Market Stability with Machine Learning Agents*

Christophre Georges† Javier Pereira‡ Department of Economics Hamilton College

April 18, 2019

Abstract

We consider the effect of adaptive model selection and regularization by agents on price volatility and market stability in a simple agent-based model of a financial market. The agents base their trading behavior on forecasts of future returns, which they update adaptively and asynchronously through a process of model selection, estimation, and prediction. The addition of model selection and regularization methods to the traders' learning algorithm is shown to reduce but not eliminate overfitting and resulting excess volatility. Our results accord well with the empirical "sparse signals" and "pockets of predictability" findings of Chinco, Clark-Joseph and Ye (2019) and Farmer, Schmidt and Timmermann (2019), but stand in contrast to, for example, the model validation framework of Cho and Kasa (2015).

1 Introduction

We consider the role of adaptive model selection and regularization by agents on market stability in a simple agent-based model of a financial market. The agents in the model base their trading behavior on forecasts of future returns, which they update asynchronously. These forecasts then collectively drive market prices, and feed-back on the data generating mechanism. It is well-known that under restrictive conditions, adaptive learning in such an environment can converge to a noisy version of the stationary rational expectations equilibrium (Honkapohja and Mitra (2003)). However, if agents are uncertain about the true model of the world that they are operating in, there is substantial room for learning to go astray. In this context, the adoption of misspecified forecast rules can lead to additional excess volatility and bubbles.

The models that economic agents entertain can be misspecified in various ways - e.g., the exclusion of relevant variables (underparameterization) (Evans and Hohkapohja (2001, Ch. 13), Branch and Evans (2006), Hommes and Zhu (2014), Gabaix (2014)), the inclusion of intrinsically irrelevant variables (Grandmont (1998), Bullard et al. (2008), Georges (2008a,b, 2015), LeBaron (2012), Branch, McGough, and Zou (2017)). In much of this literature, attention is given to the role of the misspecification in driving endogenous market dynamics, but it is often assumed that

*Preliminary.

JEL Classification: D83 D84 E44

Keywords: Expectations, Agent-based Modeling, Machine Learning, LASSO

We thank Dan Kraynak, Noah Kramer, Paul Magnus, David Urbont, and Ian Wilson for research assistance. All errors are ours.

†Christophre Georges, Department of Economics, Hamilton College, Clinton, NY 13323, USA. cgeorges@hamilton.edu, 315-859-4472, http://people.hamilton.edu/christophre-georges.

‡Javier Pereira, Department of Economics, Hamilton College, Clinton, NY 13323, USA. jpereira@hamilton.edu, 315-895-4195, hppt://people.hamilton.edu/javier-pereira.

agents adopt misspecified rules uncritically and do not attempt to conduct model selection or other corrective methods in an effort to improve their forecasts. While fully rational expectations is empirically implausible, it is equally implausible that quantitative traders do not entertain the possibility that they are over- or underfitting the data. Indeed, this problem has generated a recent explosion of interest in predictive machine learning algorithms in financial markets.

In the present paper, we draw on the statistical and machine learning literature to add tools for model selection and regularization to the learning routines of our boundedly rational artificial traders. These agents update their forecasts on two time scales. Every trading period they update their forecasts based on their current forecast rules and the most recent data. On a slower time scale, they perform forecast model selection and estimation based on a longer history of recent data. Thus, our agents engage adaptively in model selection as well as model estimation and prediction. In our primary specification, the agents use the least absolute shrinkage and selection operator (LASSO), which performs both model selection and regularization. We also consider stepwise selection and two other related regularization methods – Ridge regression and elastic nets – that similarly address the problem of overfitting. We consider the effect of ongoing endogenous model selection and regularization on price volatility and market stability in a simple artificial financial market.

We work with a very simple artificial financial market fleshed out in a series of papers by Georges (2008a,b, 2015) in which all traders are chartists who use AR rules for forecasting returns. These rules are over-parameterized but nest the fundamental rational expectations (RE) rule. Thus, traders are willing to extrapolate trends in the recent data that are not supported by rational expectations, but they could in principle learn not to follow such trends. In the baseline model, the forecasting model is fixed, and updating is by least squares. The addition of model selection and regularization methods to the traders' learning algorithm is shown to reduce but not eliminate overfitting and resulting excess volatility. We explore the sensitivity of these results to the particular learning mechanism. Our results accord well with the empirical "sparse signals" and "pockets of predictability" findings of Chinco, Clark-Joseph and Ye (2019) and Farmer, Schmidt and Timmermann (2019), but stand in contrast to, for example, the model validation framework of Cho and Kasa (2015).

Our artificial traders can be thought of as model representations of either sophisticated human technical traders or purely algorithmic traders. While financial markets were once fully human operated, modern markets are hybrid spaces where computer and human trading coexist. The use of computers in asset markets comes in many forms ranging from the simple support of human traders in the scheduling of buying and selling assets to the more sophisticated algorithmic traders which can learn and autonomously decide which assets they sell or buy (Kirilenko and Lo, 2013). De Luca and Cliff (2011) have estimated that algorithmic traders are involved in up to 70% of the total trading volume in major European and US equity exchanges. According to Kaya (2016), high frequency trading – just one form of algorithmic trading – accounted for almost 50% of all the volume traded in the US equity markets in 2014. A central issue with regard to algorithmic trading is whether this activity has improved overall market quality, thus allowing investors to raise capital and manage risks more efficiently. Behavioral finance is often skeptical of the efficient market theory, suggesting that stock prices are to a certain extent predictable due to psychological and social aspects that lead to financial market inefficiencies (Shiller, 2003). Removing emotional entities from the market might thus be expected to improve market efficiency (Chaboud et al 2014). However, algorithms that identify and trade on patterns in recent historical data may also deviate substantially from the efficient markets, rational expectations benchmark.[1] Our results suggest that

[1]Indeed, algorithmic trading has been widely sited by market observers as a likely contributor to heightened

such deviations are unlikely to be fully eliminated as algorithms become increasingly sophisticated.

2 Literature Review

Our current framework follows closely Georges (2008a,b, 2015) in which all traders are chartists who use AR rules for forecasting returns. These rules are overparameterized but nest the fundamental RE (or minimum state variable (MSV)) rule. Agents forecast each period with their current rules and also fit their forecasting rules to recent data asynchronously on a slower time scale. Georges shows via simulation that when traders fit overparameterized forecast rules to the data the learning dynamics can become unstable. The degree of instability is shown to increase in the rate of learning and in the complexity (degree of overparameterization) of the forecast rule, and to decrease in the memory of the traders.[2] For nonlinear overparameterization, the instability can persist even with large memories. However, these agents are uncritical about the structure of the forecast rule and do not make any efforts at model selection.

Georges (2008b) includes an example with a simple form of model specification testing that reduces the selection of overparameterized models by the traders but still leads to intermittent spikes in volatility. The result is intuitive. An extra (non-MSV) lagged regressor fails a t-test much of the time, but it occasionally passes and so is adopted. Consequently, long periods of stability are punctuated by shorter periods of heightened volatility driven by the occasional adoption of a more complex forecasting rule. This is illustrative, but not particularly rigorous, leading us to wonder whether agents that were both aggressive about uncovering predictable structure but also smarter about model misspecification and overfitting the data would generate similar excess volatility and instability.

The analysis of Cho and Kasa (2015) suggests that more sophisticated model selection may lead to more efficient outcomes. Their agents entertain a variety of misspecified models and perform model testing, which they call model validation, on a slower time scale than model estimation. Each model has a self confirming equilibrium, but also the possibility of escape that could trigger a switch to another model. They use an information criterion to test the current model, not explicitly against the alternative models, but rather based on the stability of the current model parameter estimates. If the current model fails the test, another model is selected at random. Thus, only one model is entertained at a time. They argue for this model validation procedure, which tends to eventually lock in on a single rule (in the limit as updating rates go to zero, in the spirit of Kandori, Mailath, and Robb (1993)) over Bayesian model averaging or other specification approaches. The presumption is that agents believe there to be a single correct model, and they believe that they have found this correct one until dissuaded, waiting for escape episodes to trigger further search and experiment in the model space.

Cho and Kasa apply this approach to the case of a single agent (a central bank) which conducts model selection knowing that its actions impact the data generating process directly. This approach appears to be less well suited to environments in which agents are highly heterogeneous and are searching for occasional pockets of predictability that they expect to appear and disappear, as is the case with technical traders in financial markets. In such an environment, individual agents are perpetually searching for better short term predictive models. Endogenous dynamics are then driven by the entire collection of the agents' forecasts, and thus, through a de facto kind of model

volatility in a variety of instances from flash crashes to market trend reversals (Kirilenko et al. (2017), Zuckerman (2018), Yang (2019), Kaminski (2019)).

[2] Branch and Evans (2011) and Georges (2015) show that adaptive learning about risk can provide an additional source of local instability in this context.

averaging.

We allow agents to perform on-line model selection using several methods including forward selection and LASSO. The latter is a parameter shrinkage method designed to combat overfitting that can handle very large numbers of regressors (even many more than the sample size). This approach, which has been referred to as betting on sparsity, assumes that there are only a handful of strong predictors at any point in time (Hastie et al. (2009, 2015)). The procedure will tend to shrink the coefficients on many of the regressors towards zero as well as set some of them equal to zero. This is a distinctive feature of LASSO relative to other similar regularization methods such as Ridge regression that by construction will tend to include all predictors in the final model. Hence, LASSO performs variable selection, and as a result it yields sparse models – that is, models that involve a subset of variables and so are easier to analyze and interpret.

There is a growing literature on the role of sparsity in financial economics. DeMiguel et al.(2014) finds that simpler predictive regressions perform better out-of sample than denser models because they mitigate overfitting. Gabaix (2014) proposes a framework in which the agent builds a simplified model of the world which is sparse – considering only the variables of first-order importance – by internally imposing a penalty a la LASSO. Chinco, Clark-Joseph, and Ye (2019) uses LASSO to identify pockets of predictability for 1 minute ahead stock return forecasts from a large number of potential predictors. They find, for stocks listed on the NYSE, that these pockets are short lived and that the set of predictors for each pocket is sparse but idiosyncratic and thus difficult to identify by economic intuition alone. Yet they document that these predictors, identified via LASSO, do tend to capture information about changes in relevant fundamentals. Consistent with these pockets of predictability that are short-lived, Brogaard et al. (2014) show that algorithmic traders' orders predict price changes over very short horizons and are correlated with macroeconomic news.

Farmer, Schmidt, and Timmermann (2019) present additional evidence on pockets of predictability. They adopt an estimation strategy based on time-varying regression coefficients that lets the data determine both how large any predictability is at a given point in time and how long it lasts. They find that return predictability is highly concentrated or local in time, tending to present itself in pockets.

It is worth noting there have been criticisms of LASSO and other predictive machine learning methods on the grounds that they are poor tools for inference. For example, the specific set of regressors selected by LASSO can be sensitive to small changes in the data, and so the selection or non-selection of specific variables should not be taken to imply that those variables do or do not matter causally (Giannone et al., 2018). While this is a serious issue for cases in which drawing causal inference is important, our traders care about prediction, not inference, and as such can be well served by LASSO and other predictive machine learning methods.[3]

3 Market Structure

The basic market structure follows Georges (2008a,b). We assume that there are two assets, a stock that pays a dividend d_t in each period t and a bond with a fixed rate of return r in each period. Dividends are given by $d_t = \overline{d} + \varepsilon_t$, where $\overline{d}$ is a constant and the ε_t are iid with zero mean and finite variance. P_t is the price of the stock at time t.

In each period t, each trader i constructs a forecast $F_t^i[P_{t+1} + d_{t+1}]$ of the price plus dividend of the stock in the following period. For simplicity, we assume the market clearing price in period t is the price that equalizes the forecasted returns on the two assets for an average trader. Hence,

[3] Belloni et al. (2014), and Mullainathan and Spiess (2017) discuss other useful applications in economics of LASSO as a predictive tool.

the price of the stock in period t satisfies

$$P_t = \frac{\overline{F}_t[P_{t+1} + d_{t+1}]}{1+r} \tag{1}$$

where $\overline{F}_t[P_{t+1} + d_{t+1}]$ is the forecast of a representative agent.[4]

There is a unique stationary rational expectations equilibrium $P_t = P^* \; \forall t$, where $P^* = \frac{\overline{d}}{r}$. As the price is constant in this equilibrium, any volatility in price represents a deviation from the stationary REE. It will be useful to define $x_t = P_t + d_t$ and $x^* = P^* + \overline{d}$.[5]

4 Forecast Rules

We assume that traders do not have enough information about the structure of the dividend process and the other agents' beliefs to form rational expectations. Rather they formulate forecasts of future prices inductively. Specifically, we suppose that all agents are technical traders who use forecast rules with a common polynomial autoregressive functional form

$$F_t^i[x_{t+1}] = P_t^i(x_t, x_{t-1}, x_{t-2}, ...) \tag{2}$$

with coefficients $a = (a_{0t}^i, a_{1t}^i, ...)$ that can vary across agents i and time t. An example of such a forecast rule would be the following quadratic AR(3) rule[6]

$$F_t^i[x_{t+1}] = a_{0t}^i + a_{1t}^i \cdot x_t + a_{2t}^i \cdot x_{t-1} + a_{3t}^i \cdot x_{t-1}^2 + a_{4t}^i \cdot x_{t-2} + a_{5t}^i \cdot x_{t-2}^2 + a_{6t}^i \cdot x_{t-1} \cdot x_{t-2} \tag{3}$$

We will take the average forecast $\overline{F}$ used in equilibrium condition (1) to be the algebraic average of the forecasts (2) over the agents.

4.1 Benchmark Forecasts

The stationary REE forecast is

$$F(x_{t+1}) = x^* \tag{4}$$

and so the stationary REE forecast rule is simply the minimum state variable (MSV) forecast rule $F(x_{t+1}) = a_0$ with $a_0 = x^*$.[7]

Another benchmark that is useful to consider is a *weak consistency condition* (Georges, 2008b) for the forecast rule that restricts the coefficients to satisfy

$$x^* = F[x_{t+1}](x^*) \tag{5}$$

This requires that forecasts are consistent with the stationary REE at the stationary REE.[8]

---

[4] We assume risk neutrality in order to remove additional volatility that can arise from risk sensitivity as in Branch and Evans (2011) and Georges (2015).

[5] Note then that at the stationary REE, $x_t = x^* + \varepsilon_t \; \forall t$.

[6] We will assume that x_t is not in traders' information sets at time t, so that traders using rule (3) would form iterated forecasts of x_{t+1} by first forecasting the current period's value x_t using the observed values from the preceding three periods.

[7] Thus, for the example forecast rule (3), the stationary REE forecast rule would be given by $a = (a_0, a_1, a_2, a_3, a_4, a_5, a_6) = (x^*,0,0,0,0,0,0)$.

[8] For the example forecast rule (3), the WCC would require the coefficient vector a to satisfy $x^* = a_0 + (a_1 + a_2 + a_4) \cdot x^* + (a_3 + a_5 + a_6) \cdot x^{*2}$ and so be restricted to lie on a 6 dimensional hyperplane in the 7 dimensional parameter space.

5 Forecast Rule Updating

All agents will update their forecasts given their current estimated rules in each period. Note that, since the form of the forecast rule is common across agents, and all agents have access to the same historical data, all heterogeneity in the model will be due to asynchronous rule updating. Rule updating will take two forms, updated estimation, and updated model selection.

In the baseline model, the forecast rule is fixed and updating is by least squares estimation. That is, we assume that agents adopt a specific rule form uncritically. More sophisticated agents, however, would entertain alternative forecasting models and would also be sensitive to the hazards of overfitting the available data. In this spirit, we will subsequently incorporate model selection and other regularization methods to the traders' learning algorithm.

5.1 Baseline: Least Squares Learning

In the baseline case, the form of the rule (2) is fixed. At the start of each period t, each agent is selected to update her forecast rule parameters with probability *pupdate*. If agent i updates her rule in t, she chooses the rule that minimizes the sum of squared forecast errors over the preceding M (memory) periods. The new rule parameters a^i_{jt} minimize

$$\sum_{k=1}^{M} (x_{t-k} - F^i[x_{t-k}])^2 \tag{6}$$

Thus, agents learn using a finite memory least squares learning algorithm – they periodically update their rules to the rule that currently best fits the recent data according to OLS. While not all agents will update their rules in a given period t, each agent who is selected to update her rule at t will select the same new rule, and therefore join a transitory cohort that shares common forecasts but progressively dissolves over time.

5.2 LASSO

Now we consider the specification of forecast rules. Our focus is on the LASSO regression, which is able to mitigate overfitting by using a penalty function that both shrinks coefficient estimates and removes all but the strongest predictors. LASSO regression is similar to least squares but the coefficients are estimated by minimizing a slightly different function. The LASSO coefficient estimates (a^L_0,a^L_1,...) are the quantities that minimize

$$\sum_{k=1}^{M} (x_{t-k} - F[x_{t-k}])^2 + \lambda \cdot \sum_{j=1}^{p} |a_j| \tag{7}$$

where $\lambda \geq 0$ is a tuning parameter to be determined separately and p is the number of regressors (predictors) in the regression.[9] As with least squares, the LASSO regression looks for coefficient estimates that fit the data well. However, the second term, called a shrinkage penalty, is small when the coefficients a_j are close to zero, and so it has the effect of shrinking the parameter estimates towards zero.[10] But different from other similar (but denser) regularization methods, such as Ridge regression, the particular structure of the penalty function allows the LASSO to exploit a bet on

[9] $P = 5$ in (3).

[10] Note that the shrinkage penalty is not applied to the intercept (a_0). We do not want to shrink the intercept which represents the mean value of the response when $x_j = 0$ for $j \neq 0$.

sparsity. While Ridge regression will tend to generate a model involving all predictors, the LASSO will not only shrink coefficient estimates but it will also force some coefficient estimates to be equal to zero. In this sense, the LASSO performs variable selection.[11] To see this, consider the solution to equation (7) when there is only one predictor

$$a^L = \begin{cases} Sign[a^O] \cdot (|a^O| - \lambda) & \text{if} |a^O| \geq \lambda \\ 0 & \text{otherwise} \end{cases}$$

where a^O represents the OLS coefficient computed as in equation (6). So if the OLS coefficient is more extreme than the LASSO's penalty parameter, $|a^O| > \lambda$, then the LASSO will estimate a shrunken version of the OLS coefficient, namely, $a^L = sign[a^O] \cdot (|a^O| - \lambda)$. Whereas, if the OLS coefficient is less extreme than the penalty parameter, the LASSO will estimate a^L=0.

5.2.1 Selecting the tuning parameter

Implementing LASSO requires a method for selecting the value for the tuning parameter λ. Cross-validation (CV) provides a simple way to address this consistent with the goal of minimizing overfitting. In particular, our agents employ *k*-fold CV that involves randomly dividing the set of observations into *k* groups [folds] of equal size. The first fold is treated as a validation set, and the LASSO regression is fit on the remaining *k-1* folds. The mean squared error, MSE_1, is computed with the observations in the held-out fold. This procedure is repeated *k* times where a different fold of observations is treated as the validation set. The process yields *k* estimates of the test error, MSE_1, MSE_2,..., MSE_k. The *k*-fold CV error is the average of these values,

$$CV_{(K)} = \frac{1}{k} \cdot \sum_{i=1}^{k} MSE_i \tag{8}$$

There is a bias-variance trade-off associated with the choice of *k*. It is thus common practice to perform *k*-fold CV using $k = 5$ or $k = 10$ as these values have been shown empirically to produce test error rate estimates with relatively small bias and variance. We have our agents use 10-fold CV.

So, following this standard practice, we assume that agents who are updating their forecasts using LASSO choose a grid of λ values, compute the cross-validation error for each value of λ, and select the tuning parameter value for which the cross-validation error is smallest. Finally, they estimate their model using all of the available observations and the selected value of the tuning parameter, and use this estimated model to make their forecasts.

6 Comparison of Estimated Forecast Rules for Artificial Data

Consider a simple case in which the forecast rule (2) is linear AR(2):

$$F_t^i[x_{t+1}] = a_{0t}^i + a_{1t}^i \cdot x_t + a_{2t}^i \cdot x_{t-1} \tag{9}$$

[11] This is a distinctive feature of LASSO, relative to its cousin Ridge regression, in that it yields simpler and more interpretable models that involve only a subset of predictors.

Let r=0.05 and $\overline{d}$=0.5 so that P^*=10, x^*=10.5, and the forecast rule parameter values corresponding to the stationary REE are $(a_0, a_1, a_2) = (10.5, 0, 0)$. To gain some intuition before turning to the full simulation model, we follow Georges (2008b) and first generate a random series $x_t = x^* + \varepsilon_t$, where the ε_t are iid uniform on (-0.5, 0.5) and consider the evolution of the estimated forecast rule $(a^*_{0t}, a^*_{1t}, a^*_{2t})$ under the OLS and LASSO estimators, i.e., the rules that minimize the loss functions (6) and (7) in each time period for this artificial history. These would be the estimated rules adopted by all agents if they updated their estimates in every period using these two model fitting approaches.

Figure 1 displays examples for 2000 periods of an artificial history with memory set at 50, 100, and 500 under OLS and LASSO updating. Estimated forecast rules under both learning mechanisms vary over time but tend to lie near the WCC (5) and are roughly centered on the stationary REE rule.[12] We can see that the degree of variation of the parameter estimates under both methods decreases as memory increases. Further, as illustrated in Table 1, the inclusion of additional regressors (from (2)) as we move from forecast rule (9) to forecast rule (3) increases the variability of parameter estimates for both methods. So in these regards, the behavior of OLS and LASSO is similar. However, Figure 1 and Table 1 also illustrate that LASSO tends to yield a substantial reduction in variance relative to OLS updating.[13] This is to be expected, as LASSO performs variable selection and shrinkage of coefficient estimates, thus reducing overfitting.[14]

Table 1: Standard deviation of estimated forecast rule parameters

OLS

M	Forecast Model			
	(9)		(3)	
	a_1	a_2	a_1	a_2
50	0.118	0.141	0.129	13.449
100	0.078	0.095	0.079	9.721
500	0.020	0.030	0.021	6.361

LASSO

M	Forecast Model			
	(9)		(3)	
	a_1	a_2	a_1	a_2
50	0.061	0.087	0.062	3.906
100	0.041	0.052	0.035	3.336
500	0.004	0.010	0.008	2.928

Same artificial x series as in Figure 1

[12] Estimated forecast rules cluster near the WCC which is a 2 dimensional plane through the 3 dimensional space.
[13] This occurs at the expense of some increase in bias.
[14] Interestingly, the cross patterns observed in Figure 1 are consistent with LASSO's variable selection feature that yields WCCs with a1=0 and a2=0, respectively

7 Simulations

Turning to the full simulation model, we first consider the case in which all agents use MSV forecast rules, namely, $F_t^i[x_{t+1}] = a_0^i$, and updating is by least squares. If for all agents $a_0^i = x^*$, then the market will be at the stationary REE. As documented in Georges (2008a,b), simulations exhibit persistent dynamics in the forecast rule parameter a_0^i and thus in the price level P_t as a result of the finite memory and dividend shocks. These dynamics become centered on the REE values x^* and P^* and excess returns hover around 0. Thus, price and expectations dynamics converge to a noisy version of the stationary REE. OLS and LASSO are the same under the MSV rule because there are no parameters to shrink; a_0^i does not enter in the penalty function in (7).

As we move to overparameterized forecast rules, we can compare simulation outcomes under OLS and LASSO rule updating. In both cases, we still observe a general attraction to the stationary REE as the forecast rule parameter values fluctuate around both (x^*,0,0,0,...) and the WCC (5), and the price levels fluctuate around P^*.

We also see that the average forecast rule parameters wander broadly and continuously under OLS updating, whereas under LASSO updating, they switch between periods in which newly estimated rules take an MSV form and periods in which traders are drawn to include AR terms in their estimated rules. Much of the time, LASSO induces traders to select no autoregressive components, but occasionally they are led to believe that there is a window of predictability that can be identified with a more complex forecast rule. These windows correspond to periods of punctuated volatility that emerge endogenously from periods of quiescence in price dynamics. Results for LASSO accord well with the empirical "sparse signals" and "pockets of predictability" findings of Chinco, Clark-Joseph, and Ye (2019). This is in contrast to OLS, under which, as an artifact of overfitting, the data appears to be predictable at every time period. The upshot for the price and returns dynamics is that, under LASSO, volatility is generally lower, but is punctuated by more pronounced volatility clustering (with lower variance but greater time separation) and greater excess kurtosis than under OLS updating. An illustration is provided in Figures 2 and 3.

Under both OLS and LASSO updating, we also observe the development of occasional explosive price bubbles and crashes, in which the price level deviates substantially and persistently from its stationary REE value of $\bar{d}/r$. We run Monte Carlo experiments in which, for each combination of parameter values that we select, we run the simulation for 2000 trading periods 500 times (with different random seeds) and count the number of runs that exhibit bubbles (extreme prices). For both OLS updating and LASSO, the incidence of bubbles is increasing in the speed of learning and the complexity of the forecast rule, and decreasing in the traders' memory.[15] Here we also see that the incidence is substantially higher for OLS than for LASSO, and this difference is more pronounced as the complexity of the forecast rule is increased.

Figures 4a - 4c and 5a - 5c indicate the frequency of the formation of explosive bubbles and crashes for different specifications of forecast rule and learning technology.[16] The ability of LASSO to perform variable selection and parameter shrinkage lowers the incidence of extreme rules which in turn lowers the frequency and magnitude of large price movements. Nevertheless, traders using LASSO periodically identify pockets of predictability and switch from MSV rules to complex rules. Even though LASSO imposes parameter shrinkage on these complex rules, and so they tend to be less extreme than rules estimated by OLS, they are still extreme enough to produce occasional bubbles driven by positive feedback between these rules and the pricing mechanism.

[15] This has been previously documented for OLS updating by Georges (2008a,b, 2015).

[16] The extreme (bubble) price thresholds are set at 0 and 20. $P^* = 10$ as in the example in Figure 2.

8 Can Agents Do Better than LASSO?

A concern that generally motivates the adoption of machine learning procedures is the potential for severe over-fitting in high dimensional settings. LASSO protects against overfitting the recent price data by a combination of coefficient estimate shrinkage and variable selection (setting sufficiently small estimates to zero). Further, above we followed the standard procedure of selecting the penalty parameter λ by cross validation so that it is optimized for out of sample fit.

We highlight two good reasons to favor the variable selection aspect of LASSO as our main specification over denser regularization methods. First, it represents a convenience for the researcher, as it facilitates the monitoring and interpretation of the real time updating of forecast rules and its implications (e.g., volatility increases when agents turn on more regressors). Second, we believe that it squares well from a behavioral standpoint, at least as far as human traders are concerned. To the degree that boundedly rational traders make sense of their worlds by focusing on specific signals in the data, this likely aligns more closely from a modeling perspective with them performing model selection by choosing specific regressors (predictors) than with them using less transparent and thus more difficult to intuit machine learning methods. In this regard, Gabaix (2014) takes a sparsity approach to modeling decision making. Freyberger et al. (2019), Feng et al. (2017), and Kozak et al. (2017) use the LASSO to choose between the various predictors that have already been identified by other researchers in the literature. Chinco et al. (2019) take a bottom-up approach and use the LASSO to show that, while a researcher cannot simply use her intuition to identify candidate predictors that are sufficiently unexpected and short-lived, such predictors are economically relevant. In a similar vein, Lo et al. (2000) note that much of human-based technical analysis involves the subjective identification of simple non-linear patterns in time series data, and the econometric evaluation of some of the more commonly used patterns.

However, since our general approach is to push our agents to do their best to avoid overfitting, we need to consider the possibility that they could do better than LASSO, either intuitively or with the assistance of machines. Behaviorally, it is also possible that more opaque machine learning algorithms better describe some of the intuitive identification of trading opportunities made by human traders. Further, computing technology is continuing to revolutionize the way financial assets are traded. In handling orders without immediate human intervention, computer algorithms increasingly are used to identify trading opportunities, make trading decisions, submit orders and manage these afterwards, all at tremendous speeds. The machine learning methods currently used in these processes for identifying trading opportunities certainly include highly dense and opaque prediction methods, such as deep neural networks, which are often treated as black box prediction machines.

For these reasons, we consider both a different model selection method – stepwise selection – as well as denser regularization methods – Ridge regression and elastic net, in order to comment on how robust our results are to relaxing the assumption of LASSO.

8.1 Stepwise Selection

As an alternative method for model selection and estimation, we first consider forward stepwise selection. Under this framework, when an agent is selected to reestimate her model, she takes the set of possible regressors available in (2) and sequentially searches for the best next regressor (in terms of smallest RSS). Once she has exhausted this sequence, she then selects among the resulting best models with 1, 2, 3 , ... ,p regressors to minimize BIC. Hence, this form of model selection focuses on variable selection but does not perform additional regularization through shrinkage. Our prior is that this will generate greater overfitting and consequently greater volatility than under

LASSO.

8.2 Ridge Regression

Ridge regression is similar to the LASSO but the coefficients are estimated by minimizing a slightly different function. Coefficient estimates (a_0^R,a_1^R,...) are the quantities that minimize

$$\sum_{k=1}^{M}(x_{t-k} - F[x_{t-k}])^2 + \lambda \cdot \sum_{j=1}^{p} a_j^2 \tag{10}$$

where $\lambda \geq 0$ is the tuning parameter to be determined separately using cross-validation. Comparing (7) and (10), we observe that LASSO and Ridge regression have similar specifications. The only difference is that $|a_j|$ is replaced by a_j^2, i.e., the l_1 penalty is replaced with an l_2 penalty. This penalty is still small when (a_1^R,a_2^R,...) are close to zero, and so still has the effect of shrinking coefficient estimates towards zero, but now it does not force some coefficients to be exactly equal to zero.

8.3 Elastic Net

The elastic net combines the LASSO and Ridge regression penalties. Coefficient estimates (a_0^E,a_1^E,...) are the quantities that minimize

$$\sum_{k=1}^{M}(x_{t-k} - F[x_{t-k}])^2 + \lambda \cdot \sum_{j=1}^{p}[(1-\alpha)a_j^2 + \alpha|a_j|] \tag{11}$$

where $\alpha \in [0, 1]$ is a parameter that determines the balance between Ridge and LASSO penalties and $\lambda \geq 0$ is the penalty tuning parameter chosen using cross-validation.[17]

Both Ridge and elastic net may perform better than LASSO when regressors are highly correlated. As with OLS, when regressors are correlated, the LASSO estimator will suffer from high variance. Ridge will tend to yield similar and less variable estimates for coefficients of correlated variables – thus clustering these variables. However, as discussed in Abadie and Kasy (2017), there is no one method for regularization that is universally optimal.[18]

8.4 Simulations

Simulations confirm our intuition that, when practiced by our agents, stepwise selection produces instability that is intermediate, between the OLS and LASSO results. We find that the variability of the estimated rule coefficients, prices and returns as well as the incidence of explosive bubbles and crashes tend to fall between the cases of OLS and LASSO updating. The intermediate incidence of bubbles is shown in Figures 4 and 5. LASSO protects better against overfitting than stepwise selection by adding parameter shrinkage to parameter selection, and so yields less self reinforcing trend chasing.

Figures 6a-6c indicate the frequency of the formation of bubbles for the two denser regularization methods. Runs for elastic net with $\alpha = 0.5$ are broadly in line with LASSO results while Ridge regression (elastic net with $\alpha = 0$) yields significantly lower instability. In our context, these results

[17] α is an additional tunning parameter. It can be determined at the researcher's discretion or using cross-validation.

[18] Abadie and Kasy (2017) also show that the choice of tuning parameters using cross-validation is guaranteed to work well in high-dimensional estimation and prediction settings under relatively mild conditions.

suggest that due to endogenous feedback, non-zero coefficients under LASSO may be more extreme than the smaller but more numerous coefficient estimates under Ridge, which leads to greater feedback and thus endogenous instability.

However, results for more complex forecast rules (more lags and powers in the forecasting rule (2)) suggest an interesting come back of LASSO relative to Ridge regression in terms of producing similar instabilities as the number of regressors climbs into the hundreds.[19] A distinctive feature of Ridge in contrast to LASSO is that it allows shrinkage without shrinking some observations all the way to zero. However, as the share of true zeros (redundant predictors) increases – which is likely to be the case with substantially more complex forecast rules – the relative performance of Ridge deteriorates.[20]

We thus confirm that no method universally dominates. The relative performance of the different regularization methods depends on whether traders entertain simpler or more complex forecast rules.

9 Conclusion

The overparameterization of traders' forecast rules exascerbates overfitting which increases market volatility and instability. LASSO mitigates this problem: traders who use LASSO to search for predictable price movements while attempting to avoid overfitting the data available to them are less likely to overparameterize their forecast models and overfit the data than are traders using OLS without model selection or regularization. Nevertheless, as in Grandmont (1998), they are willing to entertain a wide range of possible price forecasts. Thus, they will still occasionally chase apparently predictable dynamics in ways that generate heightened excess volatility or more dramatically destabilize the market. The occasional emergence of apparent pockets of predictability accords with the empirical results of Andersen and Sornette (2008), Chinco, Clark-Joseph and Ye (2019) and Farmer, Schmidt and Timmermann (2019). The simulation results reported here suggest that even a high degree of attention to overfitting on the part of traders who are engaged in data mining may not entirely eliminate destabilizing speculation.

[19] For example, simulations with forecast rule (2) with 21 lags and 20 powers (number of predictors $p = 591$) yield very similar levels of instability for both Ridge and LASSO cases.

[20] With the number of predictors p greater than the number of available observations n, LASSO will select at most n predictors to have non-zero coefficient estimates, whereas Ridge will tend to use all of the available regressors. There is also an interesting discontinuity that we observe with all three regularization methods for more complex (large p) rules as memory (and thus the number of observations n in agent regressions) is increased and we pass from $p > n$ to $p < n$. Instability is high at low memory, decreases as memory is increased, but jumps back up somewhat when p becomes lower than n, and then continues to fall again as memory increases further. This can be seen in Figure 6, for which the number of predictors (regressors) in the forecasting rule is $p = 31$.

Figure 1: 3-D Scatter Plots of Estimated Values (a_0^*,a_1^*, a_2^*) for 2000 periods of an artificial history under OLS and LASSO updating. Memory is set to 50, 100, and 500. In all cases, we see a rough centering of optimal rules on the MSV REE rule (10.5,0,0) as well as on the WCC. For each case, we provide a "zoom in" version of the 3-D scatter plot.

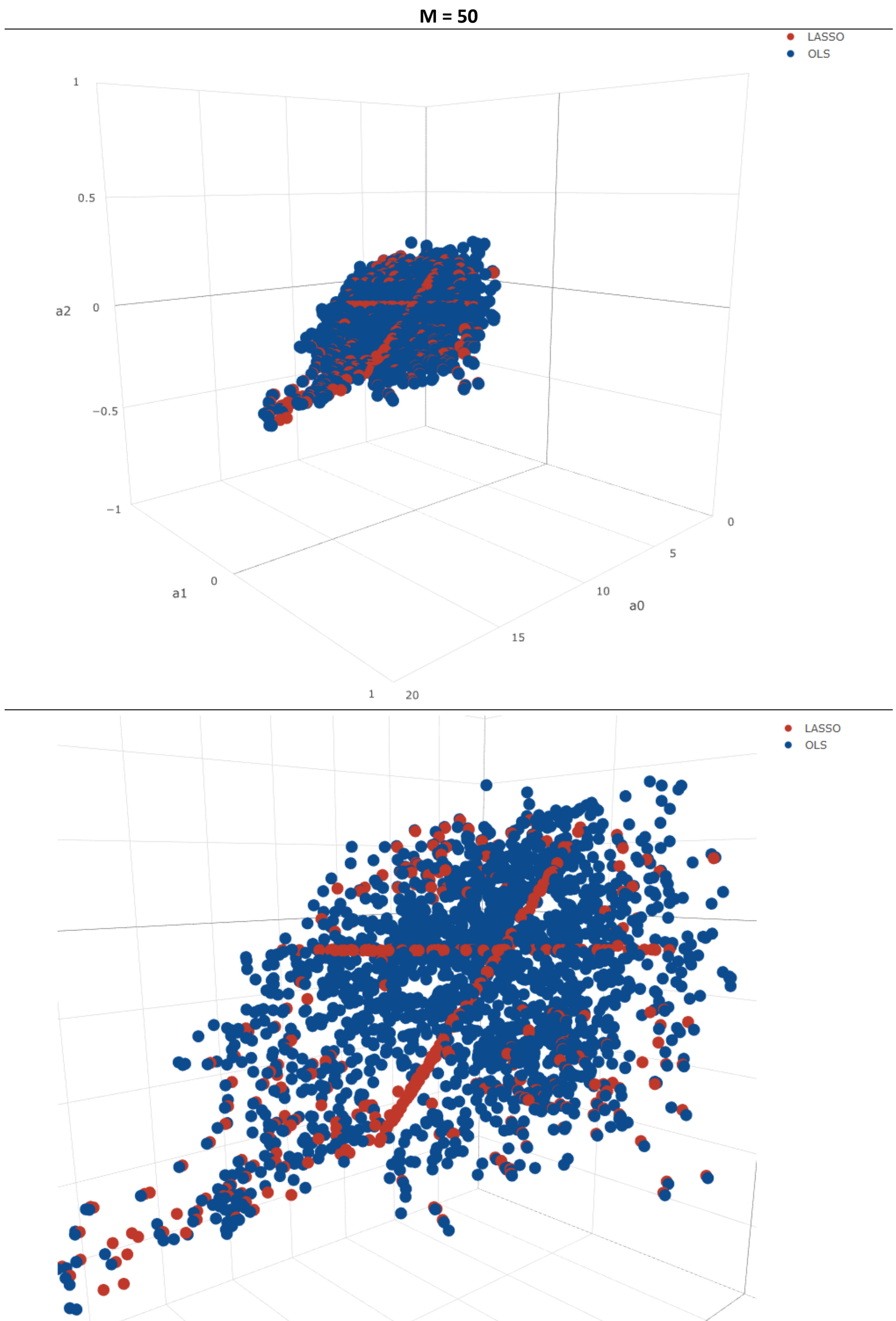
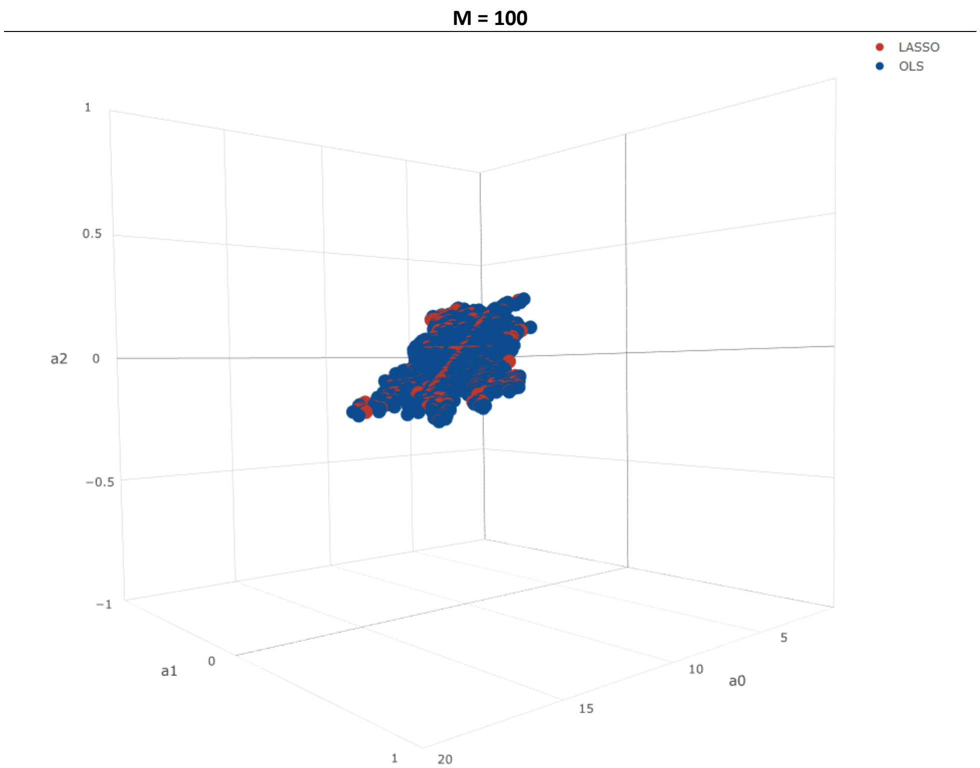
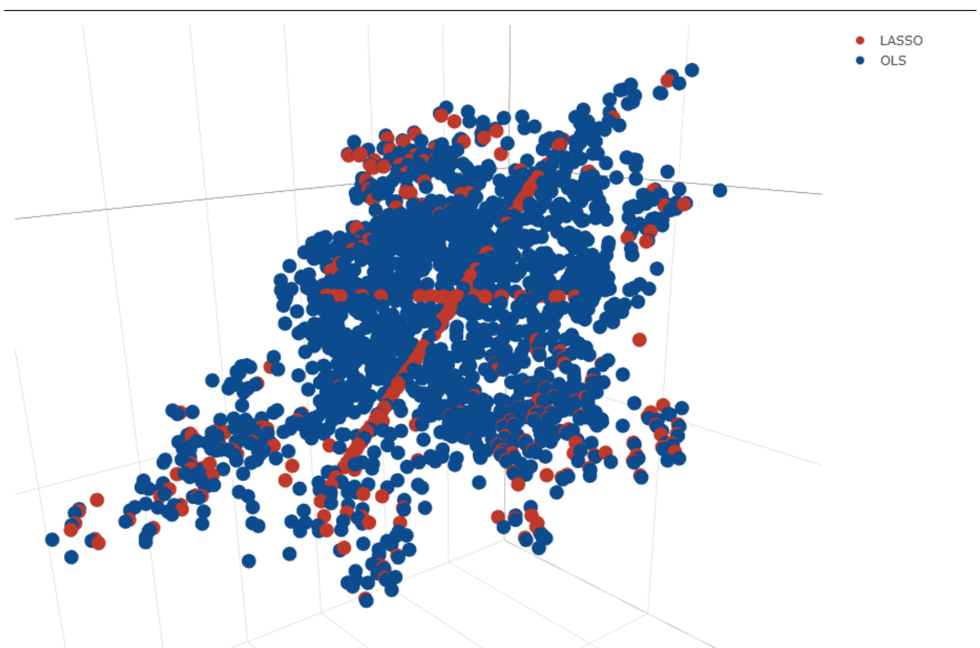
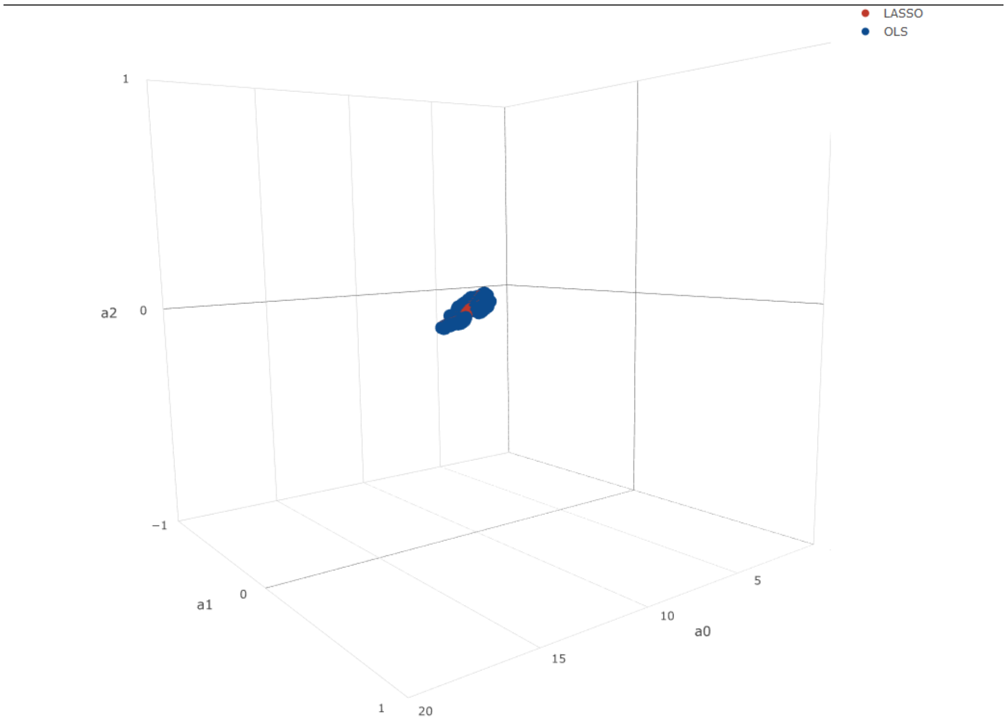


Figure 1: Continued





M = 500



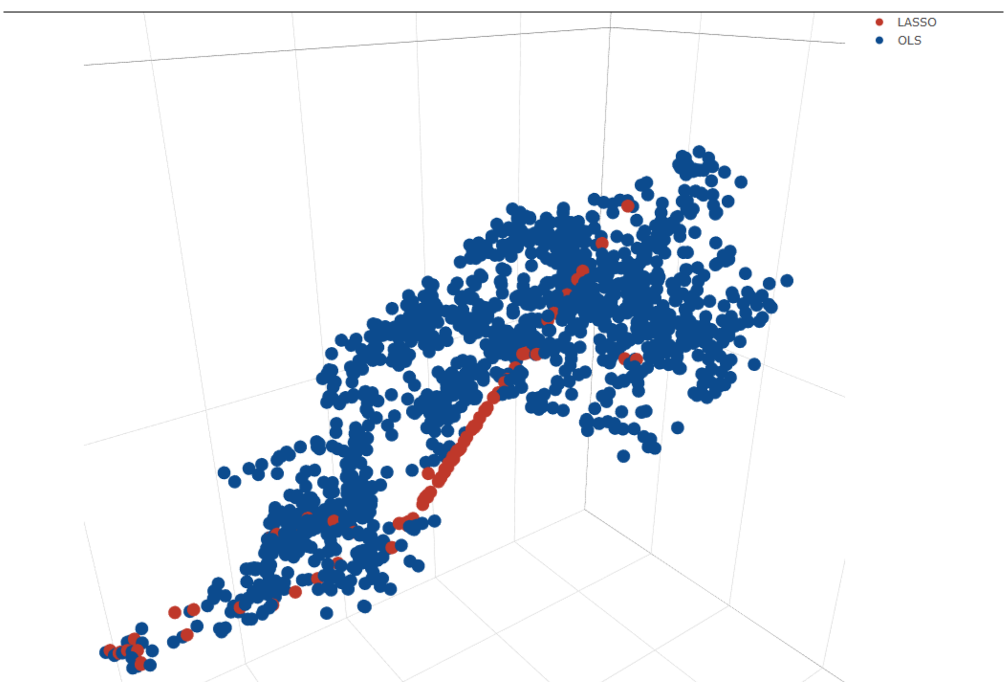
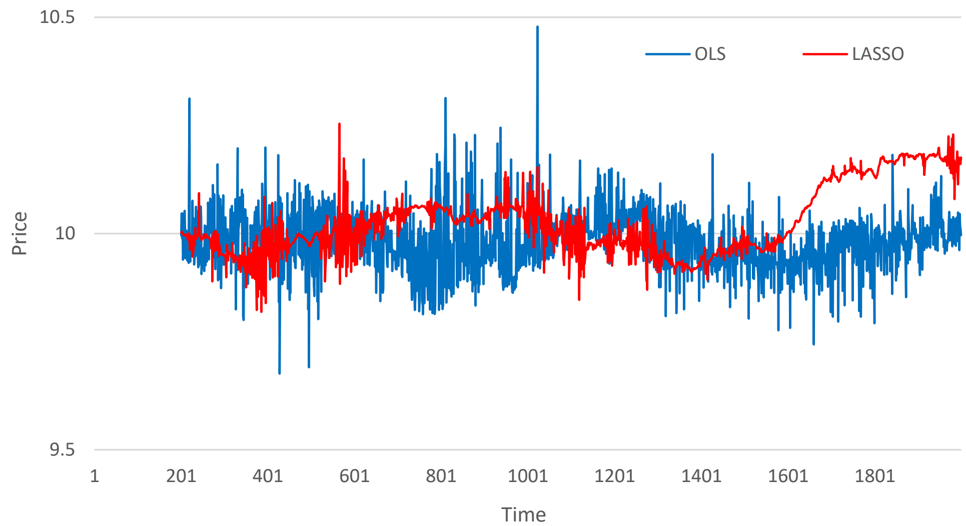


Figure 2: Run with memory M=100 and pupdate=0.5. Forecast rules are of nonlinear AR(3) form as specified in (3). $\overline{d} = 0.5$, P^{*}=10, and so P^{*}=10. Dividend shocks are uniformly distributed on (-0.025, 0.025) and occur with probability 0.8 in any period. Shown here are prices and returns for 2,000 rounds of a representative run with OLS and LASSO updating



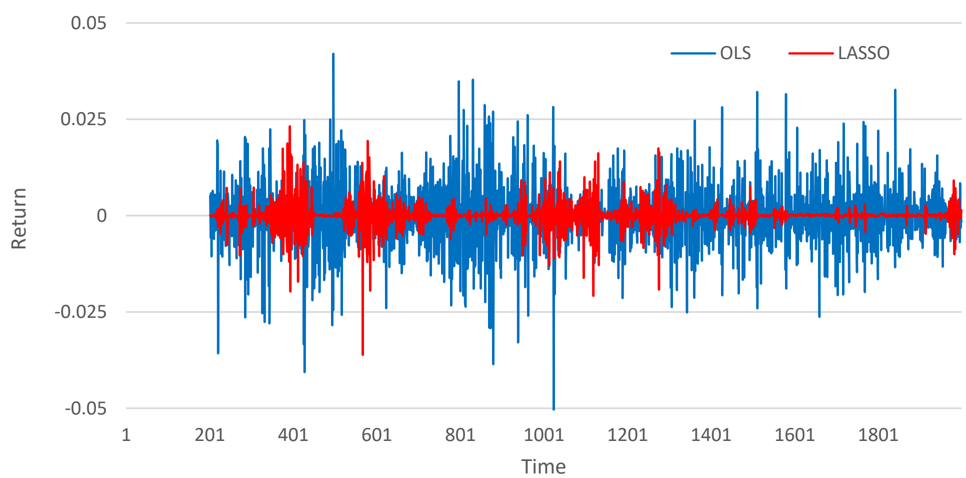
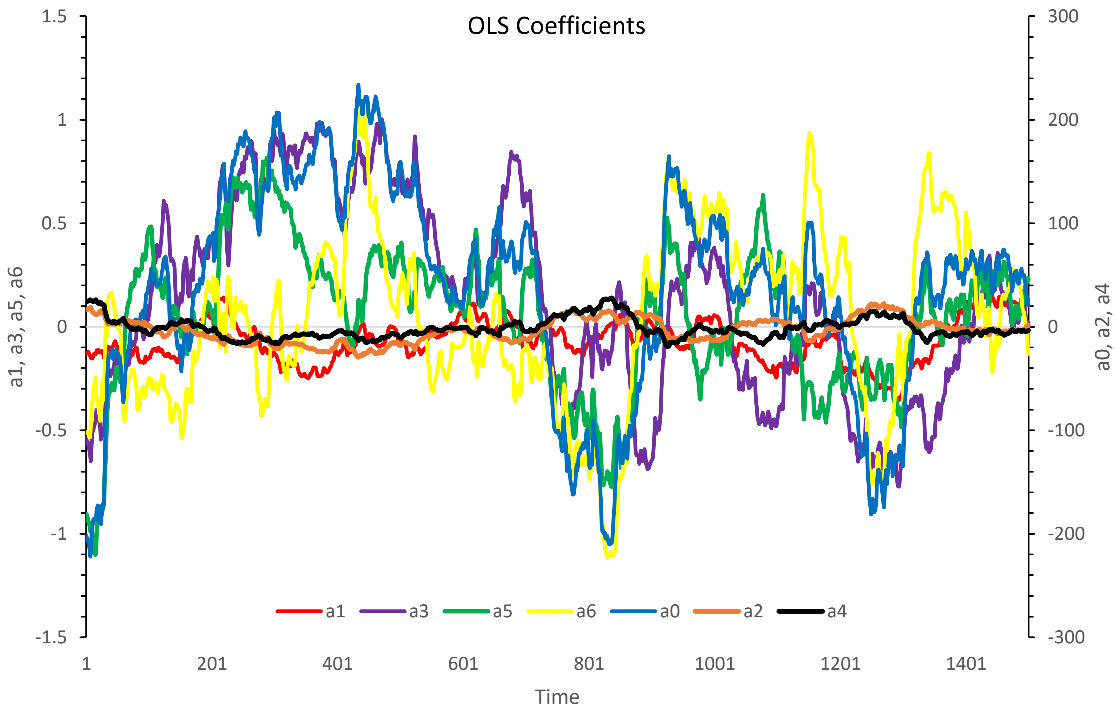


Figure 3: Additional output from previous specification. Shown here are average rule parameters ($\bar{a}_0$,...,$\bar{a}_6$) under OLS and LASSO updating. The corresponding stationary REE rule is (10.5,0,0,0,0,0,0).



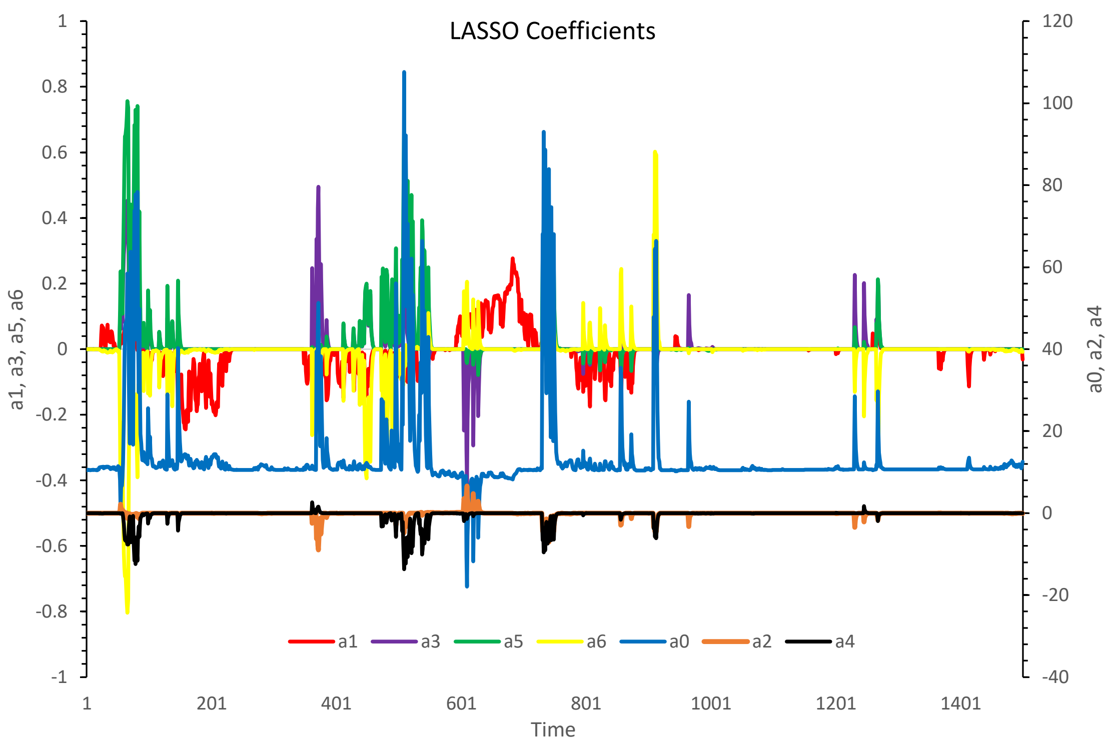


Figure 4: Frequency of runs out of 500 in which an explosive bubble forms within 2,000 trading periods for 90 combinations of memory M and learning rate pupdate. Figures 4a-4c represent different updating technologies for a linear AR(2) forecast rule (9).

Fig. 4a: Forecast rules are AR(2) (linear) with OLS updating

	OLS								
	Pupdate								
Memory	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
10	0	0	0.064	0.314	0.694	0.902	0.986	0.998	0.998
20	0	0	0.006	0.042	0.078	0.188	0.294	0.38	0.46
30	0	0	0.002	0.002	0.008	0.008	0.034	0.058	0.07
40	0	0	0	0	0.006	0.002	0	0.014	0.006
50	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0

Fig. 4b: Forecast rules are AR(2) (linear) with Forward Stepwise updating

	Forward Stepwise								
Memory	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
10	0	0.002	0.03	0.162	0.414	0.692	0.862	0.93	0.988
20	0	0	0.004	0.006	0.018	0.064	0.118	0.166	0.18
30	0	0	0.002	0.002	0.004	0.008	0.006	0.024	0.028
40	0	0	0	0	0	0	0	0	0.002
50	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0

Fig. 4c: Forecast rules are AR(2) (linear) with LASSO updating

	LASSO								
Memory	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
10	0	0	0.006	0.048	0.162	0.308	0.516	0.646	0.756
20	0	0	0	0.004	0.016	0.042	0.09	0.092	0.132
30	0	0	0	0	0.002	0.002	0.01	0.012	0.026
40	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0

Figure 5: Frequency of runs out of 500 in which an explosive bubble forms within 2,000 trading periods for 90 combinations of memory M and learning rate pupdate. Figures 5a-5c represent different updating technologies for a quadratic AR(3) forecast rule (3).

Fig. 5a: Forecast rules are AR(3) (quadratic) with OLS updating

OLS

		Pupdate								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Memory	10	1	1	1	1	1	1	1	1	1
	20	1	1	1	1	1	1	1	1	1
	30	0.976	1	1	1	1	1	1	1	1
	40	0.866	0.982	0.99	1	1	0.996	0.998	1	1
	50	0.734	0.886	0.948	0.974	0.972	0.984	0.986	0.996	0.99
	60	0.552	0.726	0.832	0.88	0.914	0.952	0.942	0.962	0.974
	70	0.432	0.588	0.656	0.766	0.826	0.854	0.866	0.874	0.92
	80	0.35	0.48	0.576	0.646	0.732	0.748	0.754	0.784	0.81
	90	0.262	0.374	0.48	0.534	0.626	0.616	0.642	0.656	0.678
	100	0.186	0.278	0.38	0.416	0.52	0.492	0.562	0.566	0.586

Fig. 5b: Forecast rules are AR(3) (quadratic) with Forward Stepwise updating

Forward Stepwise

		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Memory	10	1	1	1	1	1	1	1	1	1
	20	0.324	0.7	0.846	0.912	0.952	0.988	0.992	0.996	0.998
	30	0.074	0.206	0.326	0.382	0.468	0.558	0.63	0.664	0.744
	40	0.034	0.068	0.106	0.16	0.232	0.248	0.28	0.334	0.348
	50	0.014	0.042	0.046	0.104	0.088	0.128	0.118	0.15	0.176
	60	0.016	0.022	0.028	0.04	0.056	0.05	0.062	0.074	0.094
	70	0.01	0.008	0.016	0.008	0.028	0.044	0.042	0.04	0.054
	80	0.006	0.01	0.01	0.014	0.016	0.036	0.024	0.032	0.032
	90	0	0	0.004	0.008	0.01	0.02	0.018	0.02	0.022
	100	0.002	0.002	0.002	0.012	0.01	0.01	0.008	0.008	0.016

Fig. 5c: Forecast rules are AR(3) (quadratic) with LASSO updating

LASSO

		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Memory	10	0	0.01	0.142	0.36	0.6	0.8	0.868	0.938	0.978
	20	0.004	0.024	0.068	0.094	0.19	0.284	0.354	0.398	0.466
	30	0.004	0.048	0.072	0.102	0.116	0.154	0.192	0.208	0.244
	40	0.012	0.05	0.07	0.072	0.07	0.124	0.128	0.154	0.156
	50	0.006	0.038	0.056	0.058	0.068	0.08	0.088	0.122	0.104
	60	0.01	0.034	0.028	0.048	0.036	0.06	0.068	0.08	0.078
	70	0	0.022	0.032	0.042	0.032	0.05	0.046	0.064	0.048
	80	0.01	0.02	0.016	0.014	0.022	0.046	0.03	0.054	0.06
	90	0.008	0.016	0.026	0.01	0.026	0.032	0.02	0.024	0.05
	100	0	0.008	0.006	0.012	0.028	0.018	0.028	0.032	0.022

Figure 6: Frequency of runs out of 500 in which an explosive bubble forms within 2,000 trading periods for 90 combinations of memory M and update rate pupdate. Figures 6a-6c represent different updating technologies for a quartic AR(6) forecast rule (number of regressors $P = 31$).

Fig. 6a: Forecast rules are AR(5) (quartic) with LASSO updating

		LASSO								
		Pupdate								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Memory	10	0.008	0.146	0.49	0.734	0.864	0.928	0.972	0.992	0.994
	20	0	0.018	0.074	0.142	0.212	0.286	0.336	0.38	0.464
	30	0	0.006	0.022	0.044	0.082	0.114	0.13	0.13	0.144
	40	0.042	0.082	0.144	0.21	0.19	0.278	0.316	0.306	0.366
	50	0.032	0.036	0.112	0.104	0.166	0.174	0.192	0.21	0.248
	60	0.014	0.05	0.076	0.082	0.086	0.122	0.098	0.14	0.158
	70	0.012	0.024	0.042	0.058	0.09	0.092	0.094	0.092	0.09
	80	0.002	0.016	0.02	0.024	0.054	0.04	0.068	0.078	0.074
	90	0.004	0.022	0.018	0.024	0.044	0.042	0.034	0.064	0.04
	100	0.004	0.014	0.01	0.028	0.022	0.028	0.054	0.042	0.042

Fig. 6b: Forecast rules are AR(5) (quartic) with Elastic Net (α=0.5) updating

		Elastic Net								
		Pupdate								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Memory	10	0.002	0.122	0.45	0.688	0.856	0.926	0.98	0.984	0.994
	20	0	0.016	0.066	0.12	0.172	0.238	0.292	0.338	0.404
	30	0	0.004	0.012	0.044	0.064	0.074	0.098	0.104	0.128
	40	0.032	0.088	0.144	0.228	0.244	0.25	0.288	0.258	0.308
	50	0.026	0.048	0.088	0.098	0.166	0.184	0.232	0.208	0.218
	60	0.012	0.05	0.076	0.088	0.08	0.12	0.11	0.14	0.162
	70	0.014	0.03	0.038	0.052	0.09	0.086	0.088	0.102	0.098
	80	0.006	0.02	0.022	0.032	0.046	0.04	0.062	0.076	0.072
	90	0.004	0.022	0.024	0.028	0.042	0.048	0.044	0.052	0.038
	100	0.006	0.008	0.01	0.026	0.012	0.026	0.054	0.054	0.04

Fig. 6c: Forecast rules are AR(5) (quartic) with Ridge updating

		Ridge								
		Pupdate								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Memory	10	0	0	0.002	0.002	0.016	0.032	0.082	0.128	0.206
	20	0	0	0	0	0	0	0	0	0.002
	30	0	0	0	0	0	0	0	0	0
	40	0	0.002	0.004	0.018	0.016	0.022	0.038	0.03	0.054
	50	0	0.004	0	0.006	0.01	0.014	0.012	0.02	0.024
	60	0	0.002	0.002	0.002	0.004	0.008	0.004	0.008	0.012
	70	0	0	0	0	0.002	0	0	0.006	0.004
	80	0	0	0	0.002	0.002	0.002	0.002	0	0.004
	90	0	0	0	0	0	0	0	0.002	0.002
	100	0	0	0	0	0	0	0	0	0.002

References

[1] Abadie, Alberto, and Maximilian Kasy 2017, The risk of machine learning. Working Paper, Department of Economics, MIT.

[2] Andersen, Jorgen Vitting, and Didier Sornette, 2008, A Mechanism for Pockets of Predictability in Complex Adaptive Systems. Working paper, CNRS UMR.

[3] Bai, Jushan, and Sarena Ng, 2008, Forecasting Economic Time Series Using Targeted Predictors. Journal of Econometrics 146. 304–317.

[4] Belloni, Alexandre, Victor Chernozhukov, and Christian Hansen, 2014, High-Dimensional Methods and Inference on Structural and Treatment Effects. Journal of Economic Perspectives 28:2, Spring 2014, 29–50.

[5] Branch, William A., and Georges W. Evans, 2006. Intrinsic Heterogeneity in Expectation Formation. Journal of Economic Theory 127 (1), 264?295.

[6] Branch, William A., and George W. Evans, 2011, Learning about Risk and Return: A Simple Model of Bubbles and Crashes. American Economic Journal: Macroeconomics 3. 159–191.

[7] Branch, William A., Bruce McGough, and Mei Zhu, October 2017. Statistical Sunspots. Manuscript, U.C. Irvine.

[8] Brogaard, Jonathan, Terrence Hendershott, and Ryan Riordan, 2014, High-frequency trading and price discovery. Review of Financial Studies 27, 2267–2306.

[9] Bullard, James, George W. Evans, and Seppo Honkapohja, 2008, Monetary Policy, Judgment, and Near-Rational Exuberance. American Economic Review, 98:3, 1163–1177

[10] Chaboud, Alain, Benjamin Chiquoine, Erik Hjalmarsson, and Clara Vega, 2014. Rise of the machines: Algorithmic trading in the foreign exchange market. The Journal of Finance, 69:5, 2045–2084.

[11] Chinco, Alexander M., Adam D. Clark-Joseph, and Mao Ye, 2019, Sparse Signals in a Cross Section of Returns. The Journal of Finance Vol LXXIV, No. 1, February 2019.

[12] Cho, In-Koo, and Kenneth Kasa, 2015, Learning and Model Validation. Review of Economic Studies 82:1. 45–82.

[13] De Luca, Marco and Dave Cliff, 2011. Human-agent Auction Interactions: Adaptive Aggressive Agents Dominate. Proceedings of the international joint conference on Artificial Intelligence 22:1. 178–185.

[14] DeMiguel, V., F. Nogales, and R. Uppal, 2014. Stock Return Serial Dependence and Out-of-Sample Portfolio Performance. Review of Financial Studies 27(4), 1031–1073.

[15] Evans, Georges W., and Seppo Honkapohja, 2001. Learning and Expectations in Macroeconomics. Princeton U. Press.

[16] Fagiolo, Giorgio, Mattia Guerini, Francesco Lamperti, Alessio Moneta, Andrea Roventini, 2017, Validation of Agent- Based Models in Economics and Finance. LEM Working Paper, Institute of Economics, Scuola Superiore Sant'Anna

[17] Farmer, Leland E., Lawrence Schmidt, and Allan Timmermann, March 2019. Pockets of Predictability. Manuscript, University of Virginia.

[18] Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, 2017, Taming the factor zoo, Working Paper.

[19] Freyberger, Joachim, Andrea Neuhierl, and Michael Weber, 2019, Dissecting characteristics nonparametrically, Working Paper, Booth School of Business, U. Chicago.

[20] Gabaix, Xavier, 2014, A Sparsity-Based Model of Bounded Rationality. Quarterly Journal of Economics 129:4. 1661–1710.

[21] Georges, Christophre, 2008a, Bounded Memory, Overparameterized Forecast Rules, and Instability. Economics Letters 98. 129–135.

[22] Georges, Christophre, 2008b, Staggered Updating in an Artificial Financial Market. Journal of Economic Dynamics and Control 32. 2809–2825.

[23] Georges, Christophre, 2015, Risk Preference and Stability Under Learning. Economics Letters 132. 105-108.

[24] Giannone, Domenico, Michele Lenza and Giorgio E. Primiceri, 2018, Economic Predictions with Big Data: The Illusion of Sparsity. FRBNY Staff Report No. 847 April 2018.

[25] Gigerenzer, Gerd and Reinhard Selten, Editors, 2001. Bounded Rationality: The Adaptive Toolbox, MIT Press.

[26] Grandmont, Jean-Michel, 1998, Expectations Formation and Stability of Large Socioeconomic Systems. Econometrica, 66:4. 741–781.

[27] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science and Business.

[28] Hastie, Trevor, Robert Tibshirani, and Martin Wainwright. 2015. Statistical Learning with Sparsity: The LASSO and its Generalizations. CRC Press. Taylor and Francis Group.

[29] Hommes, Cars, and Mei Zhu, 2014. Behavioral Learning Equilibria. Journal of Economic Theory 150. 778–814

[30] Honkapohja, Seppo, and Kaushik Mitra, 2003. Learning with Bounded Memory in Stochastic Models. Journal of Economic Dynamics and Control, 27:8. 1437–57.

[31] Kaminski, Kathryn M. Short is the New Long. Alpha Simplex Group LLC, January 2019.

[32] Kandori, Michihiro, George J. Mailath and Rafael Rob Learning, 1993. Mutation, and Long Run Equilibria in Games. Econometrica 61: 1. 29–56.

[33] Kaya, Orcun, 2016. High-frequency trading: reaching the limits.Deutsche Bank Research Briefing.

[34] Kirilenko, Andrei A., and Andrew W. Lo, 2013. Moore's law versus murphy's law: Algorithmic trading and its discontents. The Journal of Economic Perspectives, 27:2. 51–72. 967–998.

[35] Kirilenko, Andrei A., Albert S. Kyle, Mehrdad Samadi, and Tugkan Tuzun. The Flash Crash: High-Frequency Trading in an Electronic Market. J. Finance LXXII No. 3, June 2017.

[36] Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, 2017. Shrinking the cross-section. Working Paper.

[37] LeBaron, Blake, 2012, Heterogeneous Gain Learning and the Dynamics of Asset Prices. Journal of Economic Behavior and Organization 83. 424–445.

[38] Lo, Andrew W., Harry Mamaysky, and Jiang Wang, 2000. Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. Journal of Finance Vol LV, No. 4. 1705–1765.

[39] Mullainathan, Sendhil, and Jann Spiess, 2017, Machine Learning: An Applied Econometric Approach. Journal of Economic Perspectives 31: 2. 87–106.

[40] Shiller, Robert, 2003. From Efficient Markets Theory to Behavioral Finance. The Journal of Economic Perspectives, 17:1, 83–104.

[41] Yang, Stephanie. Computer Models to Investors: Short Everything. Wall St. Journal, Jan 9, 2019.

[42] Zuckerman, Gregory, Rachael Levy, Nick Timiraos and Gunjan Banerji. Behind the Market Swoon: The Herdlike Behavior of Comuterized Trading. Wall St. Journal, Dec 25, 2018.