Market Stability with Machine Learning Agents*

Christophre Georges[†] Javier Pereira[‡] Department of Economics Hamilton College

July 18, 2020

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 suggest that even a high degree of attention to overfitting on the part of traders who are engaged in data mining is unlikely to entirely eliminate destabilizing speculation. They also 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).

1 Introduction

One source of mispricing in financial models is the overfitting of forecasting rules by market participants. We explore this mechanism in a simple agent-based model of a financial market. Our agents 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 in which they are operating, there is substantial room for learning to go further astray. In this context, the adoption of misspecified forecast rules can lead to additional excess volatility as well as market instability in the form of asset bubbles and crashes.

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)), or the inclusion of intrinsically irrelevant variables (Grandmont (1998), Bullard et al. (2008), Georges (2008a,b, 2015), LeBaron

*First version, July 2018. This version July 2020.
JEL Classification: D84 D83 D53 E44 G17 G14 G40
Keywords: Expectations, Machine Learning, LASSO, Agent-based Modeling, Asset Prices, Volatility
We thank Karmen Hutchinson, Noah Kramer, Dan Kraynak, Paul Magnus, David Urbont, and Ian Wilson for research assistance, and Blake LeBaron, Andrea Roventini, Doyne Farmer, Carlos Pérez, Kevin Lansing, participants at several conferences, and two anonymous referees for helpful comments and discussions. 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. Corresponding author.

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

(2012), Branch, McGough, and Zou (2019)). 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 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.

Misspecification via overparameterization is particularly relevant in financial markets today. Financial market participants increasingly find themselves in data rich environments and have access to sophisticated computational resources. They face model uncertainty as always, but now have many possible predictors at their disposal and the machine learning tools to consider a wide variety of functional forms for combining these predictors. I.e., they can now entertain very substantial model complexity. Machine learning (ML) is a set of high dimensional statistical techniques for combining potentially large numbers of predictors in flexible ways to generate predictions. Much attention is given to the substantial scope, introduced by this model complexity, for overfitting the training data, i.e., for adopting overparameterized rules that perform well in the training sample but poorly out of sample. Consequently, machine learning typically employs methods for model regularization (parameter shrinkage), predictor selection, and out of sample testing to temper the model complexity and mitigate the risk of ovefitting. (E.g., Hastie et al., (2009); Arnott et al. (2018); Gu et al., (2020); de Prado (2020)).

In the present paper, we focus on overparameterized forcasting rules and draw on the statistical and machine learning literature to endow our boundedly rational artificial traders with tools for model selection and regularization to adopt in their learning routines. 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. In our primary machine learning specification, the agents use the least absolute shrinkage and selection operator (LASSO), which performs both model selection and regularization using out of sample validation.

We work with a very simple artificial financial market based on Georges (2008a,b) in which all traders are chartists who use nonlinear autoregressive (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.

We compare market outcomes under LASSO learning with a corresponding baseline specification in which agents have fixed forecasting rules and update these with ordinary least squares (OLS). OLS learners overfit their forecasting rules to the data available to them. Trading on the resulting forecasts, they generate excess price volatility as well as market instability in the form of occasional explosive bubbles and crashes. This volatility and instability is increasing in the complexity of the agents' forecasting rules.

We find that the addition of LASSO model selection and regularization methods to the traders' learning algorithm substantially reduces, but does not eliminate, overfitting and the resulting excess volatility and market instability relative to the OLS baseline. Even though our LASSO learning traders are quite sophisticated about avoiding overfitting, they will still occasionally identify apparent predictability in the data, setting off a period of heightened volatility. These signals identified through LASSO tend to be sparse in the sense that only a subset of variables are identified as active predictors in a given episode. Interestingly, due to the intermittent nature of these discoveries of apparent predictability, LASSO acts as a powerful mechanism for generating fat tails and clustered volatility in returns, two near-universal empirical hallmarks of financial markets. The emergence of short-lived pockets of predictability with sparse signals is also supported by the empirical findings of Chinco, Clark-Joseph and Ye (2019) and Farmer, Schmidt and Timmermann (2019) for U.S. equities. Further, while LASSO's parameter shrinkage, variable selection, and out of sample validation cause estimated forecast rules to be both sparser and less extreme than rules estimated by OLS, we find that they are still extreme enough to produce occasional bubbles and crashes in the model. Our results suggest that even a high degree of attention to overfitting on the part of traders who are engaged in data mining is unlikely to entirely eliminate destabilizing speculation. We also consider several relevant machine learning alternatives to LASSO – stepwise selection, Ridge regression, and elastic nets – that similarly address the problem of overfitting and find similar results.

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 now 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). 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. In a recent global survey including FinTechs and incumbent financial institutions (Ryll et al., 2020), 60 percent of surveyed investment managers indicated that they are already using artificial intelligence (AI) in their investment process, with 55 percent implementing AI for asset price forecasting. These practitioners also overwhelmingly reported expecting AI to become increasingly important in contributing to investment returns in the next five years.

A central issue with regard to algorithmic trading and other forms of digital automation 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 purely on patterns in recent historical data may also deviate substantially from the efficient markets, rational expectations benchmark. Our results suggest that such deviations are unlikely to be fully eliminated as algorithms become increasingly sophisticated.

2 Literature Review and Further Motivation

Our current framework follows closely the model in 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 by ordinary least squares (OLS). Georges shows via simulation that when the 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.[1] 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 or other means to mitigate overfitting of the data.

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

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.[2]

This is precisely the type of problem that machine learning has been developed to handle. We consider several relevant ML techniques for our agents, but our primary focus is on LASSO. This 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 estimated model. Hence, LASSO performs variable selection, and as a result it yields sparse estimated models – that is, models that involve only a subset of the available variables – which are easier to analyze and interpret.

The extensive improvements in computing power including cloud computing, data storage, availability of big data and open source software have leveled the playing field for the widespread adoption of machine learning techniques in trading and investment management (Kaya (2016), Brummer and Yadav (2019), Buchanan (2019), Ryll et al. (2020)). While a large variety of ML methods are being employed in financial markets today, we argue that LASSO constitutes an excellent baseline framework for modeling market participants' use of these complex methods.

First, LASSO represents a powerful statistical tool for prediction and for combatting the risk of overfitting. An overfitted model is one that lacks parsimony, that is, it includes more predictors or more flexibility of functional form than is necessary, and as a result it is prone to act on noise rather than information, leading to inaccurate out-of-sample forecasts. Quantitative financial analysts are well aware of this problem. For example, none of the 31 quants at hedge funds and algorithmic trading firms interviewed by Hansen (2020) between 2017-2019 mentioned underfitting as an issue, but all of them expressed their concerns about the risk of overfitting.[3] There is a growing body of research showing that machine learning algorithms tend to outperform traditional models in financial market forecasting (e.g., Gu et al. (2020), Ryll and Seidens (2019)). Yet, no specific method has been shown to be universally better, and it is often found that simpler ML methods and models outperform more flexible ones – e.g., that shallower neural nets and sparser regression trees often outperform deeper ones.

ML methods generally address overfitting by attempting to exploit a bias-variance tradeoff, permitting some bias in order to reduce the variance of the estimates. In the face of uncertainty

[2] The analysis of Cho and Kasa (2015) provides some promise that more sophisticated model selection may lead to more efficient outcomes. However, their analysis is better suited for a single agent, such as a central bank, learning how to operate in a self-referential environment, rather than an ecology of heterogeneous traders searching for potentially short-lived unexploited predictability in a financial market.

[3] The following is an explanation of the problem of overfitting provided by one quantitative analyst interviewed in Hansen (2020): "A big mistake lots of people make is that they always want to use something really complex to solve a problem that does not need it. You always want to default to the model with the least complexity. You do not want to be fitting to an artifact in the dataset. You just want to make sure that what you are seeing is real and you are not actually kidding yourself with any kind of predictive power."

about which of many possible predictors are relevant at a given time, or how flexibly they should be combined to form predictions, they use regularization to improve robustness by simplifying the prediction model and making it less responsive to changes in the values of the predictors.[4]

Nevertheless, no method appears to be immune to overfitting. A case in point is the series of time-series forecasting competitions run by Spyros Makridakis (e.g., Makridakis et al., 2018), in which sophisticated machine learning methods have historically had difficulty beating extremely simple rules of thumb or averages across multiple methods.[5]

Hastie et al. (2009, 2015) have made the case that LASSO's bet on sparsity is a good one, particularly for high dimensional problems. If the true signal is sparse, then LASSO will perform well. If the true signal is dense, then it is likely that no method will perform well; in other words, LASSO may still perform relatively well!

Gu et al. (2020) find that ML methods that allow for nonlinearities and predictor interactions perform substantially better in predicting financial returns than methods that do not. We allow for both under LASSO by specifying the forecasting rules that our agents entertain below as polynomials that include cross products.

A second reason for us to favor LASSO is evidence that market participants care about interpretability. The interpretability and explainability of AI models are important and growing concerns for a wide range of actors in financial markets including investors, investment managers, financial institutions, regulatory authorities, and academic researchers (see for example Bracke et al. (2019), World Economic Forum (2019), Hansen (2020)). The complexity and opacity of some AI methods make it difficult to obtain an interpretable explanation for why the system has produced a given output (Burrell, 2016). With this opacity comes a loss of control and oversight and thus heightened risk.[6] Hansen (2020) finds in his interviews that quants routinely employ a simplicity heuristic, preferring simpler ML models over more complex ones, *both* to avoid overfitting *and* to increase control and interpretability.[7]

The model selection feature of LASSO in which only but the strongest predictors are chosen, is a good representation of this simplicity heuristic. Thus LASSO can act as a representation of either human based forecasting or algorithmic forecasting. In both cases, model parsimony serves both humans' desire to avoid overfitting and their cognitive limitations. The use of LASSO to model the effects of human cognitive limits is also supported by Gabaix (2014). He proposes a framework for modeling bounded rationality 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 LASSO

[4]Gigerenzer and Selten (2001) and Gigerenzer and Brighton (2009) have argued that a similar mechanism explains why more generally, in complex environments, simple, fast and frugal heuristics can produce better systemic results than more complicated behavioral heuristics. Similarly, Dosi et al. (2020) show via simulation that, in a complex macroeconomic environment, very simple forecast heuristics (in their case validated by experiments by Anufriev and Hommes (2012)) used by firms can lead to substantially better macroeconomic outcomes than when firms try to make more sophisticated forecasts.

[5]The winning team of the most recent competition, from Uber Technologies, combined ML and more conventional statistical methods.

[6]These are some reflections from market leaders and regulators extracted from the WEF (2019) report: BlackRock (Head of Liquidity Research): "The industry needs to address the problem of interpretability [...] before putting investor money into play."; Capital One Managing Vice-President): "We only roll out machine learning where we feel comfortable there are no biases or lack of transparency..."; BaFin (Regulator): "Ultimately, [explainability] is a prerequisite to secure the very principle of responsibility."; FSB (Financial Stability Board): "Efforts to improve the intepretability of AI may be important conditions not only for risk management, but also for greater trust from the general public."

[7]In Hanson's words, "... users of machine learning models are concerned that models might learn the wrong things and thus, become deceitful rather than informative. They use Ockham's razor as a heuristic to help strike a balance between simplicity and complexity, and interpretability and accuracy."

penalty. I.e., the agent makes sense of the world by focussing on specific signals in the data.[8] In a related 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 econometrically evaluate some of the more conventionally used patterns.

The interpretability of machine learning algorithms has also become an issue in academic research in finance. For example, Chinco, Clark-Joseph, and Ye (2019) use LASSO to identify interpretable pockets of predictability in stock returns that could be associated with changes in fundamentals. Similarly, Freyberger et al. (2019), Feng et al. (2020), and Kozak et al. (2020) have used LASSO to choose between various predictors that have already been identified by other researchers in the literature. Since ML is likely to play an increasingly active role in decision making and order execution, its use is likely also to be increasingly scrutinized by regulators. Consequently, while full model interpretability may not be always possible or required, transparency and interpretability are increasingly becoming necessary conditions for the broad and responsible adoption of these technologies.

It is worth noting that a key feature of machine learning methods, as practiced today compared to standard econometric analysis, is their focus on prediction rather than causal inference (Varian, 2014). Consequently, independent of the issue of opacity, there has been criticism of these 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.[9]

In our model, the traders believe that they may be in a complex, non-stationary environment, and so persistently search for predictable patterns in returns. To make our analysis crisp, the actual environment that we place them in is extremely simple, with stationary underlying fundamentals. Indeed, the very simple heuristic of forecasting by taking sample averages of past data actually corresponds to the MSV rational expectations equilibrium, and the traders could in principle learn this simple heuristic.[10] Because rational expectations are static in this model and all price volatility is excess volatility, our simple modeled environment provides a clean baseline for studying the effect of expectations formation on market volatility and stability. Further, the adoption of LASSO by our agents actually biases the agents toward adopting the MSV rule and not chasing spurious trends in the data. Nevertheless, believing that the environment may be non-stationary, they are constantly on the search for predictable patterns, including pockets of predictability that may be short-lived and idiosyncratic.

There is empirical evidence for such predictability in asset prices. Farmer, Schmidt, and Timmermann (2019) present evidence for daily U.S. stock returns based on time-varying regression methods, finding that return predictability is highly concentrated or local in time and that these pockets can be long lived. Andersen and Sornette (2008) find shorter pockets in NASDAQ daily data. At higher frequency, 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.

[8]Beunza and Stark (2012) document the cognitive overload faced by traders working at an investment bank's derivatives trading desk.

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

[10]Thus, our results share the 'simpler is better' flavor of Gigerenzer and Brighton (2009) and Dosi et al. (2020). However, this is not because the modeled environment is fundamentally complex, but rather because agents are simply willing to entertain the possibility that the environment is complex, as in Grandmont (1998).

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, as identified by LASSO estimation, 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 (measured in seconds) and are correlated with macroeconomic news.

Finally, there is a set of well documented stylized facts for empirical returns that appear to be robust, and indeed are often described as universal, for a variety of financial markets at various frequencies. These include that return autocorrelations are essentially zero except at very high frequencies, that returns distributions tend to display heavy tails, and that volatility tends to be clustered temporally (e.g., Mandelbrot (1963), Cont (2000), Lux and Alfarano (2016), LeBaron (2017)). More specifically, distributions of returns at moderate to high frequencies tend to display leptokurtosis with tail behavior approximated by a cubic power law,[11] and the autocorrelation functions of measures of volatility such as absolute or squared returns display slow decay (long-range dependence). We will discuss how the application of ML methods such as LASSO by our agents acts as a mechanism for producing or accentuating fat tails and volatility clustering in aggregate returns.

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, 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.[12]

There is a unique stationary rational expectations equilibrium (REE) $P_t = P^* \; \forall t$, 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 $x^* = P^* + \overline{d}$.[13]

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

[11] Estimated tail exponents of daily returns tend to fall between 2 and 4, and are often close to 3. While this indicates that the variance should be finite, convergence toward normality under time aggregation tends to be slow due to the non-iid nature of the returns.

[12] 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).

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

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[14] [15]

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

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 Forecast

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 $a_0 = x^*$.

Note that our general forecast rule specification (2) nests the MSV rule and thus the stationary REE rule. For example, for forecast rule (3), the stationary REE forecast rule would be given by $a = (a_0, a_1, a_2, a_3, a_4, a_5, a_6) = (x^*,0,0,0,0,0,0)$.

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

[14] 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.

[15] Note that we include cross products in the quadratic specification. Both non-linearity generally and interactions between predictors specifically were found by Gu et al. (2020) to improve empirical asset returns forecasts.

M (memory) periods. The new rule parameters a^i_{jt} minimize

$$\sum_{k=1}^{M}(x_{t-k} - F^i[x_{t-k}])^2 \tag{5}$$

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

We now allow the agents to consider the specification of their 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. For time t, the LASSO coefficient estimates (a^L_{0t},a^L_{1t},...) are the quantities that minimize

$$\sum_{k=1}^{M}(x_{t-k} - F[x_{t-k}])^2 + \lambda_t \cdot \sum_{j=1}^{p} |a_{jt}| \tag{6}$$

where $\lambda_t \geq 0$ is a tuning parameter to be determined separately and p is the number of regressors (predictors) in the regression.[16] 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.[17] 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 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.[18] To see this, consider the solution to equation (6) when there is only one predictor, x_t – i.e., the forecast rule (2) is linear AR(1). Suppressing the time and agent indexes,

$$a^L_1 = \begin{cases} Sign[a^O_1] \cdot (|a^O_1| - \lambda) & \text{if} |a^O_1| \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where a^O_1 represents the OLS coefficient for the single predictor computed as in equation (5). So if the OLS coefficient is more extreme than the LASSO's penalty parameter, $|a^O_1| > \lambda$, then the LASSO will estimate a shrunken version of the OLS coefficient, namely, $a^L_1 = sign[a^O_1] \cdot (|a^O_1| - \lambda)$.

[16] $p = 6$ in (3).

[17] 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 $a_j = 0$ for all $j \neq 0$.

[18] 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.

Whereas, if the OLS coefficient is less extreme than the penalty parameter, the LASSO will estimate a_1^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 λ and thus 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 Exogenous Artificial Data

To gain some intuition before turning to the full simulation model, we first consider forecast rule updating under OLS and LASSO when the feedback between forecasts and prices is shut down. Specifically, we generate a purely exogenous random series $x_t = x^* + \varepsilon_t$, where the ε_t are iid uniform on (-0.5, 0.5),[19] and consider the evolution of the estimated forecast rule parameters under OLS and LASSO for two specifications of the forecast rule (2).

The first forecast rule specification is linear AR(2):

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

Re-estimating (9) in each period under OLS and LASSO yields $(a_{0t}^O, a_{1t}^O, a_{2t}^O)$ and $(a_{0t}^L, a_{1t}^L, a_{2t}^L)$, the sequence of rules that minimize the loss functions (5) and (6) in each time period for the artificial x_t history. These would be the estimated rules adopted by all agents if they experienced

[19]This shock range matches that of the dividend series in the next section.

this artificial time series and updated their estimates in every period using these two model fitting approaches.

The second forecast rule specification is the moderately more complex quadratic AR(3) given in (3) above. Repeating the estimation for this specification yields a second set of parameter estimates $(a^O_{0t}, a^O_{1t}, a^O_{2t}, a^O_{3t}, a^O_{4t}, a^O_{5t}, a^O_{6t})$ and $(a^L_{0t}, a^L_{1t}, a^L_{2t}, a^L_{3t}, a^L_{4t}, a^L_{5t}, a^L_{6t})$.

We report on single example runs with a common exogenous x_t history for each of these forecast rule specifications and estimation methods. We consider memory M of 50, 100, and 500, and OLS and LASSO updating over 2000 rounds for each run. Estimated forecast rules under all specifications vary over time and are roughly centered on the stationary REE rule: $(a_0, a_1, a_2, ...) = (x^*, 0, 0, ...)$. Table 1 shows the standard deviations of the series of two of these estimated parameters: a_1 and a_2 (the coefficients on the first and second linear lags). We can see that the degree of variation of the parameter estimates under both estimation methods decreases as memory increases, i.e., as the estimation sample size increases. Further, the inclusion of additional regressors (from (2)) as we move from the simpler forecast rule (9) to the more complex forecast rule (3) increases the variability of parameter estimates for both methods (particularly for a_2 in this example). Increasing the complexity of the forecast rule increases the scope for overfitting each sample and so increases the variance of the estimators. So in these regards, the behavior of OLS and LASSO is similar. However, Table 1 also illustrates that LASSO tends to yield a substantial reduction in variance relative to OLS updating. This is to be expected, as LASSO performs variable selection and shrinkage of coefficient estimates in order to mitigate overfitting of the data.[20]

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	14.122
100	0.078	0.095	0.079	10.209
500	0.020	0.030	0.021	6.679

LASSO

M	Forecast Model			
	(9)		(3)	
	a_1	a_2	a_1	a_2
50	0.061	0.087	0.062	3.947
100	0.041	0.052	0.035	3.359
500	0.004	0.010	0.008	3.023

7 Model Simulations

We saw in the last section that if there was no feedback between forecasts and prices, there would still be some variation in the forecasts due to finite memories, and that the degree of variation depends on the memory of the agents, the complexity of the forecast rule, and the estimation

[20]The reduction in variance occurs at the expense of some increase in bias.

method used. We now turn to the full simulation model to examine how the feedback between forecasts and prices affects the volatility and the stability of the price dynamics. Recall that all price volatility is excess volatility in this model.

7.1 Excess Volatility Under OLS and LASSO Learning

We first consider the case in which all agents i use MSV forecast rules, namely, $F_t^i[x_{t+1}] = a_0^i$, and updating is by least squares. If it were the case that for all agents $a_0^i = x^*$, then the market would 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, as in Honkapohja and Mitra (2003). OLS and LASSO are the same under the MSV rule because there are no parameters to shrink; a_{0t}^i does not enter in the penalty function in (6).

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, with forecast rule parameter values fluctuating around (x^*,0,0,0,...) and the price levels fluctuating around P^*.

We also find 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 can be initiated by purely random sequences of dividend shocks but then become amplified by the feedback between the forecast rule estimates and the forecast-driven stock prices. Thus, these windows correspond to periods of punctuated volatility that emerge endogenously from periods of quiescence in the price dynamics. These 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 the traders 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 1 and 2. There, we show the evolution of price, return,[21] and average forecast rule coefficients for a representative run in which agents use the non-linear AR(3) forecast rule (3).[22] Comparing Figures 1 and 2 we can clearly see the correspondence for the LASSO run between the windows of predictability, in which more LASSO coefficients become non-zero, and the punctuated volatility of returns. A similar feedback between forecast rule estimation and returns is at work in the OLS run, but it is more pervasive and exhibits far less clustering in time.

[21] The returns reported here and in Table 2 below are log changes in price, $\ln P_t - \ln P_{t-1}$, i.e., continuously compounded returns net of dividends.

[22] The run is representative among runs in which an explosive bubble or crash does not emerge – see below.

Figure 1: Run with memory M=100 and pupdate=0.5. Forecast rules are of nonlinear AR(3) form as specified in (3), for which the number of predictors p is 6. $\overline{d} = 0.5$, r=0.05, and so P^*=10. Dividend shocks are uniformly distributed on (-0.05, 0.05) and occur with probability 0.8 in any period. Shown here are prices and returns for 3,500 rounds (rounds 4300-7800) taken from representative 10,000 round runs with OLS and LASSO updating. The runs share the same random seed.

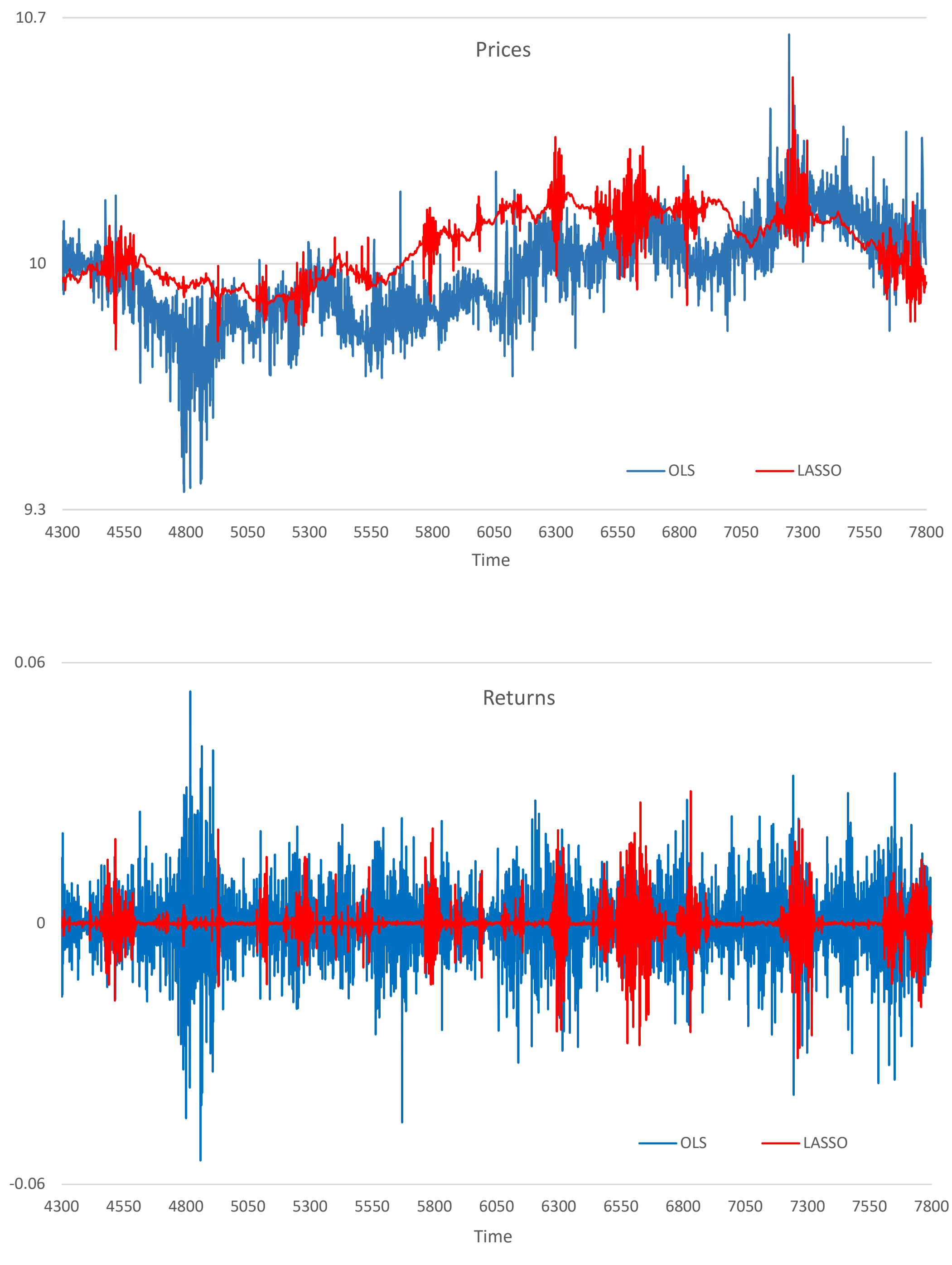
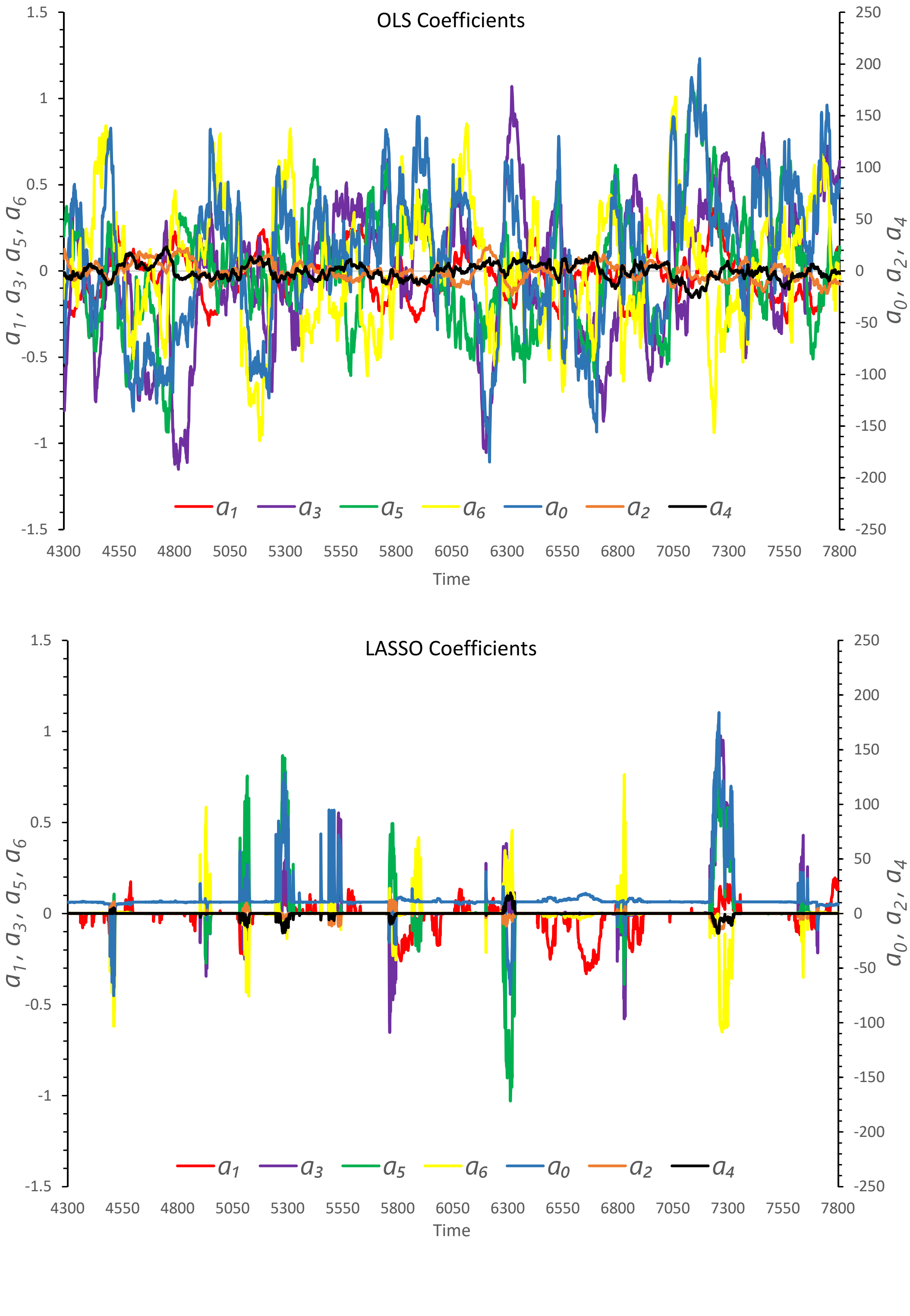


Figure 2: Additional output from previous specification. Shown here are average rule parameters ($\bar{a}_0$,...,$\bar{a}_6$) under OLS and LASSO updating. The corresponding stationary REE rule is (10.5,0,0,0,0,0,0).



Indeed, we can see here that LASSO's regularization properties act as a mechanism for accentuating fat tails and long-memory in volatility. As noted above, these are features of empirical returns seen across a wide range of financial markets and at a variety of time scales. We are not interested here in replicating these empirical regularities in any detail, as the model has been simplified to be able to specifically highlight the role of overfitting in generating excess volatility and the capacity of machine learning methods to mitigate that overfitting. However, the careful search for pockets of predictability using LASSO produces periods of quiescence punctuated by clusters of volatility, and so provide one mechanism for understanding the sources of these stylized facts.

In Table 2, we provide statistics for 10,000 periods of returns data generated by the same representative runs under OLS and LASSO learning that we exhibited above in Figures 1 and 2. In Figure 3, we display the corresponding returns distributions (along with their Gaussian counterparts) as well as autocorrelation function (ACF) plots for absolute and squared returns.

Table 2: Statistical Features of Returns

	Mean Price	Std Dev Price	Mean Return (*100)	Std Dev Return (*100)	Skewness	Kurtosis	OLS Hill Estimator	Q-Ratio
OLS	10.0264	0.1403	0.0002	0.8256	-0.0973	7.0833	2.9846 (0.0097)	1.7976
LASSO	9.9665	0.1119	-0.0003	0.3414	-0.0684	22.3553	1.7218 (0.0125)	2.7745

Same runs as displayed in Figures 1 and 2. Estimated std errors for Hill Estimator in parens.

In Table 2, we see that price is centered approximately on the stationary REE value of 10, and that average (excess) returns are essentially zero under both the OLS and LASSO simulations.[23] In both cases, the distribution of returns display greater probability mass in their center and tails than their respective Gaussian distribution counterparts, as can be seen also in the plots in Figure 3. We see that this excess kurtosis is substantially more pronounced under LASSO than under OLS learning. This is confirmed by two more robust measures of tail shape, the OLS Hill estimator, which provides an estimate of the power-law exponent for extreme returns,[24] and the Q-ratio, which is the ratio of the 1 percent returns quantile divided by the 5 percent quantile. This ratio takes a value of 1.3 for the Gaussian distribution, and larger values indicate relatively fatter tails. The Hill estimate for OLS is in line with the near-universal empirical findings of an approximate cubic law in (extreme) daily returns across a wide range of financial markets, whereas the LASSO returns exhibit even more extreme fat tails. Both estimates indicate that the 4th moments of the return distributions may well not be finite, and so sample kurtosis may not be a reliable measure. [25] Similarly, the Q-ratios indicate fatter than Gaussian tails in both cases, with relatively fatter tails under LASSO. Interestingly, the variance of returns is lower for LASSO than OLS, so the fatter

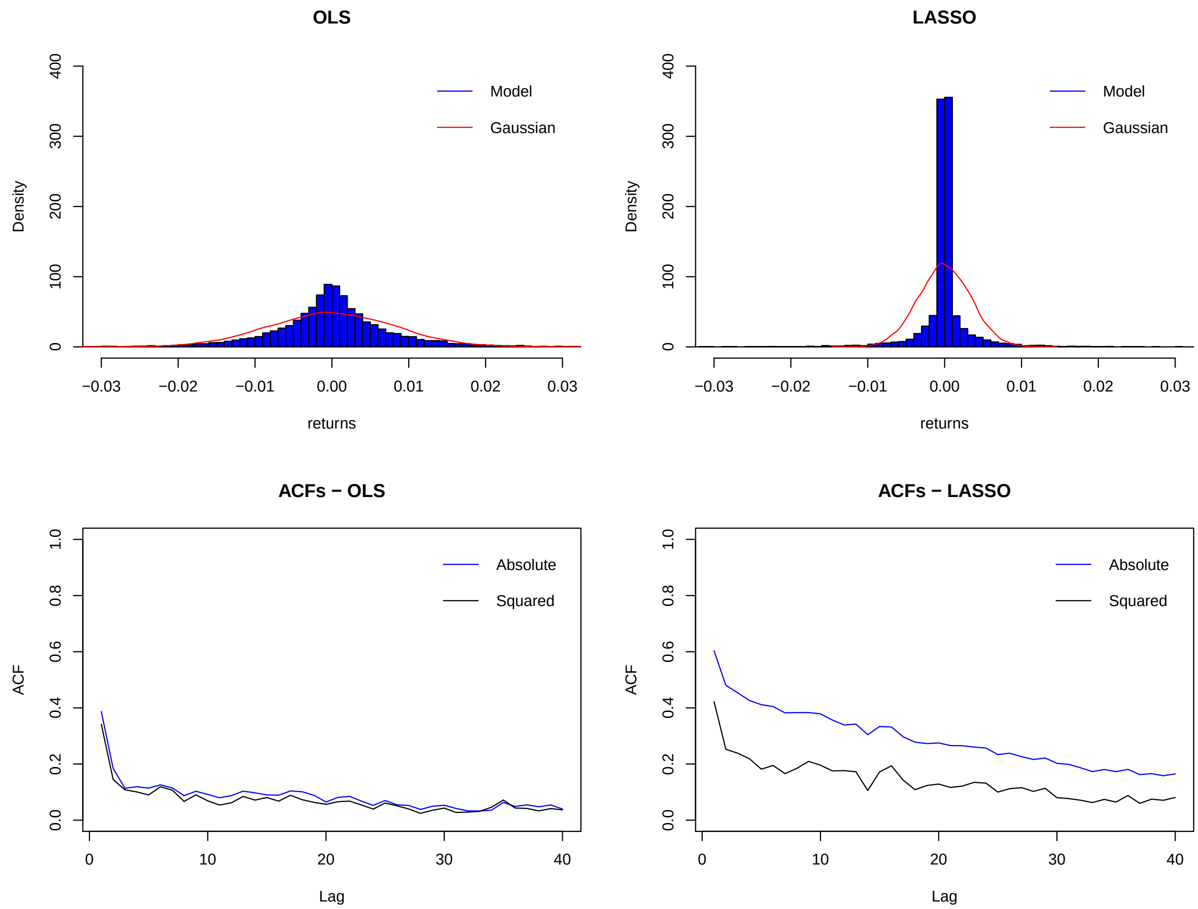
[23] These returns are period by period log changes in price, here multiplied by 100.

[24] Estimation of the tail exponent presents a classic bias-variance trade off problem. The estimate bias increases as we include more observations located deeper into the distribution. On the contrary, as we move out to include observations located further into the tail, we end up with a smaller sample, and we now face a problem of variance in the estimate. To address this, we estimate the OLS Hill Estimator which uses the bias adjusted estimator of the tail exponent developed in Huisman et al. (2001).

[25] Early power-law estimates that found evidence for the tail exponents of approximately 1.7 as we see in the LASSO experiment and which would indicate that even the second moment is not finite (Mandelbrot, 1963) have since been superseded by substantial evidence that tail exponents typically lie between 2 and 4, as we indicated in section 2.

tails for LASSO are accompanied by lower overall volatility.

Figure 3: Returns Distributions and ACF Plots for Absolute and Squared Returns



Both the OLS and LASSO experiments also generate clustered volatility in returns; large price changes have the tendency to be followed by large price changes. We can see in Figure 3 that the ACFs of both absolute and squared returns exhibit slow decay under both OLS and LASSO, with substantially slower decay (longer-range dependence) under LASSO.[26] [27]

So both OLS and LASSO learning generate fat tails in the market returns distributions and time dependence in volatility, both hallmarks of empirical financial markets, with these effects being more pronounced for LASSO. At the same time, LASSO experiments exhibit lower overall returns volatility. These comparisons between OLS and LASSO are intuitive. OLS overfits the forecast rules, leading traders to observe apparently predictable patterns in the data at each time step. LASSO updating is designed to combat overfitting by shrinking parameter estimates toward zero and frequently setting parameter estimates to zero, as guided by out-of-sample testing. Thus,

[26]Tail exponent estimates for these ACFs are 0.5278 for the OLS simulation and 0.3813 for the LASSO simulation. Empirically, these values tend to fall between 0.2 and 0.4, so here the LASSO returns series is consistent with the empirical regularity, while the ACF from the OLS simulation decays too rapidly. These estimates were computed as functions of the estimated Hurst exponent as in Lux and Ausloos (2002).

[27]The slower decay of the ACF for absolute returns than squared returns is also widely observed in empirical returns data (e.g., Ding et al., 1993).

under LASSO updating, the windows of predictability in which forecast rule coefficients become non-zero become infrequent and the average size of non-zero coefficients relatively smaller than under OLS learning. Only occasionally do agents perceive there to be predictable trends to try to exploit. With asynchronous updating, once some agents have acted on this perception, others may observe the same patterns and pile on, and the endogenous price response will feed back into this process creating a period of heightened volatility and complex dynamics. Thus, LASSO updating lowers overall volatility relative to OLS updating, but at the same time amplifies the fat tails and volatility clustering seen in this artificial market under OLS since it promotes periods of quiescence, in which traders follow an MSV rule or something close to it, punctuated by periods of speculation which are still tempered relative to the OLS case.[28] It is important to note that even the extreme tail behavior under LASSO is less extreme than under OLS, which will become apparent when we turn to the incidence of extreme bubbles and crashes below.

In summary, attempts by our market agents to mitigate overfitting through regularization and cross-validation using LASSO serves as a mechanism for generating and amplifying fat tails and volatility clustering. We reiterate that we are not attempting here to calibrate this model to empirical data, and indeed we see in the experiment above that LASSO learning, when practiced by all agents, produced substantially fatter tails than is observed empirically. This suggests that this mechanism can be a relatively powerful generator of these phenomena.

7.2 Large Bubbles and Crashes

The last section explored what we might call "normal" price and forecasting dynamics, when these dynamics remain stable. However, under both OLS and LASSO updating, we also observe the development of occasional extreme events, explosive price bubbles and crashes, in which the price level deviates substantially and persistently from its stationary REE value of $\bar{d}/r$. In this section, we report on 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 or crashes (extreme prices).[29] Restricting our attention for now to cases in which the number of predictors p is less than the memory M of the agents (which is a requirement for identification via OLS), we find that, for both OLS updating and LASSO, the incidence of bubbles and crashes is generally increasing in the speed of learning and decreasing in the traders' memory. As noted above, this has been previously documented for OLS updating by Georges (2008a,b, 2015). More rapid learning (more frequent re-estimation of the rules by the traders) causes the traders to herd on similar observed trends in recent returns, creating more destabilizing feedback between these trends and the average forecast. On the contrary, greater memory reduces the response of forecasts to recent trends and thus tempers the feedback between these rules and the pricing mechanism.

While these patterns are similar for both OLS and LASSO, here we also observe that the incidence of explosive bubbles and crashes is substantially higher for OLS than for LASSO, and that this difference is more pronounced as the complexity of the forecast rule (2) is increased. As documented in Georges (2008a,b, 2015), the incidence of these extreme events under OLS learning is increasing in the complexity of the forecast rules. However, this relationship is more ambiguous

[28]This switching between periods of MSV forecasts and overparameterized forecasts share some similarity with the switching between fundamentalists and chartists in popular two type heterogenous agent models (HAMs) (see e.g., Dieci and He, 2018) but is more continuous and organic here. Our forecast rules admit a continuum of agent types. All are chartists, but they can learn to follow MSV forecasts, and so could learn to be fundamentalists in this simple environment.

[29]We selected the pair of runs in Figures 1 and 2 from the roughly half of 500 pairs of runs with the same parameter settings for which neither the OLS or LASSO runs resulted in an explosive bubble or crash in the 2000 periods.

under LASSO learning. As the number of lags and the order of the polynomial in (2) are increased, LASSO has more predictors to select among, which can lead to increased or decreased volatility and incidence of bubbles and crashes.

These results are illustrated in Figures 4a - 4b and 5a - 5b, which indicate the frequency of the formation of explosive bubbles or crashes for different specifications of forecast rule and learning technology.[30] In Figure 4, agents employ low complexity linear AR(2) forecasting rules (9), with number of predictors $p = 2$, and in Figure 5, they employ modestly more complex non-linear AR(3) rules (3), with $p = 6$. In both cases, 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. Moving from Figure 4 to Figure 5, we see this difference becoming more striking as the number of predictors p grows. 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 and crashes driven by positive feedback between these rules and the pricing mechanism.[31]

[30] The extreme (bubble or crash) price thresholds are set at 0 and 20. $P^* = 10$ as in the example in Figure 1.

[31] Note that, while we exogenously specify the memory available to the traders in these experiments, actual traders may not wish to employ long histories of data if they perceive their environment to be non-stationary and are looking for momentary pockets of predictability. LeBaron (2012) considers endogenous heterogeneity in the weights placed by traders on recent data. Note also that for sufficiently high memories, bubbles and crashes will not be observed in our experiments but will emerge (with low frequency) over sufficiently long observation windows.

Figure 4: Frequency of runs out of 500 in which an explosive bubble or crash forms within 2,000 trading periods for 90 combinations of memory M and learning rate pupdate. Figures 4a-4c represent different updating technologies for the linear AR(2) forecast rule (9), for which $p = 2$. Model parameters are otherwise as in Figures 1 and 2.

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

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

Figure 5: Frequency of runs out of 500 in which an explosive bubble or crash forms within 2,000 trading periods for 90 combinations of memory M and learning rate pupdate. Figures 5a-5c represent the same model parameters and updating technologies as in Figure 4, but here agents employ the more complex quadratic AR(3) forecast rule (3), for which $p = 6$.

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

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

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.

As we discussed in Section 2, there are good reasons to favor the variable selection aspect of LASSO as our main specification over denser regularization methods. It certainly 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). However, we also believe that it squares well from a behavioral standpoint for modeling both human based and algorithmic forecasting.

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 the Bayesian Information Criterion (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

$$\sum_{k=1}^{M}(x_{t-k} - F[x_{t-k}])^2 + \lambda_t \cdot \sum_{j=1}^{p} a_{jt}^2 \qquad (10)$$

where $\lambda_t \geq 0$ is the tuning parameter to be determined separately using cross-validation. Comparing

(6) and (10), we observe that LASSO and Ridge regression have similar specifications. The only difference is that $|a_{jt}|$ is replaced by a_{jt}^2, i.e., the l_1 penalty is replaced with an l_2 penalty. This penalty is still small when (a_{1t}^R,a_{2t}^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_{0t}^E,a_{1t}^E,...) are the quantities that minimize

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

where $\alpha \in [0, 1]$ is a parameter that determines the balance between Ridge and LASSO penalties and $\lambda_t \geq 0$ is the penalty tuning parameter chosen using cross-validation.[32]

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 (2019), there is no one method for regularization that is universally optimal.[33]

8.4 Simulations

Simulations confirm our intuition that, when practiced by our agents, forward 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 or crashes tend to fall between the cases of OLS and LASSO updating. The intermediate incidence of bubbles and crashes is illustrated in Figures 4c and 5c. LASSO protects better against overfitting than stepwise selection by adding parameter shrinkage to parameter selection, and so yields less self reinforcing trend chasing.[34]

Figures 6a-6c illustrate the frequency of the formation of bubbles or crashes for the two denser regularization methods for the case of quartic AR(6) forecast rules ($p = 31$). In this example, 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 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 substantially 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 or lower instabilities as the number of regressors climbs into the hundreds.

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

[33] Abadie and Kasy (2019) 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.

[34] Since the number of predictors in this experiment is modest, we also perform simulations with agents using exhaustive selection, rather than forward stepwise selection, and find a similar intermediate incidence of bubbles and crashes, which tends to be somewhat higher (rather than lower) than under stepwise selection. Exhaustive selection quickly becomes computationally infeasible as the number of predictors increases (see e.g., Aragones et al., 2005), and so we focus on stepwise selection above.

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, while increasing the complexity to 26 lags and 25 powers ($p = 926$) yields modestly lower instability for LASSO than Ridge. 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. We find in our simulations that, for $p > M$, as p grows large, the performance of Ridge continues to slowly deteriorate, while the performance of LASSO stabilizes and can eventually even improve on the margin.[35]

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.

[35] 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 observed in Figure 6, for which the number of predictors (regressors) in the forecasting rule is $p = 31$.

Figure 6: Frequency of runs out of 500 in which an explosive bubble or crash forms within 2,000 trading periods for 90 combinations of memory M and learning rate pupdate. Figures 6a-6c represent different updating technologies for a quartic AR(6) forecast rule (number of regressors $p = 31$). Model parameters are otherwise the same as in the previous Figures.

Fig. 6a: Forecast rules are quartic AR(6) 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 quartic AR(6) 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 quartic AR(6) 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

It is also worth noting that, while Ridge estimation does not set forecast rule parameters exactly equal to zero, in our simulations, it does often set some parameters very close to zero, thus performing a kind of near model selection. Consequently, we tend to observe clustering of volatility in average forecast rule parameters and in returns that are closer to the LASSO case than the OLS case, again indicating that dynamics are driven by apparent pockets of predictability, in this case with near-sparse signals.

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 is unlikely to entirely eliminate destabilizing speculation.

References

[1] Abadie, Alberto, and Maximilian Kasy, 2019. Choosing among Regularized Estimators in Empirical Economics: The Risk of Machine Learning. Review of Economics and Statistics 101:5. p.743-762.

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

[3] Anufriev, Michail and Cars Hommes. 2012. Evolutionary Selection of Individual Expectations and Aggregate Outcomes in Asset Pricing Experiments. American Economic Journal: Microeconomics, 4(4). 35-64.

[4] Aragones, Enriqueta, Itzhak Gilboa, Andrew Postlewaite, and David Schmeidler. 2005. Fact-Free Learning. American Economic Review 95:5. 1355–1368.

[5] Arnott, Rob, Campbell R. Harvey, and Harry Markowitz. 2018. A Backtesting Protocol in the Era of Machine Learning. Manuscript, Duke U.

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

[7] 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.

[8] Beunza D and Stark D (2012) From dissonance to resonance: Cognitive interdependence in quantitative finance. Economy and Society 41(3): 383–417.

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

[10] 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.

[11] Branch, William A., Bruce McGough, and Mei Zhu, May 2019. Statistical Sunspots. Manuscript, U.C. Irvine.

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

[13] Brummer, Chris and Yesha Yadav, 2019. Fintech and the Innovation Trilemma, Georgetown Law Journal. 107:2.

[14] Buchanan, Bonnie G., (2019) Artificial Intelligence in Finance. Manuscript. The Alan Turing Institute.

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

[16] Burrell J., 2016, How the machine 'thinks': Understanding opacity in machine learning algorithms. Big Data and Society 3(1). 1-12.

[17] 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.

[18] 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.

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

[20] Cont, Rama. Empirical properties of asset returns:stylized facts and statistical issues. 2001. Quantitative Finance. Vol 1. 223–236.

[21] de Prado, Marcos Lopes. March 2020. Overfitting: Causes and Solutions. Manuscript (seminar slides), Cornell U.

[22] Diece, Roberto and Xue-Zhong He. 2018. Heterogeneous Agent Models in Finance. In Cars Hommes and Blake LeBaron eds. Handbook of Computational Economics. Volume 4. Heterogeneous Agent Modeling. Elsevier, North-Holland. 257–328.

[23] Ding, Zhuanxin, Clive W.J.Granger, Robert F.Engle. 1993. A long memory property of stock market returns and a new model. Journal of Empirical Finance Volume 1, Issue 1,f Pages 83-106

[24] Dosi, Giovanni, Mauro Nepolitano, Andrea Roventini, Joseph E. Stiglitz, and Tania Treibich. 2020. Rational Heuristics? Expectations and Behaviors in Evolving Economies with Heterogeneous Interacting Agents. Economic Inquiry. 58:3. 1487-1516.

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

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

[27] Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, 2020, Taming the Factor Zoo, J. Finance. 75:3. 1327-1370

[28] Freyberger, Joachim, Andrea Neuhierl, and Michael Weber, 2019, Dissecting Characteristics Nonparametrically, Working Paper, Booth School of Business, U. Chicago.

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

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

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

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

[33] 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.

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

[35] Gigerenzer, Gerd and Henry Brighton. 2009. Homo Heuristicus: Why Biased Minds Make Better Inferences. Topics in Cognitive Science I, 107–143.

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

[37] Gu, Shihao, Bryan Kelly and Dacheng Xiu. 2020. Empirical Asset Pricing via Machine Learning. Review of Economic Studies. 33: 5. 2223–2273.

[38] Hansen, Kristian Bondo. The virtue of simplicity: On machine learning models in algorithmic trading. Big Data and Society January–June: 1–14

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

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

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

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

[43] Huisman, R., Koedijk, K. G., Kool, C. J. M. and Palm, F. 2001, Tail-index estimates in small samples. Journal of Business and Economic Statistics 19, 208–216.

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

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

[46] Kaya, Orcun, 2016. High-frequency Trading: Reaching the Limits. Deutsche Bank Research Briefing.

[47] 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.

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

[49] Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, 2020. Shrinking the cross-section. Journal of Financial Economics.135:2. 271-292

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

[51] LeBaron, Blake, 2017. The Optimality of Short Term Rules of Thumb at Long Horizons for an Agent-based Financial Market. Manuscript, Brandeis U.

[52] 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.

[53] Lux, Thomas and Simone Alfarano 2016. Financial power laws: Empirical evidence, models, and mechanisms. Chaos, Solitons and Fractals 88 3–18.

[54] Lux T., and M. Ausloos. Market fluctuations i: scaling, multiscaling and their possible origins. In: Bunde A, Kropp J, Schellnhuber H, edi- tors. The science of disasters: climate disruptions, heart attacks, and market crashes. Springer; 2002. ISBN 9783540413240 .

[55] Makridakis, Spyros, Evangelos Spiliotis and Vassilios Assimakopoulos, 2018. The M4 Competition: Results, Findings, Conclusion and Way Forward. International Journal of Forecasting 34, 802–808

[56] Mandelbrot, B. 1963 The variation of certain speculative prices. J. Business XXXVI 392–417

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

[58] Ryll, Lukas, and Sebastian Seidens, 2019. Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey, Manuscript, Centre for Alternative Finance, University of Cambridge, Paper 1906.07786, arXiv.org.

[59] Ryll, Lukas, Mary Emma Barton, Bryan Zheng Zhang, andJesse McWaters. 2020. Transforming Paradigms: A Global AI in Financial Services Survey. report Cambridge Centre for Alternative Finance and World Economic Forum,

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

[61] Varian, H. R. 2014. Big data: New tricks for econometrics. Journal of Economic Perspectives, 28(2), 3-28.

[62] World Economic Forum, 2019. Navigating Uncharted Waters: A Roadmap to Responsible Innovation with AI in Financial Services. Report WEF and Deloitte.

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

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