}$ is the T dimensional vector of residuals for the current step.

- Choose the single variable among the covariates with the highest correlation with the current residuals. $\hat{c} = X'u$ is a vector with entries from $\hat{c}_j$ from $j = 1, \ldots, p$ that are proportional to the current correlation between x_j and the residuals.

$$j_m = \arg\max |\hat{c}_j| \tag{4.8}$$

3. Update $\hat{\phi}_m(x_t) = \hat{\phi}_{m-1}(x_t) + \delta sign(\hat{c}_{j_m}) x_{tj}$, where δ is the stepsize.

This algorithm is the same as L_2-Boost, except the updates take a small step in the direction of the most correlated single variable regression using the sign of the coefficient of the single variable regression, rather than taking the step using the magnitude of the coefficient. In the scenario where $\delta = v|\hat{\beta}_{j_m}|$ then the two methods are equivalent.

4.2.1 LARS

Efron *et al.* (2004) showed that forward stagewise linear regression and lasso are both specific cases of a more general algorithm called LARS. Furthermore, LARS provides a computationally efficient way of computing the solution path for lasso as λ varies. I describe in a rough sense the LARS algorithm below[2]. LARS takes as input observations y_t and each of p observed predictors $x_t = (x_{t1}, ..., x_{tp})'$ for $t = 1, \ldots, T$. Let $\hat{\beta}$ be the LARS estimated coefficients. Start with all coefficients $\hat{\beta}$ equal to zero.

1. Let $\hat{u} = y - \bar{y}$

2. Find the predictor x_j most correlated with $\hat{u}$

3. Increase $\hat{\beta}_j$ and compute residuals $\hat{u} = y - \hat{\beta}X$ until another predictor x_k has as much correlation with $\hat{u}$ as x_j does

4. Increase $\hat{\beta}_j, \hat{\beta}_k$ in a direction that is in equiangular between the two predictors until a third variable enters the most correlated set.

[2] For the details on the exact algebra and implementation I refer the reader to the original paper (Efron *et al.* , 2004)

5. Increase all three coefficients equiangularly between the three variables until a fourth variable enters the active set, and so on, until all the predictors are in the active set and the correlation of the residuals with the predictors are zero.

Theorem 4 (Efron *et al.* (2004))**.** *Under the Lasso Modification, and assuming the "one at a time" condition*[3]*, the LARS algorithm yields all Lasso solutions.*

Theorem 5 (Efron *et al.* (2004))**.** *Under the Stagewise Modification, the LARS algorithm yields all Stagewise solutions.*

The lasso modification is a minor modification of the LARS procedure, while the stagewise modification is a moderate modification of the procedure. I first summarize the modifications at a high-level, adapted from (Hastie, 2003). A more explicit characterization is presented later in the section in Definition 4.5, while characterizing a modification of lasso that yields forward stagewise directly. In LARS, the active set (the indices of the coefficients that are currently being increased in a direction equilangular between them) can only monotonically increase. For the lasso modification, modify LARS so that if a coefficient ever crosses zero, drop it from the active set, recompute the equilangular distance between the current active set, and continue. For Stagewise regression, the authors consider an idealized procedure where the stepsize δ tends to zero. The Stagewise Modification proceeds as follows. During the LARS procedure, if the direction for any predictor j doesn't agree in sign with $corr(\hat{u}, x_j)$ then project the direction into the positive cone and use the projected direction instead. Under a restrictive condition on the covariate matrix, called the "positive cone condition", then Efron *et al.* (2004) show that lasso, forward stagewise, and LARS coefficient paths coincide.

4.2.2 Monotone Lasso

However, under most scenarios the paths of lasso and forward stagewise coefficients are very different, and the forward stagewise paths are much smoother than the lasso paths.

[3] This means there are no ties so that only one index added to the active set at each step

Hastie *et al.* (2007) derive a related result that links forward stagewise regression directly to lasso and decribes explicitly the difference between the two in terms of the optimization problem solved to derive the coefficient path. It characterizes the version of forward stagewise regression as a monotone version of lasso. First, it is necessary to introduce an expanded form of the $n \times p$ set of covariates X. $\tilde{X}$ includes each variable x_i and its negative $-x_i$.

The monotone lasso is defined on the expanded covariate space $\tilde{X}$ as the regular lasso problem, plus an additional constraint that the coefficient paths must be monotone non-decreasing. In the expanded coefficient space for both algorithms, the monotone lasso provides the same solution path for coefficients as the limiting version of forward stagewise regression. This leads to a succinct characterization of forward stagewise regression. Every point on the coefficient path of the regular version of lasso can be defined as the solution to a convex optimization problem. The monotone lasso, however, cannot be characterized as a convex optimization problem due to the monotonicity restriction, but the moves at each point in the coefficient path can still be characterized as locally optimal. A few definitions are presented before the main result. Let t be a continuous-valued variable that indexes the steps taken in the LARS algorithm defined in the previous section, where each step size is considered to be very small, tending to zero.

Definition 4.4. Assume $\beta(t)$ is a differentiable curve in $t \geq 0$, with $\beta(0) = 0$. The **$\mathbf{L_1}$ arc-length** of $\beta(t)$ in $[0, t]$ is:

$$TotalVariation(\beta, t) = \int_0^t \left|\left| \frac{\delta\beta(s)}{\delta s} \right|\right|_1 ds \tag{4.9}$$

This is a measure of smoothness of the curve of the coefficient path $\beta(t)$.

Definition 4.5. Let $\hat{\beta} \in \mathbb{R}^{2p}$ be an estimated coefficient for a linear model on the expanded variable set $\tilde{X}$ and let $\hat{u} = y - \tilde{X}\hat{\beta}$. Let A be the active set of variables achieving maximal correlation with $\hat{u}$.

1. The **lasso move direction** $\rho_l(\beta) : \mathbb{R}^{2p} \to \mathbb{R}^{2p}$ is:

$$\rho_l(\beta) = \begin{cases} 0 & if \tilde{X}'\hat{u} = 0 \\ \theta / \sum\limits_j \theta_j & otherwise, \end{cases}$$

   with $\theta_j = 0$ except for $j \in A$, where θ_A is the least squares coefficient of $\hat{u}$ on $\tilde{X}_A$.

2. The **monotone lasso move direction** $\rho_{lm}(\beta) : \mathbb{R}^{2p} \to \mathbb{R}^{2p}$ is:

$$\rho_{lm}(\beta) = \begin{cases} 0 & if \tilde{X}'\hat{u} = 0 \\ \theta / \sum\limits_j \theta_j & otherwise, \end{cases}$$

   with $\theta_j = 0$ except for $j \in A$, where θ_A is the non-negative least squares coefficient of $\hat{u}$ on $\tilde{X}_A$.

Though the monotone lasso can't be formulated as a solution to a global optimization problem, it can be formulated in terms of local optimality.

Theorem 6 (Hastie *et al.* (2007))**.** *The lasso and monotone lasso (forward stagewise) move directions defined in Definition 4.5 are optimal in the sense that:*

1. *The lasso move decreases the residual sum of squares at the optimal quadratic rate with respect to the L_1 coefficient norm;*

2. *The monotone-lasso move decreases the residual sum of squares at the optimal quadratic rate with respect to the coefficient L_1 arc-length.*

Furthermore, the coefficient paths for both methods can be defined by differential equations, the first for lasso and the second for monotone lasso/forward stagewise regression:

1. $\frac{\delta\beta}{\delta t} = \rho_l(\beta(t))$

2. $\frac{\delta\beta}{\delta t} = \rho_{lm}(\beta(t))$

with initial conditions $\beta(0) = 0$ *for both*

These results lead to an expectation that the boosting coefficient paths will be smoother than lasso coefficient paths. Lasso is known for having the tendency to switch back and forth between correlated variables depending on the regularization parameter and the data; boosting would be much less likely to do this given each step of boosting takes into account the smoothness of the coefficient path over previous iterations of the algorithm.

4.2.3 Subgradient Optimization

LARS is not the only framework that has united L_2-Boosting, forward stagewise regression, and lasso. Recent results in Freund *et al.* (2017) show that L_2-Boost can be formulated as a solution to a convex optimization problem that have the residuals as the optimization variable rather than the parameters. They show that L_2-Boost, forward stagewise regression and lasso can all be viewed as special instances of subgradient descent method of convex optimization applied to the following parameteric class of optimization problems:

$$P_\sigma : \min_u ||X'r||_\infty + \frac{1}{2\delta}||u - y||_2^2, \text{ where } u = y - X\beta \text{ for some } \beta$$

and where $\sigma \in (0, \infty]$ is a regularization parameter. The first term is the maximum correlation between the predictors and the residuals and the second is a regularization term that penalizes residuals that are far from the observations. This problem can be shown to be a dual of the lasso problem. They use this insight to form computational guarantees on the level of shrinkage and training error as boosting in the $p < N$ case converges to the least squares solution as the number of boosting iterations increase. For future work it would be interesting to study the impact of this insight on statistical guarantees of interest to econometricians, such as finite sample forms of consistency for boosting.

4.2.4 Discussion

The work unifying lasso and boosting under a common framework is important to understand how boosting and lasso might be different in practice. First, it is clear that lasso and forward stagewise regression are extremely similar. Under certain conditions, the coefficient path for both will be the same; in most scenarios, the path will be different, but the differences can be succinctly characterized as slight modifications of the unifying algorithm LARS. This leads us to expect that boosting and lasso, despite appearing very different, should have similar limitations in the prediction and variable selection tasks for macroeconomic data. Second, boosting enforces smoothness on coefficient paths as the regularization parameter varies; for lasso, correlated variables may switch in and out of the active set more often. This divergence will affect the algorithm's relative performance in variable selection under block-correlated simulations and macroeconomic data. Third, while lasso can be formulated as a convex optimization problem for each point in the coefficient path, the forward stagewise path's monotonicity restriction precludes such a characterization; however, formulating forward stagewise regression as locally optimal. This suggests that some results that are available for lasso may be much more difficult to derive for boosting. Many recent results on lasso relating to confidence intervals and standard errors (van de Geer *et al.* , 2014) are derived starting from the KKT conditions of the lasso optimization problem for the parameters.

5 Simulations

In this section, I investigate the model selection and prediction forecasting of lasso, elastic net, and boosting, for a high-dimensional model, with $p >> T$. I chose $T = 200$ since in many macroeconomic applications, there is only a few hundred data observations available, at most.

The data generating process, for $t = 1 \dots T$ and $T = 200$ is:

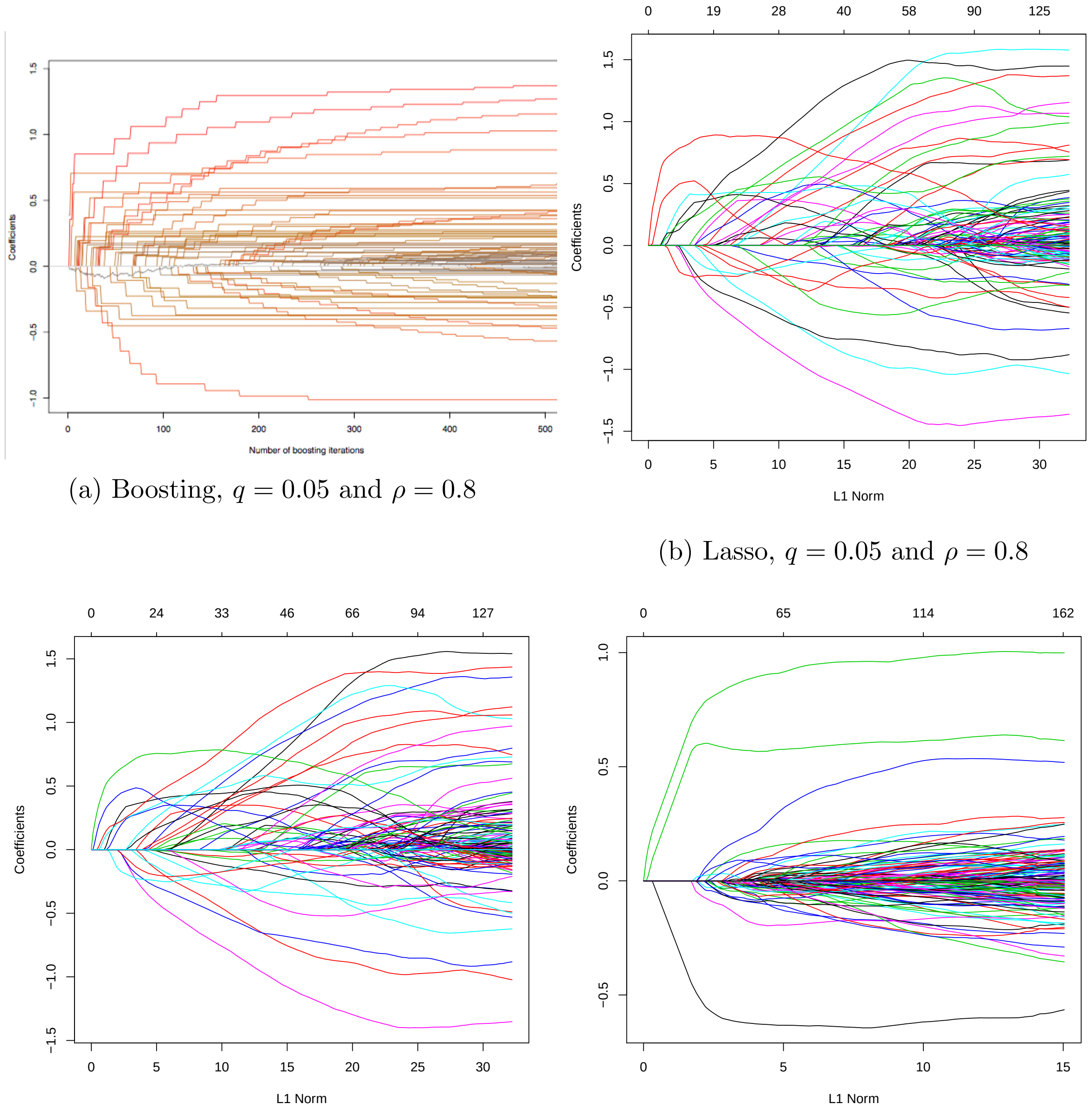
$$y_t = x_t\beta + \epsilon_t, \qquad \epsilon_t \sim N(0,1) \tag{5.1}$$

Let X be the $T \times p$ matrix of independent variables. There are $p = 900$ covariates in 30 blocks of 30 variables. x_t is i.i.d. and $x_t \sim N(0, \Sigma)$ where Σ is block-diagonal. Within each block variables are correlated at ρ and have variance 1. I simulate β by assuming that for each $n = 1, \ldots, 900$, β_n is 0 with probability $1 - q$ and 1 with probability q. If β_i is non-zero then $\beta_i \sim N(0.5, 1)$. Increasing q decreases sparsity of the data generating processes; increasing ρ increases correlation within blocks[4]. I will examine the results of varying both ρ and q on coefficient path, model selection and prediction error for both lasso and L_2-boost.

5.1 Coefficient Path under Regularization

The first feature of regularized regression that I examine is the coefficient path as the regularization parameter is varied. The results for simulations for $\rho = 0.8, q = 0.05$ for elastic net, lasso, and boosting are in Figure 2. Each line in the figure represents the magnitude of one of the 900 individual coefficients as the regularization parameter for the algorithm is relaxed. So, for boosting, the x-axis is the stopping parameter M as M increases, and for elastic net and lasso, the x-axis is λ as it is relaxed from high values to low values. With substantial collinearity between blocks of variables, even under significant sparsity the lasso coefficients are highly unstable. Many coefficients that are non-zero for high values of λ drop and then are zero for moderate values of λ, then non-zero and rising again for low values of λ. Using elastic net with $\alpha = 0.7$ does not alleviate this problem; elastic-net, although it is designed to deal with correlated blocks of variables, still shows a tendency to switch from one variable to another entirely as λ is varied. Boosting is much more stable. In L_2-Boost, once a variable is selected, its

[4] In real data, it is likely that ρ will not be constant across blocks; however, it is held constant in the simulations for clarity, given the results aren't materially different if ρ is varied across blocks.



(a) Boosting, $q = 0.05$ and $\rho = 0.8$

(b) Lasso, $q = 0.05$ and $\rho = 0.8$

(c) Elastic Net, $q = 0.05$ and $\rho = 0.8$

(d) Lasso, $q = 0.01$ and $\rho = 0.8$

Figure 2: Coefficient Paths Under Block Correlation

coefficient only monotonically increases. The final subfigure shows how lasso stability begins to improve even in the collinear setting with $r = 0.8$ within each 30-variable block when sparsity drops to $q = 0.01$.

I have demonstrated that for small fluctuations in the regularization parameter, variables selected by lasso and elastic net can fluctuate widely, while for boosting, coefficients monotonically increase as the regularization parameter is relaxed. This is expected given the theoretical results on the local optimization problem that boosting solves, which takes into account the smoothness of the coefficient path, compared to lasso, which is concerned only with the model's error and the L_1 norm of the estimated coefficients. In the rest of this section, I investigate whether the relative stability of the boosting coefficient path as the regularization parameter varies helps or hinders model selection and prediction.

5.2 Model Selection

According to the theoretical results on model selection consistency presented earlier, lasso has issues when sparsity doesn't hold and variables are correlated; boosting, given the close connections to lasso through LARS, is likely to have the same issues, although it is not clear if the conditions on boosting are more or less strict than on lasso. To test the theoretical predictions from Section 4, I present the model selection results for elastic net, lasso, and boosting for four levels levels of sparsity and collinearity in Table 1 for $p = 900$. For each combination of q and ρ, the first column gives the number of non-zero coefficients, the second shows the % of non-zero coefficients in the true model that are non-zero in the estimated model, and the third shows the % of zero coefficients in the true model that are non-zero in the estimated model. The first scenario is $q = 0.02, \rho = 0.3, p = 900$, which involves only moderate correlation and sparsity of approximately $T/10$. The second and fourth involve higher sparsity, approximately $T/20$ with the second having high correlation of 0.8 and the fourth having very high correlation of 0.95 within blocks. The third involves higher density of approximately $T/4$ and high correlation of 0.8.

With $p >> T$, for all the block-correlated scenarios examined, elastic net, lasso, and boosting all select far more variables than the true data-generating process, with the effect exacerbated as the data generating process becomes more dense (as q increases); boosting and elastic net both tend to select a similar number of non-zero variables that is higher than lasso. For $p = 900$, in all scenarios except the most dense scenario, the three high-

Table 1: Simulation Model Selection Results

	Non-Zero	% Correct	% Incorrect	Non-Zero	% Correct	% Incorrect
	1) q=0.02,$\rho = 0.3$, p=900			2) q=0.01,$\rho = 0.8$,p=900		
True	18	N/A	N/A	9	N/A	N/A
Lasso	82	86%	7.5%	44	79%	4.2%
ENet	90	86%	8.4%	51	79%	5.0%
Boost	90	86%	8.4%	49	79%	4.7%
	3) q=0.05,$\rho = 0.8$, p=900			4) q=0.01,$\rho = 0.95$, p=900		
True	45	N/A	N/A	9	N/A	N/A
Lasso	118	67%	10.3%	37	64%	3.5%
ENet	123	67%	10.9%	45	65%	4.4%
Boost	123	61%	11.3%	46	63%	4.5%

dimensional methods select the same percentage of correct variables, within 2 percentage points. For the dense scenario, boosting selects less correct variables and more incorrect variables than the other competing high-dimensional methods. The percentage of correct variables selected by all methods drops as density increases or as correlation increases. Between scenario 2 and 4, the density remains the same but correlation increases, and the percentage of non-zero variables correctly selected drops by 15 percentage points for all methods, from 80% to 65%. Between scenario 2 and 3, the correlation remains the same but the density of the data-generating process increases. The percentage of nonzero variables correctly selected drops by at least 12 percentage points for all three methods. This makes sense; the task of distinguishing between true non-zero coefficients and those correlated with true non-zero coefficients becomes increasingly difficult as correlation increases and as density increases, which corresponds to the theoretical guarantees on model selection consistency in lasso. For the percentage of zero coefficients incorrectly set to zero, some common patterns occur; in all scenarios but one, boosting selects more variables incorrectly than lasso and elastic net. This is likely due to the complications introduced by the L_1 arc length local optimization of boosting ; as M increases, once a variable is set to non-zero, boosting can't later drop that variable and set it to zero, whereas both lasso and elastic net, which ignore the smoothness of the coefficient path as λ varies, can make switches in the active set. So the instability of elastic net and lasso shown in the previous section may actually help improve model selection and prediction in correlated data as the regularization parameter is moved to an optimal value.

5.3 Selection within Blocks

Table 2: Estimated Non-Zero Coefficients for Block #2 when q=0.05,ρ=0.8

	True Model	Lasso	Elastic Net	Lin. Boost
$\hat{\beta}_{33}$	2.49	2.05	1.95	1.97
$\hat{\beta}_{34}$	-	0.21	0.24	0.38
$\hat{\beta}_{43}$	0.44	-		-
$\hat{\beta}_{46}$	-	0.06	0.09	-
$\hat{\beta}_{47}$	-	0.46	0.47	0.36
$\hat{\beta}_{50}$	-	-	-	-0.06
$\hat{\beta}_{55}$	-	0.03	0.04	-
$\hat{\beta}_{59}$	-	0.07	0.06	0.06
$\sum_{p=31}^{60} \hat{\beta}_p$	2.93	2.88	2.85	2.71

Given the macroeconomic data that I will examine in the next section is also dense and correlated in economically-interpretable blocks, it is interesting to examine what sort of mistakes boosting and lasso are making. Table 2 shows the results for the second block of thirty variables in a single run of the simulation with $q = 0.05, \rho = 0.8$, which had the worst results in terms of false positive rate. It describes how when making a mistake in model selection in the dense and correlated scenario, the high-dimensional models are not setting very many variables in blocks with no true non-zero coefficients to non-zero, but instead are setting incorrectly activating variables within the same block as true non-zero coefficients; however, the sum of the estimated coefficients for all coefficients is close to the sum of the true model for the block. For example, in this simulation, L_2-Boost sets variable 33 and 34 non-zero, such that the sum of the coefficients on both is approximately equal to the the true non-zero coefficient on variable 33. It set variable 47 non-zero instead of variable 43, and incorrectly sets 50 and 59 to non-zero (though their coefficients offset each other). The sum of the estimated coefficients for boosting in this block is 2.71, which is within 10% of the true sum of 2.93. To examine this finding in a more general scenario, Table 3, shows the average absolute value of the sum of the coefficients within a block for the true model for each of the four scenarios in the first column.

$$Sum_{block} = \frac{\sum_{b=1}^{30} \left| \sum_{i=(b\times 30-29)}^{b\times 30} (\beta_i) \right|}{30}$$

The latter three columns describe MSE_{block}, which is the average squared difference between the sum of the estimated and true coefficients for each block for a single iteration of a simulation for each of the four scenarios:

$$MSE_{block} = \frac{\sum_{b=1}^{30} \sum_{i=(b\times 30-29)}^{b\times 30} (\hat{\beta}_i - \beta_i)^2}{30}$$

Table 3: MSE of sum of coefficients at block level

	Avg block sum	Lasso MSE	Elastic Net MSE	Lin. Boost MSE
q=0.02, $\rho = 0.3$	0.53	0.015	0.013	0.012
q=0.01, $\rho = 0.8$	0.32	0.005	0.005	0.005
q=0.05, $\rho = 0.8$	0.97	0.022	0.023	0.025
q =0.01, $\rho = 0.95$	0.43	0.006	0.007	0.009

In all scenarios, the squared difference between the estimated sum of the parameters within each block is very small compared to the average absolute value of the sum of the true coefficients within a block. This is true in dense and correlated scenarios where the high-dimensional methods have a low rate of selecting the true variables correctly and a high rate of setting zero variables incorrectly. The mistakes that the methods are making in model selection appear to generally be within blocks, rather than across blocks. This has not yet been explored theoretically, and motivates investigating whether high-dimensional methods are consistent with respect to model selection at the block level, which I leave to future work, but investigate further empirically in Section 6.

The simulations have shown, as expected from the theoretical results on lasso and the connections between lasso and boosting, that model selection performance worsens as the block-correlated data generating process becomes more dense or correlated. However, I show that at a block-level, variable selection and parameter estimation improves, motivating grouping and interpreting variables in economically meaningful blocks when analyzing high-dimensional macroeconomic data with a block-correlated structure.

5.4 Prediction

Note that for the previous section on model selection, I selected the stopping parameters of lasso and boosting based on cross-validation, meaning the model selected was optimized for prediction error, not for model selection. Given that the same selected model can't be optimal for both prediction error and model selection, that the sample size was reasonably small so asymptotic results may not hold, and that the sparsity assumption that lasso and boosting rely on for consistency may not hold in some of the scenarios, it is no surprise that there were some issues in the previous simulations with the true variables selected. It is still valuable to examine the issues that arose given many economic practicioners select parameters based on cross-validation and would still like to know the limits of interpretation for high-dimensional models. Given the model estimation was done to be optimal for forecasting, it is interesting to also examine how the previous results correspond to out of sample forecasting performance of the three methods for the same simulated data generating processes.

I hold back the last 10% of the simulated data as a test set and get the average out of sample MSE over 1000 runs of the 200 sample data generating process described at the beginning of this section. The in sample MSE reported in the table corrsponds to the following equation, calculated on each of the three models for $S = 1000$:

$$M\hat{S}E^{(Model)} = \frac{\sum_{s=1}^{S}\sum_{t=1}^{180}(x_t\hat{\beta}_s^{(Model)} - y_t)^2}{180 \times S} \tag{5.2}$$

The out of sample MSE reported in the table corresponds to the following equation, which is calculated for each of the three models for $S = 1000$:

$$M\hat{S}E_{OOS}^{(Model)} = \frac{\sum_{s=1}^{S}\sum_{t=180}^{200}(x_t\hat{\beta}_s^{(Model)} - y_t)^2}{20 \times S} \tag{5.3}$$

where for both equations $\hat{\beta}_s^{Model}x_t$ is the result of running boosting, lasso, or elastic net on the training data from $t = 1, \dots, 180$ for Monte-Carlo simulation s with stopping parameter selected by cross-validation on those first 180 observations.

The results are described in Table 4. Lasso has the lowest out of sample MSE in every scenario, although for three out of four of the scenarios the performance of all three high-dimensional methods is quite close, which makes sense given their close theoretical links. The out of sample MSE for elastic net and boosting are generally comparable, except for the third scenario where the data generating process is dense and correlated. The in-sample MSE is generally comparable across the three methods, except in the third scenario again, where boosting struggles both in and out of sample and performs far worse than the other two competing methods. Unlike for model selection, only density has a negative effect on prediction performance. The difference between scenario 2) and 4), where sparsity remains the same and correlation increases from high to very high, is limited for all three methods, and the prediction performance actually improves in the very highly correlated case for every method. This is likely because the model selection mistakes within a block in the very highly correlated case have less of an effect on out of sample error than in the $\rho = 0.8$ case since variables are so similar. In Section 4, I showed that guarantees on model selection consistency required both restrictions on the correlation matrix of the predictors and sparsity. However, asymptotic guarantees for prediction consistency only required a bound on the sum of the true coefficients. This is evident in that increasing density from 2) to 3) results in a deterioration of out of sample performance for all three methods, especially boosting.

Table 4: Simulation Prediction Results

	MSE (Test)	MSE (Training)	MSE (Test)	MSE (Training)
	1) q=0.02, $\rho = 0.3$, p=900		2) q=0.01, $\rho = 0.8$, p=900	
Lasso	1.92	0.58	1.38	0.76
ENet	2.05	0.55	1.43	0.73
Boost	1.98	0.53	1.44	0.74
	3) q=0.05, $\rho = 0.8$, p=900		4) q=0.01, $\rho = 0.95$, p=900	
Lasso	4.29	0.54	1.29	0.81
ENet	4.48	0.54	1.32	0.79
Boost	7.28	1.25	1.38	0.80

How does the especially poor result for boosting in the dense and correlated case relate to the theory for model selection in Section 4 for lasso and boosting, which provided asymptotic guarantees on both that guaranteed prediction consistency? The finite sample results for lasso show that the order of the difference between the predicted and true value of y decreases with sample size. The results presented in Section 4 for boosting for prediction are asymptotic, but the same idea holds in finite samples. For the relatively small sample size of $T = 200$, the performance of the high-dimensional methods may not approach the asymptotic performance when $p >> T$. Furthermore, for different methods the rate of convergence to the asymptotic result may be different.

6 Application: U.S. Macroeconomic Analysis

The source data is comprised of 123 monthly stationary-transformed U.S. macroeconomic series from Jan. 1960, to July 2017 from the Fred-MD database described in McCracken & Ng (2016) and included in Appendix A. I transform each series according to the stationary transformation identified in the Appendix. I also add up to 3 lags of each monthly series, giving a total of 492 potential explanatory variables and an intercept.

Table 5: Groups in Fred-MD Dataset

	Name	Number of Series
Group 1	Output and income	16
Group 2	Labor market	31
Group 3	Housing	10
Group 4	Consumption, orders and inventories	7
Group 5	Money and credit	14
Group 6	Interest and exchange rates	21
Group 7	Prices	20
Group 8	Stock market	4

The dataset is divided into eight groups of similar variables. These groups are output and income, labor market, housing, consumption, orders and inventories, money and credit, interest and exchange rates, prices, and the stock market. The groups and the number of variables in each are described in Table 5. The correlation map of the series is in Figure 3. It is clear that the data displays block-correlated characteristics, with series within each group correlated with each other, sometimes strongly, and less correlated with series outside the group. The first small block in the bottom left is made up of various series of industrial production. The second strong block is made up various employment series. The third is formed from housing permit series. These three blocks combine to form a less strong, but still defined correlated block of real variables from the first three categories of Fred-MD. In the upper half of the correlation matrix, there are two small and strongly correlated blocks within the interest rate category, one for interest rate levels and one for spreads. There is a large block made up of price level series and a very small one for the stock category. It is clear that the correlation matrix displays the block-diagonal characteristics of the simulation that I explored in the previous section.

To reconcile the latest appendix of McCracken & Ng (2016) with the latest dataset posted on the website and to create a balanced panel, it is necessary to drop thirteen series. The seven ISM series listed in the appendix are no longer included in Fred-MD releases since June 2016. Furthermore, ACOGNO, ANDENOx, TWEXMMTH, UMCSENTx and VXOCLSx are all dropped due to missing data in order to create a balanced panel from

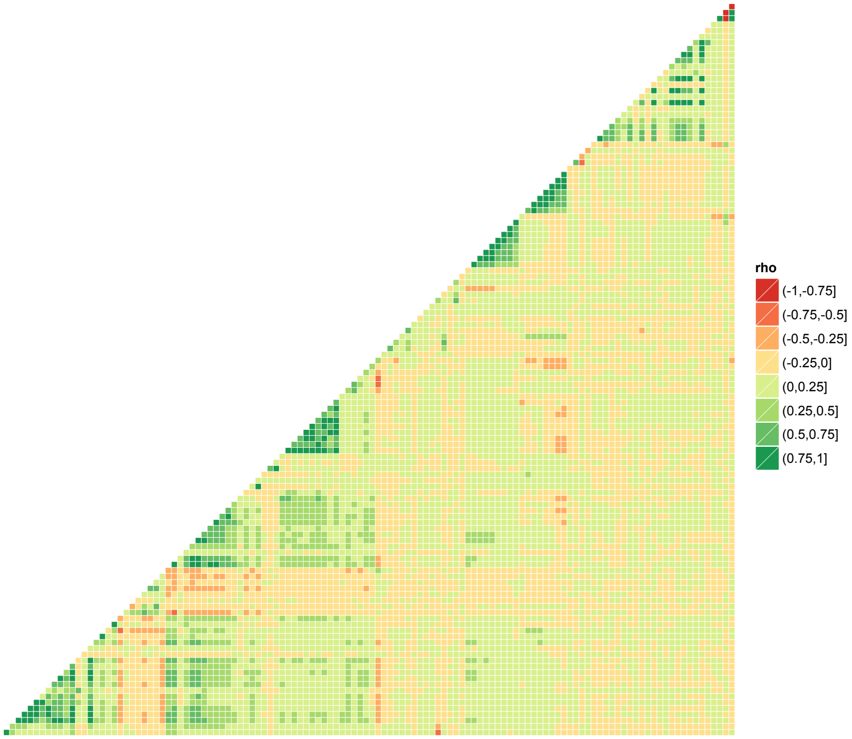


Figure 3: Correlation Heatmap for Fred-MD Data

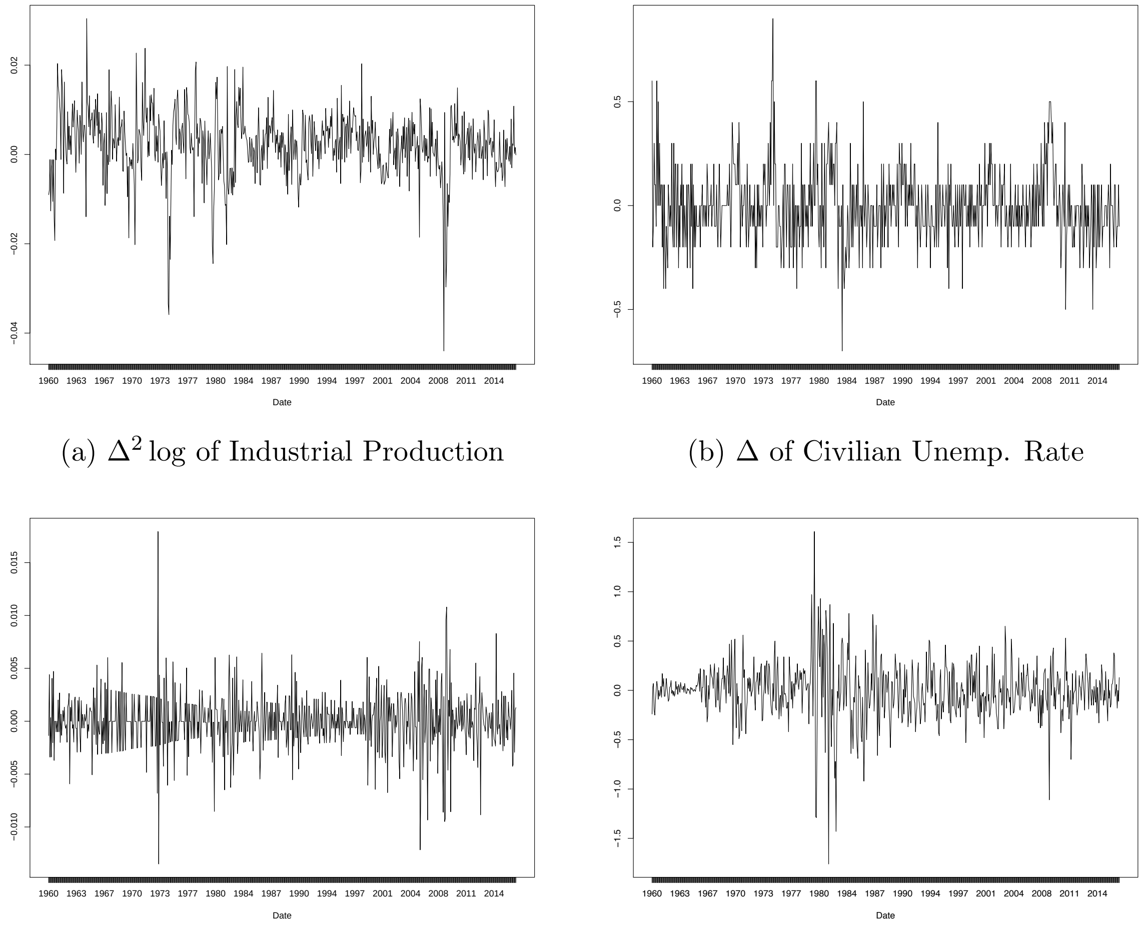
January 1960 to July 2017.

The four series that I focus on in the following section for prediction and variable selection are stationary-transformed industrial production (Group 1), civilian unemployment rate (Group 2), 10 year treasury rate (Group 6), and CPI (Group 7). All four are plotted from January 1960 to July 2017 in Figure 4.

6.1 Prediction

In this section, I describe the relative performance of lasso, L_2-boosting, regression tree boosting, and elastic net. I also illustrate a regression tree used in the non-linear tree boosting procedure and describe the resistance of overfitting that non-linear boosting displays compared to lasso.

The forecasting model used in this section is as follows. y_t is one of the four outcome series plotted in Figure 4 and x_t, the set of explanatory variables, are stationary-transformed



(a) Δ^2 log of Industrial Production

(b) Δ of Civilian Unemp. Rate

(c) Δ of 10 year Treasury Rate

(d) Δ^2 log of CPI (All Items)

Figure 4: Transformed Target Variables

variables calculated from the raw data provided in Fred-MD.

$$100y_{t+h} = \beta_{00} + \sum_{k=0}^{K} \alpha_k y_{t-k} + \sum_{i=1}^{V} \sum_{k=0}^{K} \beta_{ik} x_{i,(t-k)} + \epsilon_{t+h}, \qquad t = 1 \ldots T \tag{6.1}$$

with $K = 3$. The AR(1) and random walk baseline models are as follows:

$$100y_{t+h} = \beta_0 + \alpha y_t + \epsilon_{t+h} \tag{6.2}$$

where $\alpha = 1$ for the random walk model and is unrestricted for the AR(1) model.

I compute direct recursive forecasts for h =3 months, 6 months and 12 months for four series in Fred-MD for a model that includes lags up to and including $K = 3$. The Mean Squared Forecast Error (MSFE) is computed using one step ahead forecasts starting with the first third of the data as follows:

$$MSFE^{(model)} = \frac{\sum\limits_{t=1979:03}^{2017:07-h} (\hat{y}_{t+h} - y_{t+h})^2}{N},$$

where N is the number of one-step-ahead forecasts computed and $\hat{y}_{t+h}$ is the estimated direct h month ahead forecast from a model trained with regularization parameters selected by cross-validation on the subsample of the data from 1960:01 to month $t - 1$.

I presented the more general version of gradient boosting in Section 3. This is easily adapted to adding more complex base learners other than single variable regression or different loss functions other than L_2-loss. The most well-known versions of boosting use decision trees or regression trees for classification and prediction. Given in other domains the most successful application of boosting has been as a non-linear classifier and predictor, I have also included the results for boosting regression trees of depth 2, which allows for interaction terms between variables and captures non-linearities in the data generating process. The results for lasso, elastic net, L_2-boosting, tree boosting, and the random walk baseline are presented in Table 6. For all models except tree boosting, I cross validate at every step. For tree boosting, I choose the number of trees based on the first third of the data, since cross validating at every step was computationally infeasible. Figure 5 shows a single regression tree base learner that makes up part of the overall boosting classifier for the 1 month ahead forecast of the change in the unemployment rate. The split variables are named by their code in the FRED-MD appendix. The tree assigns the value for the outcome variable for an observation by splitting on the observation's value for total nonfarm employees and the 3 month lag of the spread between the 1 year T-Bill Rate and Fed Funds rate. For example, for observations that reach the split in the second level of the tree, if at that date the 1yr-Fed Funds spread is inverted, then the tree estimates an increase in the unemployment rate of 0.02; if the spread is not inverted, then the tree estimates an decrease in unemployment rate three months ahead of -0.03. This corresponds to previous evidence showing that inverted yield curves correspond to future real economic downturns.

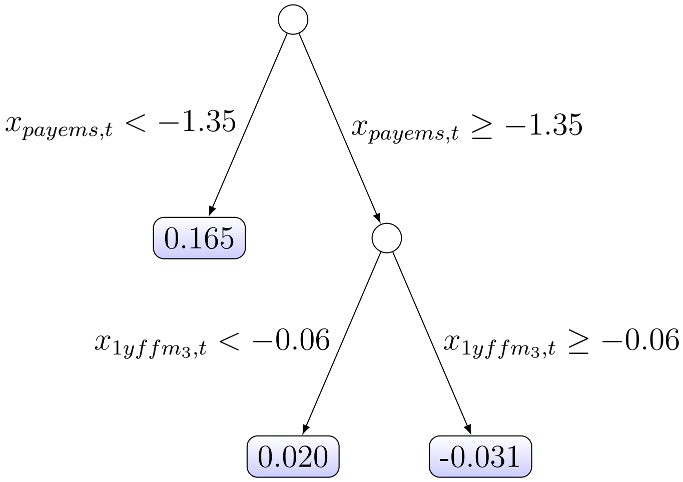
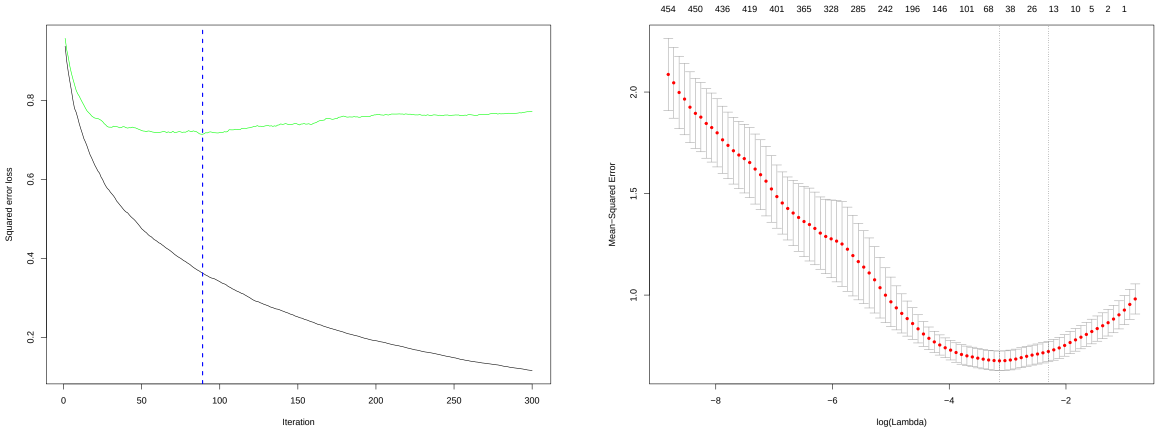


Figure 5: Boosted Regression Tree for 1-month ahead forecast of unemployment rate

Figure 6 shows the average out of sample error, in green, for tree boosting and in red for lasso. This is derived from 10-fold cross validation estimation of the parameters for the prediction function for 1 month ahead change in unemployment rate. For boosting, the in sample error, in blue, is also shown and the graph from left to right describes the change in the MSE as the regularization parameter is relaxed (M is increased). For lasso, the top axis also shows the number of non-zero coefficients, the MSE also includes standard error bars, and the graph from left to right describes the change in the MSE as the regularization parameter is strengthened (λ is increased). Boosting only a few trees results in poor performance; as the number of trees increases up to 100 the out of sample error and in sample error decreases. As too many trees are added, though, the estimated functions fits too much noise in the training sample and out of sample error worsens. As the number of trees approaches 300 the training data is fitted nearly perfectly. It is interesting though that tree boosting, compared to lasso, despite using a highly complex learner, does not overfit as significantly as the number of trees is increased. The out of sample increases past 100 but not steeply. For lasso, on the other hand, as λ is relaxed, overfitting occurs quickly, and the out of sample error increases sharply away from the optimal value of λ. This resistance to overfitting is known to be a property of boosting when using base learners of a certain complexity (such as trees) and is described in more detail in Schapire & Freund (2012).



(a) Tree Boosting

(b) Lasso

Figure 6: Out of Sample MSE as Regularization Parameter Varied

The forecast error results are presented in Table 6 as a percentage of the AR(1) error since that model is the better baseline method. The random walk baseline model is far worse than the AR model for every series at every forecast horizon. For the one month ahead direct forecast, the three linear methods for unemployment, industrial production, and for CPI outperform the AR model by a wide margin[5], by 15-24%. Tree boosting also outperforms significantly but with less of a margin than the linear methods. For the 3 and 6 month ahead direct forecast, the methods outperform the AR model by 1-5% for the unemployment rate and 10-year T-bill rate series. Boosting methods perform best at the 3 month time horizon. For the 6 month ahead industrial production, the AR model is the best model, by 6-20%. For the rest, the methods perform similarly to the AR model. Unlike in the simulation results, there is no linear method that consistently outperforms the other linear methods. For the horizons and series where the high-dimensional methods outperform, lasso is the best model twice, L_2-boosting is the best method 3 times, and elastic net is the best 4 times. The tree boosting method is the best method of the high-dimensional models in 5 scenarios, but in only two of those outperforms the AR model by more than 1 percent, and in two of those performs worse then the AR model. It is not surprising that tree boosting shows some tendency to overfit, given boosting regression trees of depth 2 allows a far more complex model than any of the linear models.

I performed a few robustness checks and alternative analyses, and report the qualitative results here. I checked adding variables with lags longer than 3 months and found that performance worsened for all of the high-dimensional methods due to overfitting. Including no lags of the variables also decreased performance; the information in a few lags does help forecasting even though it increases the dimensions of the predictors significantly. I also ran the forecasting exercise without cross-validation at every step, by choosing the regularization parameter purely from the first third of observations for each series for every estimation method. This is significantly less computationally intensive, but I found that cross-validating at every step improves performance by at least a few percent for the high-dimensional methods. Finally, I checked the results for several other

[5] For future work, it would be useful to test if this difference is statistically significant using a test like the Diebold-Mariano test (Diebold & Mariano, 2002). The original test, though, was not designed for comparing models that are nested and also not designed for comparing forecast errors derived from repeated out of sample errors (Diebold, 2012).

Table 6: MSFE vs. AR Baseline for Four U.S. Macroeconomic Series

	Industrial Prod.			Unemp. Rate		
	h=1	h=3	h = 6	h=1	h=3	h=6
Lasso	**0.819**	1.03	1.10	0.797	0.957	0.972
ENet	0.825	1.02	1.11	0.766	0.955	**0.968**
Lin. Boost	0.820	1.01	1.21	**0.765**	0.974	**0.968**
Tree Boost	0.859	**0.993**	**1.06**	0.785	**0.953**	0.970
No-Change	1.52	1.53	1.42	1.72	1.81	1.76
	CPI Total			10 Yr T-Bill		
	h=1	h=3	h = 6	h=1	h=3	h=6
Lasso	0.813	**0.997**	1.03	1.02	0.987	0.993
ENet	**0.805**	1.00	1.04	**0.992**	0.986	**0.988**
Lin. Boost	0.816	1.00	1.03	1.00	**0.974**	0.996
Tree Boost	0.949	**0.997**	**1.00**	1.09	0.992	1.00
No-Change	2.29	2.33	2.33	1.50	1.35	1.35

series not reported in detail here and found the results to be qualitatively similar to the results on the main series. I focus on the main four series reported in detail in the following section on variable selection.

6.2 Variable Selection

In this section I describe the variable selection results for L_2-boosting, lasso, and elastic net for the 1 month ahead forecast of the target series; I also examine the results from grouping coefficients based on economic intuition. I use the following model, which is the 1 month-ahead forecast model from Equation 3.1 with 3 additional lags of each independent variable included ($K = 3$), and each of the dependent and independent variables scaled to mean 0 and standard deviation 1. I choose to focus on the 1-month ahead forecast model since that is the time horizon where the machine learning methods most clearly outperform the AR(1) baseline.

$$\widetilde{y}_{t+1} = \sum_{k=0}^{K} \alpha_k \widetilde{y}_{t-k} + \sum_{i=1}^{V} \sum_{k=0}^{K} \beta_{ik} \widetilde{x}_{i,(t-k)} + \epsilon_{t+1}, \qquad t = 1 \ldots T \tag{6.3}$$

In Table 7, for the 1 month ahead industrial production, unemployment rate, inflation, and interest rate variable predicted in the previous subsection, I demonstrate which are the explanatory variables with the top 8 largest coefficients, for each of the linear estimation methods. Given the variables are all scaled for this section, the coefficient indicates for a standard deviation shift in the independent variables, which variables result in the largest forecasted shift, in standard deviations of the forecasted variable. In the first row for each of the sections of the table, I also include the number of non-zero variables for each of the methods. Apart from CPI, where boosting selects quite a few more variables than lasso and elastic net, the three methods select very similar numbers of variables for each forecasting function. For the 10Y T-Bill forecasting function, boosting actually selects the most parsimonious model, in contrast to the simulations where lasso always selected the most parsimonious model. More generally, for all four series, the identity, sign, and size of the coefficients selected for each of the three variables are very similar. For the 10Y T-Bill 1 month forecast, the three methods select the exact same variables, in the same order. Before examining boosting and lasso theoretically in section 4, this result would likely have been very surprising; on the surface, the optimization function of lasso and the stepwise procedure of boosting appear very different. However, in reality, both are forms of regularized regression.

Looking at some of the individual series, the results are sensical. 1 month ahead changes in industrial production is associated positively with positive measures of average hourly earnings in construction, other measures of industrial production, and reductions in unemployment. 1 month ahead changes in the unemployment rate is negatively associated with positive measures of employment and positively associated with increasing inversion in term spreads, which is a known leading indicator of recession. Acceleration in 1 month CPI is associated positively with growing money supply, expenditure and commodity proces, as expected, but shows some tendency to revert to the mean, with a negative association with current movements in CPI . There are, in addition, some variables where the signs do not hold with common intuition. For example, for the elastic net estimation

Table 7: Top 8 Variables and Coefficients for 1 month ahead forecast functions

	Lasso		**Elastic Net**		**L_2-Boosting**	
	Industrial Production					
NZ	41		47		48	
1.	Emp. Non-Dur.	0.111	Emp. Non-Dur.	0.109	AHE: Constr.	0.086
2.	S&P Div. Yield	-0.071	AHE: Constr.	0.077	Emp. Non-Dur.	0.085
3.	IP: Non-Dur.$_2$	0.069	S&P Div. Yield	-0.073	Inventory: Sales$_1$	-0.081
4.	AHE: Constr.	0.065	IP: Non. Dur.$_2$	0.069	S&P Div. Yield	-0.0741
5.	S&P Industr.$_2$	0.064	S&P Industr.$_2$	0.066	IP: Non-Dur.$_2$	0.068
6.	Initial Claims	-0.062	Initial Claims	-0.061	Emp: Goods-Prod.	0.067
7.	Inventory: Sales$_1$	-0.06	AAA Rate	0.052	S&P Industr.$_2$	0.063
8.	H. Permits Midwest	-0.046	3mT-FF	0.050	Initial Claims	-0.060
	Unemployment Rate					
NZ	57		60		57	
1.	Emp: TPU	-0.112	Emp: TPU	-0.109	Emp: Goods-Prod.	-0.179
2.	3mT-FF$_1$	-0.098	Unemp: < 5 wks	-0.095	Unemp Rate	-0.017
3.	Unemp: < 5wk	-0.095	Unemp Rate	-0.092	Unemp: < 5 wks	-0.0903
4.	Unemp Rate	-0.095	3mT-FF	-0.091	Emp: TPU	-0.089
5.	Initial Claims	0.089	Initial Claims	0.087	Initial Claims	0.082
6.	Emp: Constr.	-0.084	Emp: Constr$_1$	-0.082	Emp: Goods-Prod$_1$	-0.077
7.	Help-Wanted$_1$	-0.08	Help-Wanted$_1$	-0.079	Help-Wanted$_1$	-0.076
8.	Emp: Durables$_3$	0.072	3m CP -FF$_2$	0.067	3mT-FF$_3$	-0.075
	CPI: All Items					
NZ	70		68		89	
1.	CPI:All	-0.302	CPI:All	-0.284	CPI: All	-0.317
2.	Oil Px	0.157	Oil Px	0.149	Oil Px	0.166
3.	M2 Real	0.148	M2 Real	0.148	M2 Real	0.152
4.	CPI:Transp.$_1$	-0.108	CPI:Transp.$_1$	-0.108	CPI:Transp.$_1$	-0.144
5.	Dep. Reserves	-0.095	Dep. Reserves	-0.092	Dep. Reserves	-0.097
6.	Real PCE	0.081	Real PCE	0.079	Real PCE	0.086
7.	IP: Res. Utilities	0.060	IP: Res. Utilities	0.055	5Y Treas.	0.0773
8.	Nonrev. Credit	-0.060	Nonrev. Credit	-0.055	IP: Res. Utilities	0.073
	10Y T-Bill					
NZ	26		26		22	
1.	10Y T-Bill	0.214	10Y T-Bill.	0.207	10Y T-Bill	0.215
2.	S&P Div. Yld$_1$	-0.10	S&P Div. Yld$_1$	-0.098	S&P Div. Yld$_1$	-0.102
3.	1Y - FF$_1$	-0.090	1Y- FF$_1$	-0.080	1Y - FF$_1$	-0.102
4.	Inventory:Sales	-0.076	Inventory:Sales	-0.075	Inventory:Sales	-0.078
5.	10Y T-Bill$_1$	-0.071	10Y T-Bill$_1$	-0.067	10Y T-Bill$_1$	-0.072
6.	Retail Sales	0.0495	Retail Sales	0.0494	Retail Sales	0.053
7.	Emp: wholesale	0.0458	Emp: wholesale	0.0405	Emp: wholesale	0.044
8.	CPI: ex. Food	0.0429	CPI: ex. Food	0.0405	CPI: ex. Food	0.042

of 1 month ahead unemployment rate, the 2nd lag of 3m CP - FF spread coefficient is positive, while the 3m - FF spread coefficient is negative; this could be a result of the total effect for a set of correlated variables (term spreads) showing up partially in multiple coefficients, which I seek to address by interpreting in blocks.

The problem with the method above is that, as seen in the simulation, for high-dimensional methods many correlated variables have non-zero, but smaller coefficients. Simply ranking individual coefficients can miss groups of coefficients that move together and have a larger overall effect on the dependent variable. This motivates interpreting the variables within economically-meaningful blocks to reach stable conclusions on variable importance for boosting, lasso, and elastic net, as suggested in Li & Chen (2014). For each of the four variables in this section, I rank the top 3 categories of variables by the sum of the non-zero coefficients within each category. This is shown for lasso only given that the top three groups are identical for all three methods with nearly identical total coefficients. The results are closely associated with economic intuition. 1 month ahead industrial production is positively associated with increases in employment and output and negatively associated with measures of increases in financing costs. 1 month ahead changes in unemployment rate is negatively associated with increases in labor market, consumption and manufacturing measures and increases in term spreads. 1 month ahead acceleration in CPI shows reversion to the mean with strong negative sign on other price measures, and has a positive association with increases in output and increases in rates. 1 month ahead changes in the 10y rate is positively associated with previous changes in interest rates, with increases in employment and decreases in stock market yields.

Note that the method of aggregation used in this section is imperfect. Though I demonstrated that within most of the eight groups of variables there is strong positive correlations, that is not always the case; for example initial claims and the unemployment rate would be negatively correlated with measures that count total employment per industry. Simply adding the coefficients of all the employment variables together likely

Table 8: Sum of coefficients of top 3 groups selected for lasso

	Industrial Production		**Unemployment Rate**	
1.	Labor Market	0.25	Labor Market	-0.54
2.	Output and income	0.16	Consumption, orders, inventories	-0.124
3.	Interest and exchange rates	0.155	Interest and exchange rates	-0.120
	CPI: All		**10Y T-Bill**	
1.	Prices	-0.475	Stock. Market	-0.077
2.	Output and Income	0.178	Labor Market	0.076
3.	Interest and exchange rates	0.103	Interest Rates	0.071

underestimates the relation between the manually defined employment block and the outcome variable. This work, however, is meant to show the sensical results that come from interpreting high-dimensional methods between relatively independent blocks of variables, which is still accomplished using simple grouping and aggregation procedures. Examining which are the main individual coefficients that are positive before interpreting the sign of the coefficient on the groups also helps in interpreting when there is some complexity in correlations within a group. For future work, it would be fruitful to dicuss statistical procedures that would not rely on manually defined groups, but instead take into account correlation structure so that aggregation best captures the reduced form relationship between an outcome variable of interest and changes in unemployment. Various forms of unsupervised learning could be used in this capacity, for example.

I also run some alternative methods to check the variable selection results and show why high-dimensional methods like lasso and boosting that retain such interpretable results are advantageous compared to OLS or factor model alternatives. First, for each of the four series analyzed above, for the 1 month ahead forecast, I estimate a one step ahead forecast model

$$y_{t+h} = \beta x_t + \epsilon_t$$

using OLS, where the vector of predictors x_t includes only the top eight variables selected by lasso (which is nearly identical to the top 8 selected by elastic net and L_2-boosting) and check the significance level of each of these coefficients. The results are in Table 9,

Table 9: OLS coefficients for top 8 variables selected by lasso for 1m ahead forecast

Industrial Production		**Unemployment Rate**	
Emp. Non-Dur	0.26***	Emp: TPU	-0.25***
S&P Div. Yield	-0.12***	3mT-FF$_1$	-0.21***
IP: Non-Dur$_2$	0.14***	Unemp: $<$ 5wk	-0.13***
AHE: Constr.	0.13***	Unemp Rate	-0.10**
S&P Industr.$_2$	0.14 ***	Initial Claims	0.18***
Initial Claims	-0.15 ***	Emp: Constr.	-0.17***
Inventory: Sales	-0.11***	Help-Wanted$_1$	-0.13***
H. Permits Midwest	0.14***	Emp: Durables$_3$	-0.10***
R^2	0.33	R^2	0.33
CPI: All Items		**10Y T-Bill**	
CPI: All	-0.29***	10Y T-Bill	0.28***
Oil Px	0.20***	S&P Div. Yld$_1$	-0.16***
M2 Real	0.19***	1Y - FF$_1$	-0.16***
CPI:Transp.$_1$	-0.18***	Inventory:Sales	-0.12 ***
Dep. Reserves	-0.16***	10Y T-Bill$_1$	-0.16***
Real PCE	0.13***	Retail Sales	0.08
IP: Res. Utilities	-0.109***	Emp: wholesale	0.13***
Nonrev. Credit	-0.11***	CPI: ex. Food	0.10***
R^2	0.30	R^2	0.24
***Significant at a 1% level, ** Significant at a 5% level			

where variables are identified by the short names given in Table 7.

The top 8 variables selected by lasso are significant at a 1% level for all series, except retail sales for the 10Y T-Bill forecasting function and unemployment rate for the unemployment rate forecast, which is significant at a 5% level. Lasso, L_2-boost, and elastic net are selecting variables that correspond to those that have large and significant coefficients under a regular OLS regression, in one step without the testing of different model combinations that would be required to select from a set of high-dimensional covariates using regular OLS.

So far in this paper, we have ignored dynamic factor models, since from the estimation of a basic factor model it is difficult to interpret the direct effect of a single variable or group of variables on the outcome variable. An alternative method of testing the relationship between manually defined groups to a forecasted variable is running a model using OLS and the first principal component of each group in the FRED-MD dataset. The model is:

$$y_{t+h} = \alpha f_t + \epsilon_t$$

where f_t is a 8 dimensional and contains the first principal component of each of the groups of variables defined in the FRED-MD dataset. The results are in Table 10. For some of the series, the conclusions on which groups influence the forecasted variable by examining the significance level of the first principal component is similar to the conclusion obtained by examining the size of the lasso coefficients in groupings. For unemployment rate, the first principal components of labor market and interest and exchange rates are significant at a 1% level and also rank in the top 3 groups by size of coefficient in the grouping exercise for lasso. For 10Y rate, the stock market and interest rate components are significant at a 5% level and appear in the top 3 groupings from the lasso exercise. However, for CPI, none of the first principal components of the groups are significant in predicting 1 month ahead CPI, even though we know from the OLS regression ran in the previous table that there are many individual variables that are significant in the prediction function for CPI. The R^2 is close to zero for CPI and 10Y Rate, indicating that the first principal components of the groups are explaining very little variation in the forecasted variable.

The issue is that even within groups that relate strongly to the outcome variable, not all variables are relevant for predicting CPI. The variation of all variables in a group contributes to the first estimated factor. This could be improved by using a likelihood approach for constructing factors that takes into account the forecasted variable, rather than PCA, although this is out of scope of this paper. Furthermore, the variation that is relevant for predicting CPI may be contained not in the first, but in the second, or tenth principal component. But, including all of the principal components for every group quickly expands the covariate set, eventually resulting in the need for a high-dimensional

Table 10: OLS coefficients on first principal component of groups for 1m ahead forecast

	Ind. Prod.	**Unemp. Rate**	**CPI: All**	**10Y Rate**
Output and income	-0.06***	0.04***	-0.009	0.007
Labor market	-0.021	0.06***	0.004	-0.016
Housing	-0.013	0.00	-0.002	-0.003
Cons., Orders, Invent.	-0.022	-0.004	-0.002	-0.013
Money and credit	0.01	0.007	-0.031	0.005
Interest and exch. rates	-0.035***	0.032***	-0.006	0.024**
Prices	-0.004	-0.009	-0.016	-0.012
Stock Market	-0.104***	0.062***	0.008	-0.08***
R^2	0.27	0.26	0.01	0.05

method estimation method again, negating the dimension reduction provided by principal components in the first step. Furthermore, if more than the first principal component is included, there is not a clear interpretation if the third component of a group, for example, is significant in predicting CPI whereas the first two are not. This in contrast to one-step methods where the groups are derived from summation of existing coefficients and it is straightforward to drill down in groups and see which variables are non-zero within the group (even though we know that more detailed interpretation is muddled by correlation among the predictors). This comparative exercise indicates the difficulty with interpreting factor models, even one modified so that factors correspond to economically meaningful blocks.

The results for OLS on the top 8 variables selected by lasso and boosting shows that the methods are capable of selecting variables that explain a significant amount of variation in the forecasted variable, without multiple testing and search methods that are required to find the variables when running OLS only. The results for an interpretable factor model, where factors are estimated individually for each group in the in FRED-MD dataset, runs into issues where the dimension reduction occurs independently of the model estimation, so the first principal components of relevant groups of variables do not necessarily relate to the outcome variable. Including too many components would improve this but would eventually negate the dimension reduction provided by estimating the factors in groups in the first place. With lasso and boosting, methods that combine dimension reduction and estimation in one step, I was able to quickly identify which economically-meaningful blocks are relevant, with sensical results, with a post estimation summation of coefficients.

7 Conclusion

The statistical theory behind lasso has been studied extensively and the method is becoming increasingly familiar to economists. In this paper, I examine the main results for lasso on prediction consistency, which are strong, and model selection consistency, which rely on assumptions that are not likely to hold in most large macroeconomic datasets. Boosting is not as familiar to economists and statisticians, so there are still some results that are unavailable for L_2-Boost that are available for lasso, such as finite sample prediction risk and results for model selection consistency. Given the close ties between lasso and L_2-Boost examined through LARS in this paper, I would expect that for future work theoretical results for model selection consistency would be possible to derive for boosting. They would likely also rely on an even more strict condition on density and correlation of the covariates, along with a beta-min condition, given the simulation results presented in this paper, which showed how boosting's performance deteriorated more rapidly than lasso's as the density of the data generating process was increased.

I find that for prediction and variable selection performance, boosting is more sensitive to lack of sparsity in the generating process than lasso is for finite samples in simulations. Otherwise, in simulations and applications of data-generating processes with block-correlated predictors I find that the results for both are very similar, although lasso selects the smallest number of variables while maintaining prediction performance. I have examined the forecasting performance of high-dimensional linear methods for unemployment, real production, price level, and interest rate series at a variety of time horizons and have found robust evidence, especially at the 1 month time horizon, that high-dimensional linear methods outperform the AR baseline by a wide margin. The non-linear boosting method performs well, especially at longer time horizons, but only result in minor gains compared to the AR model for the horizons where it is the best model. There was no clear winner between the linear methods, with comparable results in all time horizons and with all four series examined.

Given the close linkage between lasso-type methods and L_2-Boost through LARS, this comparable performance is not surprising, despite how different the algorithms first appear. However, there are some differences that are important to consider. Boosting is much more modular than lasso; it is very easy to modify the stepwise algorithm based on different loss functions or more complex base learners. Incorporating regression trees, on the other hand for lasso, would be more difficult and require stepwise approximations to lasso. In this application I didn't find that the non-linear methods of boosting outperformed the linear methods significantly for forecasting, although there is some evidence of outperformance at the longer time horizons. It was encouraging that there wasn't much evidence of overfitting even in the 1 month forecasts where linear methods performed well, despite the complexity of regression trees as base learners compared to single variable regressions. Lasso, on the other hand, as illustrated in Section 6.1, is more prone to overfitting as the regularization parameter is relaxed. In other economic applications with data generating processes that take complex non-linear forms, tree boosting may outperform further. Given the performance of lasso and L_2-Boost are similar in the application to real data, and lasso is more parsimonious and generally more accurate in the simulated linear data, these results do not make a compelling case for economists to use L_2-Boosting instead of lasso for high-dimensional linear models. Instead, I have used the consistency results available and the linkages to lasso for L_2-boost to introduce general gradient boosting as a methodology; the case for using more complex non-linear methods of boosting is stronger once the linear case is understood from the perspective of an econometrician.

Given the likely violations of the assumptions ensuring model consistency for all three methods, interpreting coefficients on individual variables is difficult. Individual variable selection, though, is suprisingly similar across the lasso, boosting and elastic net for each of the four forecasting functions examined, and the top coefficients have the identity and sign that would be expected (for example, employment variables are the best leading indicator for changes to the unemployment rate). I suggest that grouping variables in economically meaningful groups can make interpretation more robust. I show that sensical and interpretable group coefficients result when aggregating the individual coefficients of the forecasting functions for four U.S. macroeconomic series. Two further avenues are suggested by the results presented here. First, it would be interesting to develop a formal proof showing that if variables are tightly clustered in correlated blocks, the high-dimensional models are block-consistent for model selection, meaning the total effect of the block of variables is estimated correctly, even if individual coefficients within a block are not according to the true model. Second, it would be useful to develop interpretable grouping methods that take into account the complexity of correlation structure in real data, where clean blocks as in the simulation are not likely to occur.

This is the first time that the difficulties with the block-correlated nature of high-dimensional macroeconomic time series has been investigated closely from the perspective of regularization techniques for estimation. This is also the first time that the forecasting and variable selection performance of boosting and lasso has been compared directly for economic data both theoretically and in an applied contex. I find that model selection issues due to correlation are best addressed by grouping variables when interpreting coefficients. I motivate additional work on basic theory for boosting, on formulating a notion of block-consistency for high-dimensional methods dealing with block-correlated data, and deriving interpretable statistical grouping methods rather than relying on manual groups.

References

Athey, Susan, & Imbens, Guido W. 2015. Machine learning methods for estimating heterogeneous causal effects. *Stat*, **1050**(5).

Bai, Jushan, & Ng, Serena. 2009. Boosting diffusion indices. *Journal of Applied Econometrics*, **24**(4), 607–629.

Basu, Sumanta, Michailidis, George, *et al.* . 2015. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, **43**(4), 1535–1567.

Bickel, Peter J, Ritov, Yaacov, Tsybakov, Alexandre B, *et al.* . 2009. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, **37**(4), 1705–1732.

Buchen, Teresa, & Wohlrabe, Klaus. 2011. Forecasting with many predictors: Is boosting a viable alternative? *Economics Letters*, **113**(1), 16–18.

Bühlmann, Peter. 2006. Boosting for high-dimensional linear models. *The Annals of Statistics*, **34**(2), 559–583.

Bühlmann, Peter, & Hothorn, Torsten. 2007. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 477–505.

Bühlmann, Peter, & van de Geer, Sara. 2011. *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media.

Callot, Laurent AF, & Kock, Anders B. 2014. Oracle efficient estimation and forecasting with the adaptive lasso and the adaptive group lasso in vector autoregressions. *Essays in Nonlinear Time Series Econometrics*, 238–268.

Diebold, Francis X. 2012 (September). *Comparing Predictive Accuracy, Twenty Years Later: A Personal Perspective on the Use and Abuse of Diebold-Mariano Tests.* Working Paper 18391. National Bureau of Economic Research.

Diebold, Francis X, & Mariano, Robert S. 2002. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, **20**(1), 134–144.

Doornik, Jurgen A, & Hendry, David F. 2015. Statistical model selection with Big Data. *Cogent Economics & Finance*, **3**(1), 1045216.

Dopke, Jorg, Fritsche, Ulrich, & Pierdzioch, Christian. 2017. Predicting recessions with boosted regression trees. *International Journal of Forecasting*, **33**(4), 745 – 759.

Efron, Bradley, Hastie, Trevor, Johnstone, Iain, Tibshirani, Robert, *et al.* . 2004. Least angle regression. *The Annals of Statistics*, **32**(2), 407–499.

Freund, Robert M, Grigas, Paul, Mazumder, Rahul, *et al.* . 2017. A new perspective on boosting in linear regression via subgradient optimization and relatives. *The Annals of Statistics*, **45**(6), 2328–2364.

Freund, Yoav, Schapire, Robert E, *et al.* . 1996. Experiments with a new boosting algorithm. *Pages 148–156 of: Machine Learning: Proceedings of the Thirteenth International Conference.*

Friedman, Jerome, Hastie, Trevor, & Tibshirani, Rob. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**(1), 1.

Friedman, Jerome H. 2001. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 1189–1232.

Giannone, Domenico, Lenza, Michele, & Primiceri, Giorgio. 2017. *Economic Predictions with Big Data: The Illusion Of Sparsity.* Tech. rept. CEPR Discussion Papers.

Greenshtein, Eitan, & Ritov, Ya'Acov. 2004. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, **10**(6), 971–988.

Hastie, Trevor. 2003 (March). *Lecture on Least Angle Regression, Forward Stagewise, and the Lasso.* Stanford University Statistics Department.

Hastie, Trevor, Taylor, Jonathan, Tibshirani, Robert, & Walther, Guenther. 2007. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, **1**, 1–29.

Hepp, Tobias, Schmid, Matthias, Gefeller, Olaf, Waldmann, Elisabeth, & Mayr, Andreas. 2016. Approaches to regularized regression–a comparison between gradient boosting and the lasso. *Methods of Information in Medicine*, **55**(05), 422–430.

Hothorn, T, Bühlmann, P, Kneib, T, Schmid, M, Hofner, B, Sobotka, F, & Scheipl, F. 2012. mboost: Model-Based Boosting. R package version 2.1-2. *URL: http://CRAN. R-project. org/package= mboost.*

Ing, Ching-Kang, & Lai, Tze Leung. 2011. A stepwise regression method and consistent model selection for high-dimensional sparse linear models. *Statistica Sinica*, 1473–1513.

Kim, Hyun Hak, & Swanson, Norman. 2011. *Forecasting Financial and Macroeconomic Variables Using Data Reduction Methods: New Empirical Evidence.* Departmental Working Papers. Rutgers University, Department of Economics.

Kock, Anders Bredahl, & Callot, Laurent. 2015. Oracle inequalities for high dimensional vector autoregressions. *Journal of Econometrics*, **186**(2), 325–344.

Kohavi, Ron, *et al.* . 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Pages 338–345 of: Proc. 14th Int. Joint Conf. Artificial Intelligence.*

Lehmann, R., & Wohlrabe, K. 2016. Looking into the black box of boosting: the case of Germany. *Applied Economics Letters*, **23**(17), 1229–1233.

Lehmann, Robert, & Wohlrabe, Klaus. 2017. Boosting and regional economic forecasting: the case of Germany. *Letters in Spatial and Resource Sciences*, **10**(2), 161–175.

Li, Jiahan, & Chen, Weiye. 2014. Forecasting macroeconomic time series: LASSO-based approaches and their forecast combinations with dynamic factor models. *International Journal of Forecasting*, **30**(4), 996 – 1015.

McCracken, Michael W, & Ng, Serena. 2016. FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, **34**(4), 574–589.

Meinshausen, Nicolai, & Bühlmann, Peter. 2006. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 1436–1462.

Moench, Emanuel, Ng, Serena, & Potter, Simon. 2013. Dynamic hierarchical factor models. *Review of Economics and Statistics*, **95**(5), 1811–1817.

Ng, Serena. 2013. Variable Selection in Predictive Regressions. *Handbook of economic forecasting*, **2**, 752–789.

Ng, Serena. 2014. Boosting recessions. *Canadian Journal of Economics*, **47**(1), 1–34.

Ridgeway, Greg. 2007. Generalized Boosted Models: A guide to the gbm package. *R package version*, **1**(1), 2007.

Robinzonov, Nikolay, Tutz, Gerhard, & Hothorn, Torsten. 2012. Boosting techniques for nonlinear time series models. *Advances in Statistical Analysis*, **96**(1), 99–122.

Schapire, Robert E, & Freund, Yoav. 2012. *Boosting: Foundations and algorithms*. MIT press.

Shalizi, Cosma. 2006 (October). *Lecture Notes on Regression Trees*. Carnegie Mellon University.

Stock, James H, & Watson, Mark W. 2002a. Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, **97**(460), 1167–1179.

Stock, James H, & Watson, Mark W. 2002b. Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, **20**(2), 147–162.

Stone, Mervyn. 1977. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44–47.

Taieb, Souhaib Ben, Hyndman, Rob J, *et al.* . 2014. Boosting multi-step autoregressive forecasts. *Pages 109–117 of: Proc. 31st Int. Conf. Mach. Learning.*

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Tibshirani, Ryan. 2015 (Spring). *Lecture Notes on Sparsity and the Lasso*. Carnegie Mellon University.

van de Geer, Sara, Bühlmann, Peter, Ritov, Yaacov, Dezeure, Ruben, *et al.* . 2014. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, **42**(3), 1166–1202.

Wang, Hansheng, & Leng, Chenlei. 2008. A note on adaptive group lasso. *Computational statistics & data analysis*, **52**(12), 5277–5286.

Weston, Steve. 2014. doParallel: Foreach parallel adaptor for the parallel package. *R package version*, **1**(8).

Wohlrabe, Klaus, & Buchen, Teresa. 2014. Assessing the macroeconomic forecasting performance of boosting: evidence for the United States, the Euro area and Germany. *Journal of Forecasting*, **33**(4), 231–242.

Yang, Yuhong. 2005. Can the strengths of AIC and BIC be shared? A conflict between model indentification and regression estimation. *Biometrika*, **92**(4), 937–950.

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

Zhao, Peng, & Yu, Bin. 2006. On model selection consistency of Lasso. *Journal of Machine Learning Research*, **7**, 2541–2563.

Zou, Hui, & Hastie, Trevor. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**(2), 301–320.

Appendix A Fred-MD Data Appendix

The column TCODE denotes the following data transformation for a series x: (1) no transformation; (2) Δx_t; (3) $\Delta^2 x_t$; (4) $log(x_t)$; (5) $\Delta \log(x_t)$; (6) $\Delta^2 \log(x_t)$; (7) $\Delta(x_t/x_{t-1} - 1.0)$. The FRED column gives mnemonics in FRED followed by a short description. The comparable series in Global Insight is given in the column GSI.

Some series require adjustments to the raw data available in FRED. We tag these variables with an asterisk to indicate that they been adjusted and thus differ from the series from the source. A summary of the adjustments is detailed in the paper https://research.stlouisfed.org/wp/2015/2015-012.pdf.

Group 1: Output and income

	id	tcode	fred	description	gsi	gsi:description
1	1	5	RPI	Real Personal Income	M_14386177	PI
2	2	5	W875RX1	Real personal income ex transfer receipts	M_145256755	PI less transfers
3	6	5	INDPRO	IP Index	M_116460980	IP: total
4	7	5	IPFPNSS	IP: Final Products and Nonindustrial Supplies	M_116460981	IP: products
5	8	5	IPFINAL	IP: Final Products (Market Group)	M_116461268	IP: final prod
6	9	5	IPCONGD	IP: Consumer Goods	M_116460982	IP: cons gds
7	10	5	IPDCONGD	IP: Durable Consumer Goods	M_116460983	IP: cons dble
8	11	5	IPNCONGD	IP: Nondurable Consumer Goods	M_116460988	IP: cons nondble
9	12	5	IPBUSEQ	IP: Business Equipment	M_116460995	IP: bus eqpt
10	13	5	IPMAT	IP: Materials	M_116461002	IP: matls
11	14	5	IPDMAT	IP: Durable Materials	M_116461004	IP: dble matls
12	15	5	IPNMAT	IP: Nondurable Materials	M_116461008	IP: nondble matls
13	16	5	IPMANSICS	IP: Manufacturing (SIC)	M_116461013	IP: mfg
14	17	5	IPB51222s	IP: Residential Utilities	M_116461276	IP: res util
15	18	5	IPFUELS	IP: Fuels	M_116461275	IP: fuels
16	19	1	NAPMPI	ISM Manufacturing: Production Index	M_110157212	NAPM prodn
17	20	2	CUMFNS	Capacity Utilization: Manufacturing	M_116461602	Cap util

Group 2: Labor market

	id	tcode	fred	description	gsi	gsi:description
1	21*	2	HWI	Help-Wanted Index for United States		Help wanted indx
2	22*	2	HWIURATIO	Ratio of Help Wanted/No. Unemployed	M_110156531	Help wanted/unemp
3	23	5	CLF16OV	Civilian Labor Force	M_110156467	Emp CPS total
4	24	5	CE16OV	Civilian Employment	M_110156498	Emp CPS nonag
5	25	2	UNRATE	Civilian Unemployment Rate	M_110156541	U: all
6	26	2	UEMPMEAN	Average Duration of Unemployment (Weeks)	M_110156528	U: mean duration
7	27	5	UEMPLT5	Civilians Unemployed - Less Than 5 Weeks	M_110156527	U < 5 wks
8	28	5	UEMP5TO14	Civilians Unemployed for 5-14 Weeks	M_110156523	U 5-14 wks
9	29	5	UEMP15OV	Civilians Unemployed - 15 Weeks & Over	M_110156524	U 15+ wks
10	30	5	UEMP15T26	Civilians Unemployed for 15-26 Weeks	M_110156525	U 15-26 wks
11	31	5	UEMP27OV	Civilians Unemployed for 27 Weeks and Over	M_110156526	U 27+ wks
12	32*	5	CLAIMSx	Initial Claims	M_15186204	UI claims
13	33	5	PAYEMS	All Employees: Total nonfarm	M_123109146	Emp: total
14	34	5	USGOOD	All Employees: Goods-Producing Industries	M_123109172	Emp: gds prod
15	35	5	CES1021000001	All Employees: Mining and Logging: Mining	M_123109244	Emp: mining
16	36	5	USCONS	All Employees: Construction	M_123109331	Emp: const
17	37	5	MANEMP	All Employees: Manufacturing	M_123109542	Emp: mfg
18	38	5	DMANEMP	All Employees: Durable goods	M_123109573	Emp: dble gds
19	39	5	NDMANEMP	All Employees: Nondurable goods	M_123110741	Emp: nondbles
20	40	5	SRVPRD	All Employees: Service-Providing Industries	M_123109193	Emp: services
21	41	5	USTPU	All Employees: Trade, Transportation & Utilities	M_123111543	Emp: TTU
22	42	5	USWTRADE	All Employees: Wholesale Trade	M_123111563	Emp: wholesale
23	43	5	USTRADE	All Employees: Retail Trade	M_123111867	Emp: retail
24	44	5	USFIRE	All Employees: Financial Activities	M_123112777	Emp: FIRE
25	45	5	USGOVT	All Employees: Government	M_123114411	Emp: Govt
26	46	1	CES0600000007	Avg Weekly Hours : Goods-Producing	M_140687274	Avg hrs
27	47	2	AWOTMAN	Avg Weekly Overtime Hours : Manufacturing	M_123109554	Overtime: mfg
28	48	1	AWHMAN	Avg Weekly Hours : Manufacturing	M_14386098	Avg hrs: mfg
29	49	1	NAPMEI	ISM Manufacturing: Employment Index	M_110157206	NAPM empl
30	127	6	CES0600000008	Avg Hourly Earnings : Goods-Producing	M_123109182	AHE: goods
31	128	6	CES2000000008	Avg Hourly Earnings : Construction	M_123109341	AHE: const
32	129	6	CES3000000008	Avg Hourly Earnings : Manufacturing	M_123109552	AHE: mfg

Group 3: Housing

	id	tcode	fred	description	gsi	gsi:description
1	50	4	HOUST	Housing Starts: Total New Privately Owned	M_110155536	Starts: nonfarm
2	51	4	HOUSTNE	Housing Starts, Northeast	M_110155538	Starts: NE
3	52	4	HOUSTMW	Housing Starts, Midwest	M_110155537	Starts: MW
4	53	4	HOUSTS	Housing Starts, South	M_110155543	Starts: South
5	54	4	HOUSTW	Housing Starts, West	M_110155544	Starts: West
6	55	4	PERMIT	New Private Housing Permits (SAAR)	M_110155532	BP: total
7	56	4	PERMITNE	New Private Housing Permits, Northeast (SAAR)	M_110155531	BP: NE
8	57	4	PERMITMW	New Private Housing Permits, Midwest (SAAR)	M_110155530	BP: MW
9	58	4	PERMITS	New Private Housing Permits, South (SAAR)	M_110155533	BP: South
10	59	4	PERMITW	New Private Housing Permits, West (SAAR)	M_110155534	BP: West

Group 4: Consumption, orders, and inventories

	id	tcode	fred	description	gsi	gsi:description
1	3	5	DPCERA3M086SBEA	Real personal consumption expenditures	M_123008274	Real Consumption
2	4*	5	CMRMTSPLx	Real Manu. and Trade Industries Sales	M_110156998	M&T sales
3	5*	5	RETAILx	Retail and Food Services Sales	M_130439509	Retail sales
4	60	1	NAPM	ISM : PMI Composite Index	M_110157208	PMI
5	61	1	NAPMNOI	ISM : New Orders Index	M_110157210	NAPM new ordrs
6	62	1	NAPMSDI	ISM : Supplier Deliveries Index	M_110157205	NAPM vendor del
7	63	1	NAPMII	ISM : Inventories Index	M_110157211	NAPM Invent
8	64	5	ACOGNO	New Orders for Consumer Goods	M_14385863	Orders: cons gds
9	65*	5	AMDMNOx	New Orders for Durable Goods	M_14386110	Orders: dble gds
10	66*	5	ANDENOx	New Orders for Nondefense Capital Goods	M_178554409	Orders: cap gds
11	67*	5	AMDMUOx	Unfilled Orders for Durable Goods	M_14385946	Unf orders: dble
12	68*	5	BUSINVx	Total Business Inventories	M_15192014	M&T invent
13	69*	2	ISRATIOx	Total Business: Inventories to Sales Ratio	M_15191529	M&T invent/sales
14	130*	2	UMCSENTx	Consumer Sentiment Index	hhsntn	Consumer expect

Group 5: Money and credit

	id	tcode	fred	description	gsi	gsi:description
1	70	6	M1SL	M1 Money Stock	M_110154984	M1
2	71	6	M2SL	M2 Money Stock	M_110154985	M2
3	72	5	M2REAL	Real M2 Money Stock	M_110154985	M2 (real)
4	73	6	AMBSL	St. Louis Adjusted Monetary Base	M_110154995	MB
5	74	6	TOTRESNS	Total Reserves of Depository Institutions	M_110155011	Reserves tot
6	75	7	NONBORRES	Reserves Of Depository Institutions	M_110155009	Reserves nonbor
7	76	6	BUSLOANS	Commercial and Industrial Loans	BUSLOANS	C&I loan plus
8	77	6	REALLN	Real Estate Loans at All Commercial Banks	BUSLOANS	DC&I loans
9	78	6	NONREVSL	Total Nonrevolving Credit	M_110154564	Cons credit
10	79*	2	CONSPI	Nonrevolving consumer credit to Personal Income	M_110154569	Inst cred/PI
11	131	6	MZMSL	MZM Money Stock	N.A.	N.A.
12	132	6	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	N.A.	N.A.
13	133	6	DTCTHFNM	Total Consumer Loans and Leases Outstanding	N.A.	N.A.
14	134	6	INVEST	Securities in Bank Credit at All Commercial Banks	N.A.	N.A.

Group 6: Interest and exchange rates

	id	tcode	fred	description	gsi	gsi:description
1	84	2	FEDFUNDS	Effective Federal Funds Rate	M_110155157	Fed Funds
2	85*	2	CP3Mx	3-Month AA Financial Commercial Paper Rate	CPF3M	Comm paper
3	86	2	TB3MS	3-Month Treasury Bill:	M_110155165	3 mo T-bill
4	87	2	TB6MS	6-Month Treasury Bill:	M_110155166	6 mo T-bill
5	88	2	GS1	1-Year Treasury Rate	M_110155168	1 yr T-bond
6	89	2	GS5	5-Year Treasury Rate	M_110155174	5 yr T-bond
7	90	2	GS10	10-Year Treasury Rate	M_110155169	10 yr T-bond
8	91	2	AAA	Moody's Seasoned Aaa Corporate Bond Yield		Aaa bond
9	92	2	BAA	Moody's Seasoned Baa Corporate Bond Yield		Baa bond
10	93*	1	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS		CP-FF spread
11	94	1	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS		3 mo-FF spread
12	95	1	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS		6 mo-FF spread
13	96	1	T1YFFM	1-Year Treasury C Minus FEDFUNDS		1 yr-FF spread
14	97	1	T5YFFM	5-Year Treasury C Minus FEDFUNDS		5 yr-FF spread
15	98	1	T10YFFM	10-Year Treasury C Minus FEDFUNDS		10 yr-FF spread
16	99	1	AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS		Aaa-FF spread
17	100	1	BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS		Baa-FF spread
18	101	5	TWEXMMTH	Trade Weighted U.S. Dollar Index: Major Currencies		Ex rate: avg
19	102*	5	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	M_110154768	Ex rate: Switz
20	103*	5	EXJPUSx	Japan / U.S. Foreign Exchange Rate	M_110154755	Ex rate: Japan
21	104*	5	EXUSUKx	U.S. / U.K. Foreign Exchange Rate	M_110154772	Ex rate: UK
22	105*	5	EXCAUSx	Canada / U.S. Foreign Exchange Rate	M_110154744	EX rate: Canada

Group 7: Prices

	id	tcode	fred	description	gsi	gsi:description
1	106	6	WPSFD49207	PPI: Finished Goods	M110157517	PPI: fin gds
2	107	6	WPSFD49502	PPI: Finished Consumer Goods	M110157508	PPI: cons gds
3	108	6	WPSID61	PPI: Intermediate Materials	M_110157527	PPI: int matls
4	109	6	WPSID62	PPI: Crude Materials	M_110157500	PPI: crude matls
5	110*	6	OILPRICEx	Crude Oil, spliced WTI and Cushing	M_110157273	Spot market price
6	111	6	PPICMM	PPI: Metals and metal products:	M_110157335	PPI: nonferrous
7	112	1	NAPMPRI	ISM Manufacturing: Prices Index	M_110157204	NAPM com price
8	113	6	CPIAUCSL	CPI : All Items	M_110157323	CPI-U: all
9	114	6	CPIAPPSL	CPI : Apparel	M_110157299	CPI-U: apparel
10	115	6	CPITRNSL	CPI : Transportation	M_110157302	CPI-U: transp
11	116	6	CPIMEDSL	CPI : Medical Care	M_110157304	CPI-U: medical
12	117	6	CUSR0000SAC	CPI : Commodities	M_110157314	CPI-U: comm.
13	118	6	CUSR0000SAD	CPI : Durables	M_110157315	CPI-U: dbles
14	119	6	CUSR0000SAS	CPI : Services	M_110157325	CPI-U: services
15	120	6	CPIULFSL	CPI : All Items Less Food	M_110157328	CPI-U: ex food
16	121	6	CUSR0000SA0L2	CPI : All items less shelter	M_110157329	CPI-U: ex shelter
17	122	6	CUSR0000SA0L5	CPI : All items less medical care	M_110157330	CPI-U: ex med
18	123	6	PCEPI	Personal Cons. Expend.: Chain Index	gmdc	PCE defl
19	124	6	DDURRG3M086SBEA	Personal Cons. Exp: Durable goods	gmdcd	PCE defl: dlbes
20	125	6	DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods	gmdcn	PCE defl: nondble
21	126	6	DSERRG3M086SBEA	Personal Cons. Exp: Services	gmdcs	PCE defl: service

Group 8: Stock market

	id	tcode	fred	description	gsi	gsi:description
1	80*	5	S&P 500	S&P's Common Stock Price Index: Composite	M_110155044	S&P 500
2	81*	5	S&P: indust	S&P's Common Stock Price Index: Industrials	M_110155047	S&P: indust
3	82*	2	S&P div yield	S&P's Composite Common Stock: Dividend Yield		S&P div yield
4	83*	5	S&P PE ratio	S&P's Composite Common Stock: Price-Earnings Ratio		S&P PE ratio
5	135*	1	VXOCLSx	VXO		