Adaptive Learning
for Applied Macroeconomics

A thesis submitted to The University of Manchester for the
degree of Doctor of Philosophy in the Faculty of Humanities

Jaqueson Kingski Galimberti

School of Social Sciences

2013
Contents

List of Figures ......................................................... 5
List of Tables .......................................................... 7
List of Abbreviations .................................................. 10
Abstract ................................................................. 11
Declaration and Copyright Statement ................................. 13
Acknowledgments ....................................................... 15
Preface 17

1 Estimation framework and the learning algorithms 21
   1.1 Introduction ...................................................... 21
   1.2 Digression into algorithms origins ............................ 23
   1.3 State-space unifying framework ............................... 31
   1.4 Design of learning-to-forecast exercises .................... 39
   1.5 Quality of learning forecasts relative to surveys .......... 49
   1.6 Concluding remarks ............................................ 56

2 The initialization of the learning algorithms 59
   2.1 Introduction ...................................................... 59
   2.2 Statistical analysis of learning algorithms ................... 61
   2.3 Initialization methods ......................................... 66
   2.4 Comparative simulation analysis .............................. 75
   2.5 Empirical application .......................................... 85
   2.6 Concluding remarks ............................................ 91

3 The determination and calibration of the learning gains 93
   3.1 Introduction ...................................................... 93
   3.2 Theoretical background ....................................... 94
   3.3 Learning gain calibrations .................................... 99
   3.4 Results and discussion ........................................ 107
   3.5 Concluding remarks ............................................ 120

4 The choice of a representative learning algorithm 123
   4.1 Introduction ...................................................... 123
   4.2 Literature background ........................................ 126
   4.3 Choice between single algorithms ............................ 133
   4.4 Learning with paired algorithms ............................. 142
   4.5 Concluding remarks ............................................ 153
List of Figures

1.1 Data series on US inflation, output growth, and survey forecasts.  
2.1 Illustration of filtering, smoothing, and the initialization routine.  
2.2 Example of inflation-like artificially generated series.  
2.3 MSD learning curves for inflation-like artificial data.  
2.4 Estimates of inflation autoregressive coefficient.  
3.1 Evolution of recursive gain calibrations through time.  
3.2 Evolution of adaptive gains through time.  
3.3 Algorithms’ forecasting accuracy by gain aggregated over time.  
4.1 Evolution of algorithms’ forecasting accuracy through time.  
4.2 Evolution of time-varying weight calibrations through time.  
C.1 Example of growth-like artificially generated series.  
C.2 MSD learning curves for growth-like artificial data.  
C.3 Estimates on growth equation coefficients.  
C.4 Evolution of adaptive weights through time.
List of Tables

1.1 Data and forecasts statistics. .......................... 51
1.2 Learning versus survey’s accuracy forecasting inflation. 53
1.3 Learning versus survey’s accuracy forecasting growth. 54

2.1 Comparative of initialization methods. .................... 67
2.2 Calibration of parameters for simulation with inflation-like data. 77
2.3 Average MSDs after initializations on inflation-like data. ... 82
2.4 Average MSEs after initializations on inflation-like data. ... 84
2.5 Statistics on inflation forecasts. .......................... 90

3.1 Fixed gain calibrations. ................................. 108
3.2 Data and forecasts statistics by calibration. ............... 114
3.3 Hit rates comparing gain selection methods. ............... 117
3.4 Hit rates comparing gain determination assumptions. ....... 119

4.1 Hit rates comparing the algorithms. ....................... 137
4.2 Hit rates comparing LS vs. survey forecasts accuracy. .... 139
4.3 Fixed weight calibrations. ............................... 146
4.4 Hit rates comparing the paired algorithms. ............... 149
4.5 Hit rates comparing best single and paired algorithms. ... 151
4.6 Hit rates comparing paired algs. vs. survey forecasts accuracy. 152

C.1 Calibration of parameters for simulation with growth-like data. 178
C.2 Average MSDs after initializations on growth-like data. .... 180
C.3 Average MSEs after initializations on growth-like data. .... 181
C.4 Statistics on growth forecasts. ........................... 184
C.5 Illustrative excerpt of output of comparative evaluation. ... 185
C.6 Data and forecasts statistics for paired algorithms. ......... 186
C.7 Hit rates on paired algorithms gain/weight selection methods. 187
C.8 Hit rates on paired algorithms gain/weight determination. ... 188
List of Abbreviations

ALM Actual law of motion
BEA Bureau of Economic Analysis
CorA Correlation with actuals
CorS Correlation with surveys
DGP Data generating process
DM Diebold and Mariano test
DSGE Dynamic stochastic general equilibrium
GDP Gross domestic product
GNP Gross national product
GW Giacomini and White test
HAC Heteroskedasticity and autocorrelation consistent
HY Hybrid
LS Least squares
LSA Least squares with adaptive gains
MSD Mean-square deviation
MSE Mean-square error
MSFCE Mean squared forecast comparison error
MSFE Mean squared forecast error
MX Mixing
OLS Ordinary least squares
PLM Perceived law of motion
RBC Real business cycle
RE Rational expectations
SD Steepest descent
SG Stochastic gradient
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGA</td>
<td>Stochastic gradient with adaptive gains</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-noise ratio</td>
</tr>
<tr>
<td>SPF</td>
<td>Survey of professional forecasters</td>
</tr>
<tr>
<td>US</td>
<td>United States</td>
</tr>
<tr>
<td>VAR</td>
<td>Vector autoregression</td>
</tr>
</tbody>
</table>
Abstract

The literature on bounded rationality and learning in macroeconomics has often used recursive algorithms to depict the evolution of agents’ beliefs over time. In this thesis we assess this practice from an applied perspective, focusing on the use of such algorithms for the computation of forecasts of macroeconomic variables. Our analysis develops around three issues we find to have been previously neglected in the literature: (i) the initialization of the learning algorithms; (ii) the determination and calibration of the learning gains, which are key parameters of the algorithms’ specifications; and, (iii) the choice of a representative learning mechanism. In order to approach these issues we establish an estimation framework under which we unify the two main algorithms considered in this literature, namely the least squares and the stochastic gradient algorithms. We then propose an evaluation framework that mimics the real-time process of expectation formation through learning-to-forecast exercises. To analyze the quality of the forecasts associated to the learning approach, we evaluate their forecasting accuracy and resemblance to surveys, these latter taken as proxy for agents’ expectations. In spite of taking these two criteria as mutually desirable, it is not clear whether they are compatible with each other: whilst forecasting accuracy represents the goal of optimizing agents, resemblance to surveys is indicative of actual agents behavior. We carry out these exercises using real-time quarterly data on US inflation and output growth covering a broad post-WWII period of time.

Our main contribution is to show that a proper assessment of the adaptive learning approach requires going beyond the previous views in the literature about these issues. For the initialization of the learning algorithms we argue that such initial estimates need to be coherent with the ongoing learning process that was already in place at the beginning of our sample of data. We find that the previous initialization methods in the literature are vulnerable to this requirement, and propose a new smoothing-based method that is not prone to this critic. Regarding the learning gains, we distinguish between two possible rationales to its determination: as a choice of agents; or, as a primitive parameter of agents learning-to-forecast behavior. Our results provide strong evidence in favor of the gain as a primitive approach, hence favoring the use of surveys data for their calibration. In the third issue, about the choice of a representative algorithm, we challenge the view that learning should be represented by only one of the above algorithms; on the basis of our two evaluation criteria, our results suggest that using a single algorithm represents a misspecification. That motivate us to propose the use of hybrid forms of the LS and SG algorithms, for which we find favorable evidence as representatives of how agents learn. Finally, our analysis concludes with an optimistic assessment on the plausibility of adaptive learning, though conditioned to an appropriate treatment of the above issues. We hope our results provide some guidance on that respect.
Declaration and Copyright Statement

Declaration

I declare that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

(i) The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

(ii) Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

(iii) The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

(iv) Further information on the conditions under which disclosure, publication and commercialization of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy¹, in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations² and in The University’s policy on presentation of Theses.

¹See http://www.campus.manchester.ac.uk/medialibrary/policies/intellectual-property.pdf.
²See http://www.manchester.ac.uk/library/aboutus/regulations.
Acknowledgments

This thesis is not about debt, but its making has put me in deficit with many people. First of all, I thank to Michele Berardi, my PhD supervisor, for the many insightful discussions, and constructive comments, and for the invaluable academic guidance. I also thank to the other great scholars that I have had the opportunity to interact during my PhD studies. Particularly I thank to: Anne Villamil, who co-supervised my doctoral studies; George Bratsiotis, who acted as an internal evaluator during my studies; George Evans and Ralf Becker for accepting to take part in my thesis evaluation panel, and for their valuable comments on my works. I also extend my acknowledgments to the remaining academic and administrative staff of the Manchester University Economics DA, and to the PhD colleagues of all cohorts that shared the ALB open plan neighborhood during the period. Obviously, I exempt everyone else other than me as responsible for any remaining error(s) in this thesis.

I also benefited from many opportunities to present my works in conferences and seminars. I extend my thanks for the comments received from participants of the following conferences/seminars: the 19th International Conference Computing in Economics and Finance (Vancouver); the 21th Symposium of the Society for Nonlinear Dynamics and Econometrics (Milan); the KOF Swiss Economic Institute seminars (Zürich); the Lund University Economics Department seminars (Lund); the CGBCR seminar series, and the 2012 Manchester Doctoral Conference (Manchester).

I also thank to the Manchester University Economics DA, and the Royal Economic Society, for providing me the financial assistance that made possible my participation in the conferences above. I also gratefully acknowledge financial support received from The Capes Foundation (BEX 0597/10-4 process), a sponsoring agency associated to the Ministry of Education of Brazil, for the realization of my PhD overseas.

Finally, and for sure most importantly, I thank to those that give a meaning to all ventures: to my family, whose guidance for life has never faded to the cross-Atlantic distance; and to Joyce and Sofia, my wife and my daughter, whose unconditional support and love have turned the rocky path into a joyful journey. As always, my accomplishments are yours, and I dedicate this thesis to you.
Preface

This thesis is about the use of adaptive learning mechanisms in order to represent how agents form their expectations in a macroeconomic context. Expectations play a central role in determining the dynamics of macroeconomies. Whenever agents’ decisions are modeled to depend on future states of the economy and those decisions affect this economy, self-referentiality also comes into play in determining the economy’s outcomes. Self-referentiality means that the actual path of the economy is determined in part by the path expected by its composing agents. Thus, in order to get an understanding of the resulting economic dynamics, the process by which agents form their expectation must be specified.

Since the seminal contributions of Muth (1961) and the Lucas (1976) critique, the standard solution adopted in macroeconomic models has been the so-called Rational Expectations (RE) hypothesis, by which it is assumed that agents form their expectations consistently with the true law of motion of the economy. Despite its apparent success in solving the self-referentiality problem, the RE approach has been severely criticized for its high demands on agents computational capabilities. One particular weakness of the RE approach relates to the absence of a procedural specification of how agents expectations may actually get to an equilibrium. A solution to this issue has been proposed through the incorporation of Simon’s (1982) types of bounded rationality features.

The adaptive learning approach reflects one such attempt to go beyond the RE hypothesis and try to explain how agents expectations are formed in a real-time process. In this approach agents are assumed to act like econometricians, with expectations formed from forecast functions that are adjusted as new data
becomes available. Once a learning algorithm is explicitly specified to update the forecasting rule under this real-time evolution of the economy, conditions for the convergence of the implied learning process to a given RE equilibrium can be derived. Works considered seminal on this literature are Bray (1982), and Marcet and Sargent (1989), between others (see Evans and Honkapohja, 2001).

In spite of the explicit introduction of bounded rationality, initially, the main purpose of the adaptive learning modeling of expectations has been to provide an asymptotic justification to the RE hypothesis. More recently, this approach has received increased attention for its appeal as a more realistic behavioral assumption for applied macroeconomics studies, a feature highlighted for instance in Sargent (1999). The main distinctive feature of works in the literature that followed has been the replacement of the RE unrealistic assumptions, implying an instantaneous adjustment of agents expectations, with a characterization of agents as adaptive learners in forecasting their own environment. This way, adaptive learning allowed establishing a new channel through which the effects of expectation shocks can dissipate dynamically through the the economy.

In this thesis we assess the practices common to this applied literature on the use of recursive algorithms to depict the evolution of agents’ beliefs over time. Our main interest is in providing a tentative assessment of adaptive learning as representative of how agents form their expectations, with a focus on the computational implementations involved in this approach. To approach the issues related to these implementations we first establish an estimation and evaluation framework that mimics the real-time process of expectation formation through learning-to-forecast exercises, which is presented in chapter 1.

From outset, it is important to state the main qualifications to our approach. First, by adopting an unrestricted non-structural modeling approach one side
of the self-referentiality introduced by expectations into the determination of macroeconomic outcomes is neglected in our approach. Namely, we model agents expectations taking observations of macroeconomic variables as given, instead of being determined jointly with our simulated forecasts. A second qualification, as with any work of empirical nature, relates to the representativeness of our sample of data. Here we use real-time quarterly data on US inflation and output growth covering a broad post-WWII period of time.

In spite of these important restrictions, we argue that our results are still representative of the approaches adopted in empirical studies of learning, where forecasts for our focused variables are usually taken as those relevant for agents economic decisions. These forecasts are also commonly introduced into modeling as new exogenous explanatory variables, hence in accordance to our approach. Our analysis then develops around three main issues we find to have been previously neglected in the literature.

In chapter 2 we investigate the methods used for the initialization of the learning algorithms, which are obvious requirements for the computational implementation of these recursive estimation mechanisms. In chapter 3, our focus shifts to an analysis of the determination and calibration of the learning gains, which are parameters determining how much weight the learning algorithms attach to past observations in the estimation process. Finally, in chapter 4 we assess the choice of a representative learning mechanism, the specification of which is a requirement in any application of the adaptive learning approach.
Chapter 1

Estimation framework and the learning algorithms

1.1 Introduction

Adaptive learning algorithms have been proposed to provide the heuristics through which agents can be assumed to form their expectations, in order to pragmatically validate whether the consistency requirements inherent to rational expectations (RE) can be satisfied by boundedly rational agents. Going beyond the RE paradigm, however, comes at the cost of introducing another degree of freedom into the analysis, as one learning algorithm must be specified to represent the evolution of agents beliefs. Two algorithms have received most of the attention in the literature, namely, Least Squares (LS) and Stochastic Gradient (SG), and the different formulations from which these algorithms can be obtained as estimators is the subject of this chapter. Other than laying the ground for the analysis that follow in this thesis, this chapter also contributes to the literature by proposing an unified estimation framework for the analysis of these learning algorithms.

Although the LS and the SG algorithms share a similar recursive formulation, which makes them suitable for mimicking agents adjusting their forecasts as new data becomes available over time, their recursions can be clearly distinguished for the application, or not, respectively, of a “normalization” step during the updating process. It is also for the absence of this specific mechanism that the SG is characterized by a lower computational complexity as compared to the LS, hence leading some authors to advocate for its use as a

1Some results in this chapter have been published in Berardi and Galimberti (2013).
more plausible learning device from a bounded rationality standpoint (Barucci and Landi, 1997; Evans and Honkapohja, 1998b).

One way to deepen our understanding of these learning rules has been through the establishment of correspondences between them and the Kalman filter (as done in Ljung and Soderstrom, 1983; Ljung and Gunnarsson, 1990; Sargent, 1999; Sargent and Williams, 2005; Branch and Evans, 2006; Evans et al., 2010), for which optimality properties are known from a long standing literature (see Anderson and Moore, 1979). Previous studies, however, have focused mainly on the analysis of the LS algorithm (Sargent, 1999; Branch and Evans, 2006), while correspondence results for the SG case have been found to hold only approximately in a long-run sense, where any transient phase affecting the algorithm’s estimates has already died out (Sargent and Williams, 2005; Evans et al., 2010). Furthermore, these correspondences have been separately drawn for the specific cases of decreasing and constant sequences of gains, which are parameters determining how quickly a given information is incorporated into the algorithm’s coefficients estimates.

The first purpose of this chapter is to extend the exact correspondence results for the LS algorithm both to the case of the SG algorithm, as well as to the more general case of an unrestricted time-varying sequence of learning gains. We do that by providing a renewed interpretation of how these correspondences can be drawn with respect to the non-recursive forms of the LS and the SG algorithms, presented in section 1.2, where the non-recursive form for the latter algorithm is also an original contribution of this chapter. We then present the exact correspondences between the learning algorithms and the Kalman filter in section 1.3, also discussing how these exact correspondences relate to the previous approximate correspondences. As we adopt an exact approach in drawing our correspondences, we argue that our results favor both the computational implementation of these algorithms, as well as their employment over out-of-equilibrium paths.
The second purpose of this chapter is to propose an evaluation framework that mimics the real-time process of expectation formation through what we define as learning-to-forecast exercises. This framework, presented in section 1.4, will be used throughout this thesis, together with US macroeconomic data, to empirically investigate the relative merits of different approaches to the use of adaptive learning algorithms in applied macroeconomics. To illustrate its use, we present in section 1.5 a preliminary assessment on the performance of the forecasts associated to the learning algorithms relative to forecasts obtained from surveys. Here the survey forecasts are taken as proxies to agents’ expectations, thus providing a benchmark to judge the plausibility of the learning algorithms as representative of these agents’ learning-to-forecast behavior.

The results we obtain with this exercise indicate that the learning algorithms are found to be outperformed by the surveys in most of the cases considered, particularly at the shorter forecasting horizons. Rather than concluding with a pessimistic view over the learning approach, we discuss the very likely possibility of informational issues affecting our results. Besides, we argue that the fixed gain calibration and the restriction of agents learning to a single algorithm can represent important constraints to the algorithms performance. This motivates our evaluation of the empirical support for these assumptions in the chapters that follow. Therefore, an ultimate conclusion on the plausibility of learning is left open until we have dealt with these issues.

We end this chapter with some concluding remarks in section 1.6.

1.2 Digression into algorithms origins

1.2.1 Preliminaries

Consider an estimation context faced by a real-time agent wishing to obtain inferences about the law of motion of a variable of interest, say $y_t$. From an
economic perspective, these inferences can be thought of as the middle step agents undertake in a process of learning-to-forecast in order to form their expectations.

To narrow down our focus, we assume this agent attempts to construct such inferences assuming that $y_t$ is statistically related to other observed variables, say a vector of (pre-determined) variables $x_t = (x_{1,t}, \ldots, x_{K,t})'$, through a linear regression of the form

$$y_t = x_t'\theta_t + \varepsilon_t,$$  \hspace{1cm} (1.1)  

where $\theta_t = (\theta_{1,t}, \ldots, \theta_{K,t})'$ stands for a vector of (possibly time-varying) coefficients, and $\varepsilon_t$ denotes a (Gaussian$^3$) white noise disturbance with variance given by $\sigma^2_t$. Both coefficients and disturbances are assumed not to be directly observable by the agent. Under this context, a technique for estimation of $\theta_t$ is required to allow the agent to construct inferences for $y_t$ on the basis of (1.1).

In the literature of learning and expectations in macroeconomics (see Evans and Honkapohja, 2001) recursive algorithms have been proposed for this task. Two of the main forms adopted are the LS and the SG specifications.

Algorithm 1 (LS). Under the estimation context of (1.1), the LS algorithm assumes

\footnote{This specification and the results we obtain with it can be straightforwardly extended to a multivariate regressions context, an autoregressive context, or yet in both dimensions to a vector autoregression (VAR) specification.}

\footnote{Distributional assumptions such as Gaussianity are not strictly necessary for our purposes, but are required to guarantee the optimality of the Kalman filter estimator associated to this non-stationary context. This latter is the basis under which a unifying smoother is derived later for the initialization of the learning algorithms.}
the form of

\[ \hat{\theta}_{t}^{LS} = \hat{\theta}_{t-1}^{LS} + \gamma_{t} R_{t}^{-1} x_{t} \left( y_{t} - x_{t}' \hat{\theta}_{t-1}^{LS} \right), \]  
\[ R_{t} = R_{t-1} + \gamma_{t} \left( x_{t} x_{t}' - R_{t-1} \right), \]

where \( \gamma_{t} \) is a learning gain parameter, and \( R_{t} \) stands for an estimate of regressors matrix of second moments, \( E [x_{t}x_{t}'] \).

Algorithm 2 (SG). Under the estimation context of (1.1), the SG algorithm is given by

\[ \hat{\theta}_{t}^{SG} = \hat{\theta}_{t-1}^{SG} + \mu_{t} x_{t} \left( y_{t} - x_{t}' \hat{\theta}_{t-1}^{SG} \right), \]

with \( \mu_{t} \) standing for the learning gain parameter.

Notice the hats in \( \hat{\theta}_{t}^{LS} \) and \( \hat{\theta}_{t}^{SG} \) indicate that they stand for estimates of \( \theta_{t} \) in (1.1), and these estimates are based on period \( t \) information.

Since the seminal works of (Bray, 1982; Marcet and Sargent, 1989) the LS algorithm has been taken as the natural choice to represent agents mechanism of adaptive learning. This choice is in general attributed to the widespread knowledge of its non-recursive counterpart, the so-called Ordinary Least Squares (OLS) estimator, between econometricians. The SG algorithm, on the other hand, provides a computationally simpler alternative, a feature clearly apparent in (1.4) for the absence of the LS “normalization” step given by the inverse of the matrix of second moments. For this reason some authors have advocated for its use as a more plausible learning device from a bounded rationality standpoint (Barucci and Landi, 1997; Evans and Honkapohja, 1998b).

---

4This form is closer to that used in the adaptive learning literature under the name of Recursive Least Squares, for the case where the gain is decreasing with time, or Constant-Gain (Recursive) Least Squares, for the case of a time-invariant gain. In the engineering literature other variations in the nomenclature can be found and a computationally less demanding form is more common where the inversion of \( R_{t} \) in (1.3) is avoided by the use of the matrix inversion lemma (see Haykin, 2001).

5This form is common to both the adaptive learning (see also Evans et al., 2010) and the engineering literature. In the latter this algorithm is generally known as least mean squares, although commonly referring to the constant gain case, while stochastic gradient is often referred to the case of a time-decreasing gain. In some contexts, however, stochastic gradient is referred as a whole family of filters (see Macchi, 1995, p. 52).
Both the LS and the SG algorithms require the specification of a sequence of learning gains. The learning gain stands for a parameter determining how quickly a given information is incorporated into the algorithm’s coefficients estimates. Three of the main alternatives for the specification of this learning gain are those of a time-decreasing, a time-constant, and a time-varying (not restricted to be decreasing) sequence of values, and their suitability depends on the time-varying nature of the environment.

A decreasing-gain LS was the seminal choice in the learning literature, so as to match the recursive form of the OLS estimator. For the case of linear models with time-invariant parameters, this estimator is known to possess some well desired properties, such as consistency and efficiency, though these properties do not extend to a time-varying context. This latter fact implies the intriguing observation that a decreasing-gain LS learning mechanism is appropriate only along the time-invariant path of a RE equilibrium, where learning itself is indeed pointless (Bray and Savin, 1986).

Extensive evidence (see Stock and Watson, 2003; Cogley and Sargent, 2005; Sims and Zha, 2006; Sargent et al., 2006) favoring time-varying parameter models of the economy has, nevertheless, challenged this paradigm, and the departure from the parameter constancy assumption (see Margaritis, 1990; Bullard, 1992; McGough, 2003) has naturally led to the requirement of adjustments to the learning rules as well. Our focus in this thesis will be initially on the constant gain specification, which has been in the spotlight of most applied research since (Sargent, 1999). Such a choice naturally sprouts from the tracking capabilities associated to the constant gain specification and its suitability for time-varying environments. Later in chapters 3 and 4 we also assess the case of time-varying gains, determined according to an outer adaptive mechanism. Hence, to keep up with generality our presentation will leave the gain specification unrestricted, unless stated otherwise.

The main difference between the LS and the SG origins resides in how the
estimation problem was first formulated, either into a non-recursive (block) minimization problem or into a recursive (filtering) form, respectively. In spite of this distinction, each algorithm can be interpreted under both formulations.

1.2.2 Non-recursive forms

The LS is originally derived from a non-recursive estimation problem (see Ljung and Soderstrom, 1983, pp. 57-61), namely the minimization of the sum of weighted error squares as given by

$$\hat{\theta}_{LS}^{t} = \arg \min_{\theta} \sum_{i=1}^{t} \beta(t,i) \left( y_i - x_i' \hat{\theta}_{LS}^t \right)^2, \quad (1.5)$$

where

$$\beta(t,i) = \begin{cases} \alpha_i \prod_{k=i+1}^{t} \lambda_k & \text{for } i < t, \\ \alpha_i & \text{for } i = t, \end{cases} \quad (1.6)$$

indicates how past observations are discounted, and thus, it is typically increasing in $i$ for a given $t$. The structure in the weighting scheme imposed by (1.6), though not required, provides an exponential forgetting profile in the criterion (1.5) when $\lambda_k \leq 1$. In this sense, $\lambda_k$ stands for the stream of forgetting factors and $\alpha_i$ regulates the ceiling to the weights these factors attach to observations. The solution to (1.5), which is quadratic in $\hat{\theta}_{LS}^t$, results in the non-recursive LS estimator,

$$\hat{\theta}_{LS}^t = \left[ \sum_{i=1}^{t} \beta(t,i) x_i x_i' \right]^{-1} \sum_{i=1}^{t} \beta(t,i) x_i y_i. \quad (1.7)$$

Letting

$$\lambda_k = \frac{\gamma_k - 1}{\gamma_k} (1 - \gamma_k) \text{ and } \alpha_i = 1, \quad (1.8)$$
we obtain

\[
\beta(t, i) = \begin{cases} 
\frac{\gamma_i}{\prod_{k=i+1}^{t} (1 - \gamma_k)} & \text{for } i < t, \\
1 & \text{for } i = t,
\end{cases}
\] (1.9)

which indicates how the gains sequence in (1.2)-(1.3) gets translated into the weights put into past observations. It is particularly interesting to note that:

(i) when \(\gamma_i = 1/i\), \(\beta(t, i) = 1\) and (1.7) reduces to the OLS estimator; and,

(ii) with a constant gain, \(\gamma_i = \bar{\gamma}\), past observations receive geometrically decaying weights, i.e., \(\beta(t, i) = (1 - \bar{\gamma})^{t-i}\). Furthermore, it requires just a few derivations to show that the recursive form in (1.2)-(1.3) corresponds to the non-recursive solution in (1.7) with the weights given by (1.9)\(^6\).

The SG algorithm can also be put into a similar non-recursive form. Using the notation of (1.6), the non-recursive SG is given by\(^7\)

\[
\hat{\theta}_{t}^{SG} = \sum_{i=1}^{t} \beta(t, i) x_i y_i, 
\] (1.10)

\[
\lambda_k = (I - \mu_k x_k x_k') \text{ and } \alpha_i = \mu_i, 
\] (1.11)

where we can see that compared to the LS non-recursive form, in (1.7), the SG does not have the “normalization” term given by the inverse of the regressors (sample) matrix of moments. On the other hand, the way the SG discounts past observations is not fully determined by the choice of the gains sequence as it happens for the LS, but it also depends on the data. This is clearly reflected into the definition of the forgetting factor in (1.11), which under a multivariate context turns itself, and the weighting factor in (1.10), into a matrix form. We take this latter difference as an explanation for the finding that the SG estimates are sensitive to data scales (Evans et al., 2010), a point that will prove very important for the calibration of this algorithm in our applications.

\(^6\)See appendix A.1.
\(^7\)See appendix A.2.
1.2.3 Filtering origins

The SG, in turn, is recursive from its origins (see Widrow, 1971): it stands as a stochastic approximation to the iterative method of Steepest Descent (SD) used to achieve the optimal solution to a linear filtering problem. Within the context of model (1.1) the filtering problem is to find a vector of coefficients \( \hat{\theta}_t \) such that the variance of the squared estimation error associated to these estimates is minimal, i.e.,

\[
\hat{\theta}_t = \arg \min \frac{1}{2} E \left[ y_t - x_t' \hat{\theta}_t \right]^2,
\]

(1.12)

the gradient of which leads to the first order condition

\[-E \left[ x_t \left( y_t - x_t' \hat{\theta}_t \right) \right] = 0,\]

(1.13)

which is famously known as the orthogonality condition, given that it states that optimality of \( \hat{\theta}_t \) requires that the estimation errors must be orthogonal to each regressor variable. Solving (1.13) for the coefficients vector we get to the optimal solution to the linear filtering problem,

\[
E \left[ x_t x_t' \right] \hat{\theta}_t = E \left[ x_t y_t \right],
\]

(1.14)

\[
\hat{\theta}_t = E \left[ x_t x_t' \right]^{-1} E \left[ x_t y_t \right],
\]

(1.15)

also known as the Wiener-Hopf equation. Notice that from a deterministic viewpoint, the use of averaged sample counterparts of the expectational operators in (1.15) would lead to a least squares solution resembling to (1.7).

From the stochastic viewpoint, if the covariance matrix of the regressors \( E \left[ x_t x_t' \right] \) and the cross-covariances between the regressors and the endogenous variable \( E \left[ x_t y_t \right] \) are known, the optimal solution can be readily computed from (1.15). Such a task, however, may become computationally cumbersome as the number of regressors increases. Furthermore, under the time-
varying context of (1.1), these (cross-)covariances would be time-varying as well, and thus a new computation of (1.15) would be required for each new observation. A simpler alternative is to use a numerical optimization method in order to iteratively navigate along the error-performance surface, which is given by the objective function in the minimization of (1.12), until the optimal vector of coefficients is found. One such a method is the SD, which is developed to apply successive corrections to the coefficients estimates in the direction opposite to the gradient vector, (1.13), i.e.,

$$\hat{\theta}_{i}^{SD} = \hat{\theta}_{i-1}^{SD} + \kappa_{i} \left( E \left[ x_{t} y_{t} \right] - E \left[ x_{t} x_{t}^{\prime} \right] \hat{\theta}_{i-1}^{SD} \right),$$  

(1.16)

where it is important to note that the coefficients estimates are not (necessarily) indexed by time, in the sense that the recursion can be applied more than once within the same set of information, and the parameter $\kappa_{i}$ controls the size of the correction from one iteration to the next.

The similarity between the recursions adopted by the method of SD and the SG are not just a coincidence. As a matter of fact, the SG algorithm of (1.4) comes from a stochastic approximation rationale to the SD for the case where the relevant (cross-)covariances are not known. The idea is simply to replace the (theoretical) gradient in (1.16) by an estimate computed from the latest squared estimation error, i.e., the gradient of $\frac{1}{2} \left( y_{t} - x_{t}^{\prime} \hat{\theta}_{i-1} \right)^{2}$ with respect to $\hat{\theta}_{i-1}$. Doing that, one readily obtains the SG recursion of (1.2) from the stochastic approximation based on the SD of (1.16), noticing that the subscript indexes of the latter are converted to period ($t$) subscripts, and the step size parameter is converted to the learning gain parameter in the SG formulation.

A stochastic interpretation can also be given to the LS algorithm under the same filtering context that gives rise to the SG algorithm (Ljung and Soderstrom, 1983, pp. 46-8). Such a rationale is obtained by employing a method of iterative solution to (1.15) more sophisticated than the SD, namely the Newton’s method. Although the same recursive structure of (1.16) is maintained,
the difference is that under Newton’s method the gradient is computed up to a
second order expansion by multiplying the gradient derivative by the inverse
of its associated Hessian matrix. Under (1.12) the Newton’s method would be
translated as

\[ \hat{\theta}^{NW}_i = \hat{\theta}^{NW}_{i-1} + \eta_i \left( E \left[ x^t_i y_i \right] - E \left[ x^t_i \hat{\theta}^{NW}_{i-1} \right] \right) . \] (1.17)

Again, a recursive estimate for the Hessian matrix \( E \left[ x^t_i x'_i \right] \) can be constructed
by noting that under the target optimal solution, the Hessian matrix is the
solution \( R \) to \( E \left[ x^t_i x'_i - R \right] = 0 \). Applying the above iterative method to solve
for this condition one obtains

\[ R_i = R_{i-1} + \eta_i \left( E \left[ x^t_i x'_i \right] - R_{i-1} \right) , \] (1.18)

which after substitution of \( E \left[ x^t_i x'_i \right] \) by its observed counterpart, \( x^t_i x'_i \), and ad-
justing the subscripts and the gain, leads to the same recursion for the estimate
of the matrix of second moments in the LS algorithm, (1.3). Substituting this
estimate for the Hessian matrix in (1.17), and proceeding with the same ap-
proximations for the first order gradient as we did for the SD, one also finds
that the stochastic version to the Newton’s method in (1.17) resembles exactly
the coefficients recursion of the LS in (1.2).

1.3 State-space unifying framework

1.3.1 State-space representation

Having established how the LS and the SG algorithms compare in terms of
their original formulations we now show how both these algorithms can be
obtained as special cases of a more general estimator. This is the Kalman filter
when applied to the estimation of the time-varying parameters of the linear
relationship assumed in (1.1). For the purpose of an appropriate identification
of the Kalman filter we further assume these parameters follow a random walk model as
\[ \theta_t = \theta_{t-1} + \omega_t, \] (1.19)
where \( \omega_t \) is assumed to be (Gaussian) white noise with variances (and covariances) given by \( \Omega_t = E[\omega_t\omega_t'] \). The random sequences \( \epsilon_t \) and \( \omega_t \) are also assumed to be mutually independent.

Equations (1.1) and (1.19) are recognizably in a state-space form for a regression with time-varying coefficients, where the former is treated as the observation equation and the latter as the state equation (see Hamilton, 1994, pp. 372-408). The main advantage of such a state-space form is that it serves as a framework for the derivation of the Kalman filter used to obtain recursive estimates of the states based on the observed signals.

### 1.3.2 Estimation algorithms and their correspondences

**Kalman filter**

Adapted to our context, the Kalman filtering recursion is given by\(^9\)
\[
\begin{align*}
\hat{\theta}_t &= \hat{\theta}_{t-1} + K_t \left( y_t - x_t' \hat{\theta}_{t-1} \right), \\
K_t &= \frac{P_{t-1}x_t}{x_t'P_{t-1}x_t + \sigma_t^2}, \quad (1.21) \\
P_t &= \left( I - \frac{P_{t-1}x_t'}{x_t'P_{t-1}x_t + \sigma_t^2} \right) P_{t-1} + \Omega_t, \quad (1.22)
\end{align*}
\]
where \( K_t \) is known as the Kalman gain vector and \( P_t \) stands for the covariance matrix of the coefficients estimates, i.e., \( P_t = E \left[ (\theta_t - \hat{\theta}_t) (\theta_t - \hat{\theta}_t)' \right] \).

Other than for its optimality properties, the Kalman filter also turns out to be useful for providing an unifying framework for the adaptive learning algorithms we are interested in. This is done by imposing restrictions on the

\(^8\)See footnote 3.  
\(^9\)See appendix A.3.
dynamics of the second moments of the disturbances affecting the motion of the assumed state-space model, i.e., $\sigma_t^2$ and $\Omega_t$ (see Ljung and Soderstrom, 1983; Ljung and Gunnarsson, 1990).

Our contribution here is to show that the specifications of $\sigma_t^2$ and $\Omega_t$ that establish the LS and the SG algorithms as special cases of the Kalman filter can be connected to the definitions of the terms $\lambda_t$ and $\alpha_t$ that we used in (1.8) to draw the correspondence between the recursive and the non-recursive forms of these algorithms. Namely, the general basis of the correspondences we are drawing here starts by assigning

$$\sigma_t^2 = \lambda_t \alpha_t^{-1}, \quad (1.23)$$

and then proceeding with a derivation of $\Omega_t$ that would turn the Kalman filter recursions, (1.20)-(1.22), into the specific cases of the LS and of the SG algorithms, (1.2)-(1.3) and (1.4), respectively.

**Least Squares**

The LS algorithm can be obtained as the special case of the Kalman filter by setting

$$\sigma_t^2 = \frac{\gamma_{t-1}}{\gamma_t} (1 - \gamma_t), \quad (1.24)$$

$$\Omega_t = \left( \frac{\gamma_t}{\gamma_{t-1} (1 - \gamma_t)} - 1 \right) \left( I - \frac{x_{t-1}^r x_t}{x_t^r x_{t-1}^r + \sigma_t^2} \right) p_{t-1}. \quad (1.25)$$

Substituting these quantities into (1.21) and (1.22), and adding a superscript to distinguish the resulting algorithm from the general Kalman filter, it is straightforward to find

$$K_t^{LS} = \frac{p_{t-1}^{LS} x_t}{x_t^r p_{t-1}^{LS} x_t + \frac{\gamma_{t-1}}{\gamma_t} (1 - \gamma_t)}, \quad (1.26)$$

$$p_t^{LS} = \frac{\gamma_t}{\gamma_{t-1} (1 - \gamma_t)} \left( I - K_t^{LS} x_t^r \right) p_{t-1}^{LS}, \quad (1.27)$$
which has the same form as the LS algorithm in (1.2)-(1.3) with the inversion of
the matrix of second moments replaced by $\gamma_i^{-1}P_i^{LS}$ using the matrix inversion
lemma\(^{10}\).

The above correspondence generalizes those of Ljung and Gunnarsson (1990, p. 10) and Sargent (1999, pp. 115-8) to the case of a time-varying gain.

**Stochastic Gradient**

The SG algorithm can be found as the special case of the Kalman filter when
we set

\[
\sigma_i^2 = \mu_i^{-1} - x_i'x_i, \quad (1.28)
\]

\[
\Omega_t = I - \left(I - \frac{P_{t-1}x_i'x_i}{x_i'P_{t-1}x_i + \sigma_i^2}\right)P_{t-1}. \quad (1.29)
\]

Substituting these into (1.21) and (1.22) we find that\(^{11}\)

\[
K_t^{SG} = \mu_t x_t, \quad (1.30)
\]

\[
P_t^{SG} = I. \quad (1.31)
\]

Although this correspondence has been mentioned in Ljung and Gunnarsson (1990, p. 11), to the author’s knowledge its derivation from specific expressions for $\sigma_i^2$ and $\Omega_t$ has never been made explicit in a reasonable sense into the previous literature. An alternative derivation was given by Karjalainen (1996, p. 34), which obtained this correspondence for the constant-gain SG. Although Karjalainen’s derivation can be extended to the time-varying gain case, it suffers with two important drawbacks: (i) it requires a specific initialization of $P_0$; and (ii) the computation of $P_t^{SG}$, though not required for the computation of the SG coefficient estimates, is dependent on $t + 1$ regressors information, which is at odds with the main idea of recursive estimation.

\(^{10}\)See appendix A.4.

\(^{11}\)See appendix A.5.
1.3.3 Discussion

Our approach follows that of Ljung and Soderstrom (1983); Ljung and Gunnarsson (1990); Sargent (1999) where specific parametrizations of $\sigma_t^2$ and $\Omega_t$ are hand-picked in order to make the algorithms match exactly the more general Kalman estimator applied to a state-space unifying framework that has been extensively explored in the empirical macroeconometrics literature (see Stock and Watson, 1996, and references therein). Theoretically, the Kalman filter learning formulation has been analyzed by Bullard (1992) and McGough (2003), in the context of the self-referential ‘cobweb’ model, which also indicates the potential of these correspondences.

Following Benveniste et al. (1990), similar correspondences to the ones we obtain here have been drawn by Sargent and Williams (2005) and Evans et al. (2010). However, instead of holding exactly, their correspondences hold only in an approximated sense, when the algorithms transient phases have already died out. From an applied standpoint, the main drawback of the approximate approach is that the accuracy of this approximation depends on how closer the initialization of the algorithm is to its steady state estimates. From the learning and expectations standpoint, the fact that the approximation holds only asymptotically makes it hard to use the resulting framework for a unifying analysis of both learning convergence and out-of-equilibrium dynamics.

An understanding of the interplay between these features is taken as the main issue of Sargent and Williams (2005), which uses the approximate framework to formalize the idea of agents learning priors about drifting coefficients. The asymptotic complementarity between convergence and escapes allowed these authors to isolate the influence of the priors over the occurrence of those distinct dynamical features, though these priors were taken as pre-determined. By generalizing the correspondences of the LS and SG learning algorithms with the Kalman filter, and further allowing for unrestricted time-varying gains, our results can be taken as providing a framework of analysis for a case under
which agents are allowed to adapt their priors in accordance to their experience.

Our choice for exact correspondences, therefore, favors both the empirical applicability of the adaptive learning algorithms as well as their interpretation as learning devices operating, not necessarily but often, off the long run steady state path of inferences.

1.3.4 Further implementation details

Projection facility

It is often of interest to restrict the coefficients for the law of motion in (1.1) to evolve within a bounded region in the parameter space. The requirement of stability of $y_t$ is one of such cases, which under the VAR specifications we explore (details are given below) can be verified by using standard tools of multivariate time series analysis (see Lütkepohl, 2005, pp. 13-18). In the adaptive learning literature, this issue has been dealt with by resorting to the use of a device known as projection facility (Evans and Honkapohja, 1998a), which is a mechanism assumed to be coupled to the learning algorithms such that whenever the estimates leave a bounded region in the parameters space the device is activated in order to contain the escape$^{12}$.

There are different ways a projection facility can be conceived for implementation (see, e.g., Ljung and Soderstrom, 1983, pp. 366-368). One of the simplest is to set the new coefficient estimates equal to their previous value whenever the new estimates obtained from the algorithm lead to a violation of the stability region. Here we even offer some additional room for the algorithms to work around its escapes by defining a broader stability region$^{13}$.

The main challenge, however, is to provide a plausible rationale for this mech-

$^{12}$A similar mechanism has been used in Cogley and Sargent (2005), though these authors refer to it as a reflecting barrier to the coefficients evolution.

$^{13}$Namely, the projection facility is activated only if the (1.1)’s companion matrix’s eigenvalue with highest modulus gets bigger than 2.
anism within the learning context. In what follows we show that our formulation provides a compelling interpretation.

Under our unified framework, the projection facility is equivalent to setting the gain vector $K_t$ to zero. Given (1.21), there is only one plausible circumstance that could legitimize this value: the denominator, given by $x^t_P_{t-1}x_t + \sigma^2_t$, has to go to infinity, which results from $\sigma^2_t \rightarrow \infty$. Recall that $\sigma^2_t$ stands for the variance of the random disturbance affecting the motion of the endogenous variable. Its value tending to infinity could therefore be associated to the observation of a large shock, or an additive outlier. Hence, the projection facility can be taken as representative of the behavior of a learning agent when faced with an observation that clashes with a “prior” of stability, leading to a treatment of such observation as an outlier.

In spite of our use of a projection facility, this mechanism is only intended to control occasional data-led escapes of the estimates, and it should not be regarded as a main player in the algorithms’ performance. The stability requirement should, notwithstanding, be guaranteed by a proper calibration of the algorithms.

**Upper bounds on learning gains**

Beyond the properties of the data, it is the calibration of the learning gain parameters in the adaptive algorithms that mainly determines whether estimates will become unstable or not. Given the relevance of these parameters for the convergence and tracking performance of the algorithms, the determination of upper bounds\textsuperscript{14} to form their range of admissible values has been a topic of great interest in the literature (see Haykin, 2001, and references therein). Stability analysis reveals that the upper bound on the SG learning gain is dependent on the data, a result clearly connected to the finding that the SG algorithm

\textsuperscript{14}The lower bound is given by zero for both LS and SG algorithms, where there will be no adaptation. Negative gain values are not reasonable as it would imply updating the coefficient estimates onto a direction opposite to that indicated by the observed error.
is not scale invariant (see Evans et al., 2010). This is not the case for the LS algorithm though, and for this reason we treat the LS case first.

Loosely speaking, the LS learning gain is usually allowed to assume any value between zero and one, i.e., $0 \leq \gamma_t < 1$. To understand these bounds recall (from eq. 1.9) that the weights given to past observations by the LS, under a constant gain, are determined by $\beta(t,i) = (1 - \tilde{\gamma})^{t-i}$. Fixing $t$, it is straightforward to notice that any reasonable value for $\tilde{\gamma}$ should lie within the above range: (i) if $\tilde{\gamma} < 0$, the weights increase as far as we go to the past, turning the estimates insensitive to new observations; (ii) if $\tilde{\gamma} = 1$, all the weights on past observations become null, and the estimation will reflect only current observations; (iii) if $\tilde{\gamma} > 1$, the weights signs oscillate alternatively, and if furthermore $\tilde{\gamma} > 2$ their magnitude increase as far as we go to the past. Notwithstanding, when implementing the LS algorithm on data, it is often the case that the unitary upper bound is not tight enough to ensure stability. Therefore, the upper bound for the LS gain, denoted by $\tilde{\gamma}_{\text{max}}$, needs to be experimentally adjusted for the application of interest. This is done by departing from the unity value as a necessary, though not sufficient, initial upper bound.

The gain calibration gets much more involved for the case of the SG algorithm, due to the dependency of this algorithm’s stability on the data. Haykin (2001) presents the derivation of a tight upper bound on the SG gain relying on an analysis of higher order modes of convergence\textsuperscript{15}. This upper bound is given by $\beta_{\text{max}} = 2/K S_{\text{max}}$, where $K$ stands for the number of coefficients being estimated, and $S_{\text{max}}$ stands for the maximum value of the power spectral density of the regressors\textsuperscript{16}. Details of this approach go beyond our scope, so we just take this expression as a reference. We then proceed in the same way as we proposed for the LS calibration by adjusting it experimentally. The goal is to compute the SG algorithm over a range of gain values as wide as possible,

\textsuperscript{15}It is important to note that this analysis does not account for self-referentiality.

\textsuperscript{16}For applied purposes, this statistic requires estimation based on the regressors data. A summary of alternative methods for power spectrum estimation is presented in Haykin (2001, pp. 78-82). Here we use a simple periodogram applied to the whole sample of data.
but yet not incurring in a too frequent activation of the projection facility.

1.4 Design of learning-to-forecast exercises

In this section we outline the methodological approach we will use throughout this thesis to empirically compare the forecasts associated to the different learning algorithms and their calibrations. These learning-to-forecast exercises are defined departing from two evaluation criteria. We then discuss the model specification and the details on the real-time dataset we use within a three-stage routine designed to compute and evaluate the algorithms’ forecasts.

1.4.1 Evaluation criteria

We propose two comparative exercises in order to achieve our purpose of assessing the learning algorithms. An evaluation criterion is associated to each of these exercises, which are designed departing from two distinct perspectives on the use of the algorithms: the first of an economic agent, and the second of the researcher.

From the point of view of an economic agent, who has to build forecasts of variables relevant for economic decisions, what matters is the accuracy of such forecasts. Here accuracy is naturally measured by the distance of the forecasts to the actual future realizations. The evaluation criterion associated to this first exercise is henceforth represented by the algorithms’ forecasting accuracy.

From the point of view of the researcher, in contrast, interest is in uncovering which mechanism better represents the learning-to-forecast behavior of the economic agents being modeled. Thus, what matters for the researcher is the resemblance of the forecasts produced by the algorithms to those forecasts observed from actual agents. To measure these latter we use data of survey

17 The definition of actuals is somewhat complicated under a real-time data setup, an issue we discuss below.
forecasts as a proxy\textsuperscript{18}. Thus, we succinctly denote the evaluation criterion associated to this exercise as resemblance to surveys.

Common to both exercises is the focus on evaluating the forecasts associated to the different learning algorithms and their calibrations, which is done using a loss function that measures their distance from a desired target.

**Definition 1** (Loss function). The loss function $L(f(x, \theta), z)$ maps a set of outcomes, $z$, and a set of forecasts, $f$, to the real number line, where forecasts may depend on a set of parameters, $\theta$, and observed variables, $x$, and the loss function may accept multiple inputs, such as forecasts at different horizons\textsuperscript{19}.

The characterization of a loss function is typically made within the context of a decision problem. A relative ordering of different forecasts is obtained by comparing their expected losses, $E[L|I]$, where $I$ stands for the conditioning information set under which the forecasts were computed and the losses evaluated\textsuperscript{20}. In our exercises, such conditioning involves three main aspects: the specification of a model, the initialization of the algorithms, and the selection sample within which the gain calibrations are determined. The variations we explore in terms of model specifications will be detailed below, whereas the issues associated to the initialization and the calibration of the learning algorithms will be analyzed in chapters 2 and 3, respectively.

The main difference between our two exercises then rests on the outcomes, $z$, being considered for the evaluation of the algorithms. Letting $y$ stand for the set of values being forecasted, $f_{i,j}$ for the set of forecasts associated to algorithm $i$ under calibration $j$, and $s$ for the set of forecasts obtained from surveys, we define our two evaluation criteria as follows.

**Criterion 1** (Forecasting Accuracy). An algorithm $a'$ with gain calibration $c'$ is said to be a more plausible representation of agents’ OPTIMAL behavior in terms

\textsuperscript{18}Another option would be to resort to laboratory experiments as in Hommes (2011).

\textsuperscript{19}We are abstracting here from a possible dependency of losses on state variables, e.g., bigger losses during downturns of the business cycles (see Elliott and Timmermann, 2008).

\textsuperscript{20}For our purposes, we abstract from the use of $I$ in the notation that follows.
of learning-to-forecast than an alternative algorithm \(a''\) with gain calibration \(c''\) if
\[
E \left[ L (f_{a',c'}, y) \right] < E \left[ L (f_{a'',c''}, y) \right].
\]

**Criterion 2** (Resemblance to Surveys). An algorithm \(a'\) with gain calibration \(c'\) is said to be a more plausible representation of agents’ ACTUAL behavior in terms of learning-to-forecast than an alternative algorithm \(a''\) with gain calibration \(c''\) if
\[
E \left[ L (f_{a',c'}, s) \right] < E \left[ L (f_{a'',c''}, s) \right].
\]

### 1.4.2 Model and data

We will keep two things fixed across our exercises: the model specification agents are assumed to use for estimation and forecasting; and the data context, where we restrict our attention to the time series of inflation and growth, denoted by \(\pi_t\) and \(g_t\), respectively.

**Model**

Our focus is on unrestricted VAR specifications with time-varying coefficients, which applied to inflation and growth follow

\[
\begin{bmatrix}
\pi_t \\
g_t
\end{bmatrix} = \begin{bmatrix}
\varphi_{\pi,0,t} & \varphi_{\pi,1,t} & \cdots & \varphi_{\pi,p,t} & \varphi_{g,1,t} & \cdots & \varphi_{g,p,t}
\end{bmatrix} \begin{bmatrix}
1 \\
\pi_{t-1} \\
\vdots \\
\pi_{t-p} \\
g_{t-1} \\
\vdots \\
g_{t-p}
\end{bmatrix} + \begin{bmatrix}
\epsilon_{1,t} \\
\epsilon_{2,t}
\end{bmatrix},
\]

(1.32)

where \(\varphi_{\pi,0,t}\) and \(\varphi_{g,0,t}\) are the intercept coefficients associated to each equation, \(\varphi_{\pi,i,t}\) and \(\varphi_{g,i,t}\) \((i = 1, \ldots, p)\) are the coefficients associated to the lagged values of \(\pi_t\) and \(g_t\) in the inflation equation, \(\phi_{\pi,i,t}\) and \(\phi_{g,i,t}\) play a similar role in the growth equation, and \(p\) denotes the VAR lag order. For robustness we estimate
VARs with lag orders from 1 to 4, though we do not allow these lag orders to change in real-time within a given exercise.

Extending the notation of (1.1) to the bivariate case, the VAR(p) specification can be abbreviated as

\[ y_t = x_t' \theta_t + \epsilon_t, \]

with \( y_t = (\pi_t, g_t) \), \( x_t = (1, \pi_{t-1}, \ldots, \pi_{t-p}, g_{t-1}, \ldots, g_{t-p})' \), \( \theta_t = \left( (\phi_{\pi,0,t}, \phi_{\pi,1,t})', (\phi_{\pi,p,t}, \phi_{\pi,p+1,t})', (\phi_{g,1,t}, \phi_{g,2,t})', \ldots, (\phi_{g,p,t}, \phi_{g,p+1,t})' \right)' \), and \( \epsilon_t = (\epsilon_{1,t}, \epsilon_{2,t}) \). This notation has the advantage of facilitating the specification of how the forecasts are computed on the basis of estimates of the vector of coefficients, denoted by \( \hat{\theta}_t \). Accordingly, multi-horizons forecasts for the endogenous variables can be computed as

\[
\begin{align*}
\hat{y}_{t+1|t} &= x_{t+1|t}' \hat{\theta}_t, \\
\hat{y}_{t+2|t} &= x_{t+2|t}' \hat{\theta}_t, \quad (1.33)
\end{align*}
\]

where the subscripts on the variables indicate that forecasts are computed on the basis of period \( t \) information. This is known as the indirect method of computation of multi-horizons forecasts (Marcellino et al., 2006). Particularly, notice that as the forecasting horizon increases the vector of regressors is updated as \( x_{t+1|t} = (1, \pi_t, \ldots, \pi_{t-p+1}, g_t, \ldots, g_{t-p+1})' \), \( x_{t+2|t} = (1, \hat{\pi}_{t+1|t}, \ldots, \pi_{t-p+2}, \hat{g}_{t+1|t}, \ldots, g_{t-p+2})' \), and so on, hence including only information available at period \( t \).

Data

In order to define the setting in which forecasts can be interpreted as proxy for agents’ macroeconomic expectations, it is important to take into account the informational restrictions under which expectations are supposed to be formed. To deal with this issue, we adopt the environment provided by a real-time dataset. A real-time dataset provides snapshots (or vintages) of its variables for given dates in the past as it was available for an observer at that time (see Croushore, 2011). The informational gains provided by a real-time
dataset clearly depends on the degree within which the data is revised through time. As stressed by Stark and Croushore (2002), the relevance of these revisions has fundamental implications for the evaluation of forecasts genuinely made in real-time.

We use quarterly data on the US real GNP/GDP and its price index from 1947q2 to 2011q4, which sums up to 259 observations for each variable. Our data on these series come from the Philadelphia’s Fed Real-Time Data Research Center\(^\text{21}\) and consists of vintages from 1966q1 to 2012q1, i.e., a total of 185 snapshots of what was known on these variables by a market participant in real-time. As our interest is on forecasts for output growth and inflation, we obtain these rates from the above data on levels computing their associated annual growth rates by compounding their simple quarterly growth factors.

For the purpose of comparing the algorithms’ forecasts to those provided by survey respondents, we use data from the Survey of Professional Forecasters (SPF)\(^\text{22}\), which are made available by the Philadelphia’s Fed as well. Each quarter, this survey asks professional economists to give their forecasts for several macroeconomic variables, including those we indicated above, and also over different forecasting horizons. Here we use the median of the individual forecasts for output growth and inflation made for a total of five horizons, namely from $t$ (nowcast) to $t+4$. The SPF data is available from 1968q4 onwards, and, consistent to our data on actuals, the last survey data we use is that of 2010q4, which contains forecasts up to 2011q4. An overview of these data series is presented in Figure 1.1.

Finally, two important issues arise with the use of a real-time dataset to evaluate the criteria proposed above. First, to evaluate the forecasting accuracy criterion a measure of actuals need to be specified, and it is not clear whether agents forecasting in real-time target the first measure they (will) observe, or

\(^{21}\)See \url{http://www.philadelphiafed.org/research-and-data/real-time-center/}. We have done some specific adjustments to the original dataset, as detailed in Appendix B.1.

\(^{22}\)See \url{http://www.philadelphiafed.org/research-and-data/real-time-center/survey-of-professional-forecasters/}.
The series of actual observations refer to those we have available at the latest vintage in the real-time dataset we are using, i.e. 2012q1. The SPF’s forecasts refer to those made for the pointed period from the information set of the corresponding forecasting horizon. The shaded area indicates the observations left aside for the initialization procedure of the algorithms. See the text for further details on sources and construction.
any of the later revised measures. In spite of some inescapable arbitrariness in this issue, here we use the first-available observations arguing that this is the target that closest reflects the short term perspective of the learning process we attempt to mimic.

The second issue is related to the evaluation of the resemblance to surveys criterion: when comparing the forecasts we construct on the basis of the real-time observations to the survey forecasts, the relative timing of information of the latter should be respected. Fortunately, the real-time dataset we use is already constructed so as to be consistent with the SPF timing of information. Namely, the real-time data is observed with one quarter of delay to reflect what one would have known at the middle of each quarter, which is when the survey forecasters submit their forecasts\(^23\). But in spite of respecting the timing of information release on the variable being forecast, there may still be violations to the evaluation’s informational assumptions regarding the variables used in the construction of the forecasts. Clearly, the survey respondents have access to a much larger set of information than our data/model specifications are providing to the learning algorithms. We return to the relevance of this issue in the analysis of the next section.

### 1.4.3 Three-stage routine

Our approach to empirically evaluate the criteria defined above develops into three stages: initialization, estimation and forecasting, and evaluation. The first two are carried out identically for both of our two exercises. The different perspectives assumed for each of our evaluation criteria will unfold into different evaluative statistics in the last stage.

\(^{23}\)The Bureau of Economic Analysis (BEA) releases the first measure (Advance Report) of the US quarterly GDP by the end of the first month subsequent to the quarter, while its first revision is released by the end of the second month (see also Stark, 2010, Fig. 1, for an illustration of this timeline).
Initialization stage

The first step in the process of obtaining forecasts from the LS and SG algorithms is to set their initializations, which requires estimates for \( \hat{\theta}_0 \). Here we will follow a smoothing approach which we detail in the next chapter. The general idea of this approach is: first, to set aside an initial portion of the available sample of data to serve as a training sample; and second, to repeat a smoothing routine within the training sample until the smoothed initial estimates converge to values consistent with the algorithm’s steady state operation. The simulation evidence we report in the next chapter indicates that this method provides the most consistent estimates of initials within our estimation framework.

For the purpose of a training sample, we use the first 75 observations of our sample. Namely, the data from 1947q2 to 1965q4 as represented by the shaded area in Figure 1.1. Devoting this amount of observations to the initialization stage goes in line with most of the empirical works, cited in the next chapter, that undertake calibrations of adaptive learning rules of the kind we are dealing with here. Also notice that under our real-time data environment the observations lying within the training period may, and occasionally do, undergo through revisions as well\(^{24}\). To match the initial estimates with the data used in the next stage of estimation/forecasting, therefore, we repeat our initialization procedure for each vintage of data.

Estimation and forecasting stage

With the initials given, we proceed to apply the LS and the SG algorithms to estimate VAR model specifications with the series of data on inflation and growth. Estimation and forecasting are carried out by vintage as follows:

(i) The recursions for each algorithm/gain are computed departing from

\(^{24}\)These are in general due to the BEA annual and benchmark revisions to the national accounts.
the vintage/algorithm/gain initials until exhaustion of the vintage sample.

(ii) The \( t, \ldots, t + 4 \) forecasts for each vintage/algorithm/gain are computed using the last estimates of the model specification, where \( t \) stands for the vintage quarter.

We repeat these computations for each vintage of data from 1966q1 to 2010q4, which results in a total of 180 forecasts for each algorithm/gain/horizon both for inflation and growth.

**Evaluation stage**

At this stage we collect all the forecasts we have obtained from the previous stage and compare their associated average losses according to the two different criteria defined above. Recall that for *forecasting accuracy* we compute average losses on the basis of forecast errors, whilst the loss associated to the *resemblance to surveys* criterion is based on the forecast comparison errors, defined as the difference between each algorithm’s forecasts and the forecasts obtained from surveys. In both cases, a parametric form for the loss function still needs to be specified. Our main focus will be on the squared error loss function, defined as \( L_2(f, z) = (z - f) (z - f)' \) under the notation of section 1.4.1.

To further substantiate our comparative analysis, we also make use of tests common to the literature on forecast evaluation. Namely, we adopt both the Diebold and Mariano (1995) (DM) test for equal (unconditional) predictive ability, and its more recently developed conditional counterpart test of Giacomini and White (2006) (GW). Other than for robustness purposes, our choice for these two tests can also be well justified: while the first stands as a classical test, whose properties have been long studied in the literature, the second clearly represents a more appropriate test for our purposes of comparison of different estimation methods. As its own proponents qualify, under the GW
test “...the finite sample properties of the estimators on which the forecasts may depend are preserved asymptotically” (Giacomini and White, 2006, p. 1545), a feature of essential relevance for our empirical applications.25

In spite of these motivations, each of these tests comes with particular restrictions with respect to its suitability to the context of our application. First, it is well known that the DM test is not the most appropriate test for the comparison of forecasts based on models with estimated parameters (see West, 1996), as it is the case here. Key to this issue is the DM hypothesis that the loss differentials are covariance stationary, which is likely to be violated due to the effects that uncertainty about the model’s parameters estimates introduce in finite samples. A reaction to this issue, according to Diebold’s (2013) summary of the twenty years of literature that followed the seminal proposal of the DM test, has been given by accumulated evidence showing that the test’s standard normal distribution often provides a fairly plausible approximation to the nonstandard distributions induced by estimation uncertainty.

With respect to the GW test, the main qualification on its use in our context relates to its unsuitability for the comparison of forecasts constructed on the basis of models estimated recursively. An intuition for this requisite can be drawn recalling that the finite sample nuisance introduced by estimation uncertainty in the comparison of model-based forecasts, as discussed above, tends to vanish asymptotically if the estimator is consistent as is usually the case for recursive schemes. Given that the test proposed by Giacomini and White (2006) aims at comparing the finite sample properties of forecasts, their approach is restricted to forecasts based on models estimated using rolling observation windows that are small relative to the number of out-of-sample observations.26 Nevertheless, our focus on constant gain specifications of the learning algorithms attenuates the violation of this “non-recursive restriction”;26

---

25We summarize the calculations involved in each of these tests in Appendix B.2.
26There is limited Monte Carlo evidence on how small the rolling window needs to be relative to the number of out-of-sample observations for the asymptotics of the GW test to apply (see Clark and McCracken, 2013).
recall that, from section 1.2.2, under the constant gain specifications the learning algorithms attach geometrically decaying weights to past observations. Hence, leaving the decreasing gain setup aside, we are essentially working with an approximation to the rolling windows estimation scheme required by the GW test.

1.5 Quality of learning forecasts relative to surveys

In this section we present a first illustration on the use of the evaluation framework outlined above. We adopt the learning-to-forecast exercise on forecasting accuracy to (preliminarily) address a key question related to the empirical relevance of adaptive learning. The plausibility on the use of the learning algorithms as representative of how agents form their expectations clearly requires evidence that these mechanism forecast at least as good (or bad) as the surveys. Therefore, we assess how the forecasts associated to the learning algorithms perform relative to those forecasts obtained from surveys.

As a preliminary assessment, we follow the literature in adopting a common constant gain calibration in the estimation of both equations of our VAR specifications on inflation and growth. Specifically, we set the LS gain to $\bar{\gamma} = 0.0495$, which is close to the usual calibrations found in the literature. Branch and Evans (2006), for example, find an optimal common gain of 0.047 if the squared losses on these variables are weighted equally. We then obtain an equivalent calibration for the SG as $\bar{\mu} = 3.84 \times 10^{-4}$.

1.5.1 Results

We first look at some descriptive and evaluation statistics for each individual series of forecasts and actuals data, as presented in Table 1.1. The statistics for the actuals are presented on both concepts of observation, showing the rel-

\[ \text{Following our discussion in section 1.3.4, this SG gain is obtained as } \bar{\mu} = \beta_{\max} \left( \bar{\gamma} / \bar{\gamma}_{\max} \right). \]
evance of the use of a real-time setting for the analysis of learning. Clearly, the revisions tend to promote predictability for inflation, but to harm it for growth: (i) whereas the variance of inflation decreases with revisions, it increases substantially for growth; (ii) the first order autocorrelation of inflation increases with revisions, but decreases for growth. Also notice that for the case of growth the average distance of the revised series to that observed in real-time is higher than the survey’s mean squared forecast errors (MSFE), hence corroborating our assumption that agents target the first available measures.

Overall, most of the statistical properties of the series of forecasts tend to be in line with each other and with their targets. Perhaps the most remarkable discrepancy is on the variance of the growth forecasts, which is found smaller than that of the actuals. This result reflects the greater difficulty in forecasting growth rates compared to inflation: the former variable is known to have a lower degree of dynamic persistence and a higher degree of volatility than the latter; these properties make most of the variability observed in growth rates look like randomness, which is a feature hard to capture by deterministic forecasts (see also Patton and Timmermann, 2011). Nevertheless, the survey forecasts were yet the best to overcome this deviation, presenting MSFE more than a half smaller than those of the algorithms forecasts on growth. On inflation the forecasts presented smaller deviations, though the surveys and the LS forecasts were found positively biased as evidenced by their minimums.

Among the algorithms forecasts, the evaluation statistics also tend to point in unison to a favorite model specification for each variable: the VAR(3) for inflation, and the VAR(1) for growth, though their supremacy over the other specifications is not pronounced. Another observation distinguishing the two algorithms relates to their mean levels and variability; on both variables: (i) the LS algorithm tends to overpredict, whilst the SG is commonly found underpredicting the actuals on average; (ii) the LS forecasts presented higher variances than those obtained with the SG. Therefore, relative to the LS, the
Table 1.1: Data and forecasts statistics.

(a) Inflation.

<table>
<thead>
<tr>
<th>Series</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
<th>Var</th>
<th>AR(1)</th>
<th>CorA</th>
<th>CorS</th>
<th>MSFE</th>
<th>MSFCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revised</td>
<td>2.75</td>
<td>-0.44</td>
<td>10.93</td>
<td>2.39</td>
<td>0.78</td>
<td>0.85</td>
<td>0.85</td>
<td>0.75</td>
<td>0.78</td>
</tr>
<tr>
<td>Real-time</td>
<td>2.67</td>
<td>-0.33</td>
<td>9.39</td>
<td>2.68</td>
<td>0.63</td>
<td>1.00</td>
<td>0.81</td>
<td>0.00</td>
<td>1.03</td>
</tr>
<tr>
<td>Surveys</td>
<td>2.93</td>
<td>0.62</td>
<td>9.49</td>
<td>2.46</td>
<td>0.94</td>
<td>0.81</td>
<td>1.00</td>
<td>1.03</td>
<td>0.00</td>
</tr>
<tr>
<td>Least Squares</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>2.98</td>
<td>-0.37</td>
<td>10.38</td>
<td>2.91</td>
<td>0.80</td>
<td>0.72</td>
<td>0.90</td>
<td>1.65</td>
<td>0.55</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>2.91</td>
<td>0.27</td>
<td>10.17</td>
<td>2.93</td>
<td>0.88</td>
<td>0.74</td>
<td>0.91</td>
<td>1.53</td>
<td>0.48</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>2.87</td>
<td>0.14</td>
<td>9.70</td>
<td>2.91</td>
<td>0.90</td>
<td>0.75</td>
<td>0.92</td>
<td>1.41</td>
<td>0.47</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>2.83</td>
<td>-0.25</td>
<td>9.85</td>
<td>3.08</td>
<td>0.89</td>
<td>0.74</td>
<td>0.90</td>
<td>1.51</td>
<td>0.56</td>
</tr>
<tr>
<td>Stochastic Gradient</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>2.62</td>
<td>-0.61</td>
<td>9.02</td>
<td>1.74</td>
<td>0.60</td>
<td>0.60</td>
<td>0.81</td>
<td>1.82</td>
<td>0.92</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>2.72</td>
<td>-0.35</td>
<td>7.90</td>
<td>1.56</td>
<td>0.86</td>
<td>0.66</td>
<td>0.83</td>
<td>1.55</td>
<td>0.79</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>2.77</td>
<td>-0.83</td>
<td>7.46</td>
<td>1.46</td>
<td>0.92</td>
<td>0.70</td>
<td>0.84</td>
<td>1.38</td>
<td>0.76</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>2.81</td>
<td>-0.50</td>
<td>6.51</td>
<td>1.37</td>
<td>0.93</td>
<td>0.70</td>
<td>0.82</td>
<td>1.38</td>
<td>0.82</td>
</tr>
</tbody>
</table>

(b) Growth.

<table>
<thead>
<tr>
<th>Series</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
<th>Var</th>
<th>AR(1)</th>
<th>CorA</th>
<th>CorS</th>
<th>MSFE</th>
<th>MSFCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Revised</td>
<td>2.78</td>
<td>-8.89</td>
<td>9.30</td>
<td>9.21</td>
<td>0.45</td>
<td>0.82</td>
<td>0.71</td>
<td>3.09</td>
<td>4.95</td>
</tr>
<tr>
<td>Real-time</td>
<td>2.54</td>
<td>-6.14</td>
<td>8.67</td>
<td>6.17</td>
<td>0.50</td>
<td>1.00</td>
<td>0.78</td>
<td>0.00</td>
<td>2.56</td>
</tr>
<tr>
<td>Surveys</td>
<td>2.24</td>
<td>-5.19</td>
<td>7.01</td>
<td>3.14</td>
<td>0.72</td>
<td>0.78</td>
<td>1.00</td>
<td>2.56</td>
<td>0.00</td>
</tr>
<tr>
<td>Least Squares</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>3.04</td>
<td>-3.66</td>
<td>6.58</td>
<td>1.58</td>
<td>0.60</td>
<td>0.39</td>
<td>0.58</td>
<td>5.55</td>
<td>2.74</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>3.03</td>
<td>-5.75</td>
<td>7.03</td>
<td>2.14</td>
<td>0.55</td>
<td>0.30</td>
<td>0.51</td>
<td>6.34</td>
<td>3.26</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>3.04</td>
<td>-8.88</td>
<td>6.80</td>
<td>2.91</td>
<td>0.40</td>
<td>0.27</td>
<td>0.46</td>
<td>6.96</td>
<td>3.91</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>3.08</td>
<td>-7.94</td>
<td>8.06</td>
<td>3.08</td>
<td>0.43</td>
<td>0.28</td>
<td>0.46</td>
<td>7.04</td>
<td>4.02</td>
</tr>
<tr>
<td>Stochastic Gradient</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>1.90</td>
<td>-2.09</td>
<td>4.60</td>
<td>1.33</td>
<td>0.48</td>
<td>0.43</td>
<td>0.59</td>
<td>5.43</td>
<td>2.14</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>1.98</td>
<td>-3.01</td>
<td>4.91</td>
<td>1.39</td>
<td>0.71</td>
<td>0.38</td>
<td>0.58</td>
<td>5.58</td>
<td>2.16</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>1.93</td>
<td>-2.45</td>
<td>4.62</td>
<td>1.15</td>
<td>0.77</td>
<td>0.35</td>
<td>0.55</td>
<td>5.78</td>
<td>2.27</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>1.90</td>
<td>-2.04</td>
<td>4.33</td>
<td>0.96</td>
<td>0.80</td>
<td>0.32</td>
<td>0.52</td>
<td>5.94</td>
<td>2.40</td>
</tr>
</tbody>
</table>

The forecasts statistics refer to those obtained for the first forecasting horizon, $h = 0$, and over the full evaluation sample from 1981q1 to 2010q4. Other than the usual descriptive statistics, AR(1) stands for the first order autocorrelation of each series, CorA and CorS stands for the correlation with the series of actual (real-time) and survey forecasts, respectively, and MSF(C)E stands for the mean squared forecast (comparison) errors.
SG algorithm seems to provide more conservative forecasts.

We now shift our focus to the comparative of the algorithms forecasts to those obtained from surveys. We present the results of the statistical tests for equal predictive ability of these forecasts in Tables 1.1a and 1.1b, for inflation and growth, respectively. In contrast to our results above, here we attempt a more comprehensive view on this issue by considering the forecasts at longer horizons too.

Clearly, we can see that the surveys’ superiority is again evidenced in these comparatives. But most importantly, this advantage tends to decrease as the horizon increases, and in some cases even revert to favor the algorithm’s forecasts. This was the case of inflation forecasts from $h = 3$, where the negative values in the DM statistics indicate the SG outperformed the surveys, and one case of growth forecasts favoring the LS at $h = 3$.

The results on the statistical significance of these differences in predictive ability also confirm that the survey’s lead is concentrated in the shorter forecasting horizons. This is particularly the case when considering the DM test. It is also noticeable that statistical significance was more scarce using the GW test. From the tests definitions, this result indicates that the different (unconditional) predictive abilities observed in this evaluation sample may not be a good indicator of the relative quality of these forecast when conditioning such comparisons to their estimation information²⁸.

1.5.2 Discussion

Our results indicate that the forecasts associated to the learning algorithms, following formulations and calibrations common to the literature, are overall outperformed by the survey forecasts. There is also some non-negligible evidence of statistically significant differences of predictive ability between these

²⁸We add to this point recalling that the DM test is known to be over-sized, i.e., falsely rejecting the null hypothesis more frequently than its significance level (Harvey et al., 1997).
Table 1.2: Learning versus survey’s accuracy forecasting inflation.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Forecasting horizons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h=0</td>
</tr>
<tr>
<td>Least Squares</td>
<td></td>
</tr>
<tr>
<td>- Diebold and Mariano (1995):</td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>3.18</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>2.29</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>2.01</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>1.77</td>
</tr>
<tr>
<td>- Giacomini and White (2006):</td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>1.00</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>0.65</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>0.10</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>0.44</td>
</tr>
<tr>
<td>Stochastic Gradient</td>
<td></td>
</tr>
<tr>
<td>- Diebold and Mariano (1995):</td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>2.53</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>2.39</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>2.31</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>1.98</td>
</tr>
<tr>
<td>- Giacomini and White (2006):</td>
<td></td>
</tr>
<tr>
<td>VAR(1)</td>
<td>1.08</td>
</tr>
<tr>
<td>VAR(2)</td>
<td>0.16</td>
</tr>
<tr>
<td>VAR(3)</td>
<td>0.54</td>
</tr>
<tr>
<td>VAR(4)</td>
<td>0.35</td>
</tr>
</tbody>
</table>

The statistics refer to tests of equal predictive ability between the algorithms forecasts (specified in the first column) and the survey forecasts, under a squared loss function. The DM and GW statistics correspond to (B.2) and (B.3), respectively, and their associated p-values. The signal of the DM statistic indicates which series of forecasts presented a smaller MSFE: positive (negative) values indicate the survey (algorithm) outperformed the algorithm (survey). Comparisons where the test indicates statistical significance below 20% are highlighted in **bold**.
Table 1.3: Learning versus survey’s accuracy forecasting growth.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Forecasting horizons</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Test</td>
<td>h=0</td>
<td>h=1</td>
<td>h=2</td>
<td>h=3</td>
<td>h=4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Least Squares</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Diebold and Mariano (1995):</td>
<td>VAR(1)</td>
<td>2.52</td>
<td>0.01</td>
<td>2.21</td>
<td>0.03</td>
<td>0.95</td>
<td>0.34</td>
<td>-0.15</td>
</tr>
<tr>
<td></td>
<td>VAR(2)</td>
<td>2.56</td>
<td>0.01</td>
<td>2.15</td>
<td>0.03</td>
<td>1.23</td>
<td>0.22</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>VAR(3)</td>
<td>2.41</td>
<td>0.02</td>
<td>1.95</td>
<td>0.05</td>
<td>1.31</td>
<td>0.19</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>VAR(4)</td>
<td>2.57</td>
<td>0.01</td>
<td>2.11</td>
<td>0.04</td>
<td>1.39</td>
<td>0.17</td>
<td>0.91</td>
</tr>
<tr>
<td>- Giacomini and White (2006):</td>
<td>VAR(1)</td>
<td>4.81</td>
<td>0.03</td>
<td>2.05</td>
<td>0.15</td>
<td>1.32</td>
<td>0.25</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>VAR(2)</td>
<td>5.05</td>
<td>0.02</td>
<td>1.35</td>
<td>0.25</td>
<td>1.45</td>
<td>0.23</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>VAR(3)</td>
<td>3.69</td>
<td>0.05</td>
<td>1.56</td>
<td>0.21</td>
<td>1.46</td>
<td>0.23</td>
<td>2.01</td>
</tr>
<tr>
<td></td>
<td>VAR(4)</td>
<td>4.25</td>
<td>0.04</td>
<td>1.72</td>
<td>0.19</td>
<td>1.31</td>
<td>0.25</td>
<td>2.47</td>
</tr>
<tr>
<td>Stochastic Gradient</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Diebold and Mariano (1995):</td>
<td>VAR(1)</td>
<td>3.79</td>
<td>0.00</td>
<td>3.70</td>
<td>0.00</td>
<td>2.91</td>
<td>0.00</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>VAR(2)</td>
<td>3.11</td>
<td>0.00</td>
<td>3.14</td>
<td>0.00</td>
<td>3.08</td>
<td>0.00</td>
<td>1.85</td>
</tr>
<tr>
<td></td>
<td>VAR(3)</td>
<td>3.01</td>
<td>0.00</td>
<td>3.08</td>
<td>0.00</td>
<td>2.94</td>
<td>0.00</td>
<td>1.80</td>
</tr>
<tr>
<td></td>
<td>VAR(4)</td>
<td>2.98</td>
<td>0.00</td>
<td>3.07</td>
<td>0.00</td>
<td>2.92</td>
<td>0.00</td>
<td>1.79</td>
</tr>
<tr>
<td>- Giacomini and White (2006):</td>
<td>VAR(1)</td>
<td>0.28</td>
<td>0.60</td>
<td>0.32</td>
<td>0.57</td>
<td>0.07</td>
<td>0.79</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>VAR(2)</td>
<td>2.52</td>
<td>0.11</td>
<td>0.44</td>
<td>0.51</td>
<td>0.10</td>
<td>0.75</td>
<td>0.27</td>
</tr>
<tr>
<td></td>
<td>VAR(3)</td>
<td>4.07</td>
<td>0.04</td>
<td>1.02</td>
<td>0.31</td>
<td>0.12</td>
<td>0.72</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>VAR(4)</td>
<td>4.65</td>
<td>0.03</td>
<td>1.46</td>
<td>0.23</td>
<td>0.06</td>
<td>0.80</td>
<td>0.05</td>
</tr>
</tbody>
</table>

See footnotes to Table 1.2.
forecasts. Taken at its face value these results would therefore suggest that *these* algorithms *specifications*, and their respective *calibrations*, do not provide plausible representatives of how agents form their expectations. However, there are two important points we believe must be taken into account to qualify any such conclusion drawn from these results.

The first qualification relates to the very likely possibility of violations to our evaluation’s informational assumptions. Namely, in spite of our concerns on the use a real-time dataset, our learning algorithms are in disadvantage relative to the set of prediction-relevant information available to the survey’s respondents. By restricting our analysis to bivariate VAR specifications on inflation and growth we are clearly neglecting other variables with likely predictive content such as interest rates, indexes of consumer prices, industrial production and housing starts, as well as non-farm payrolls. Most importantly, most of these variables are available at a higher frequency than our quarterly dataset; as remarked by Stark (2010, p.3), “it is very reasonable to assume that the survey’s participants know the values of interest rates and labor-market variables for the first month of the first quarter of the quarterly forecast horizon.”

Therefore, our results of a poor performance of the algorithms’ forecasts relative to the surveys may not reflect an appropriate assessment of the plausibility of learning, but just a breach of the informational conditioning required in this issue. Moreover, our evidence that the survey’s lead weakens for longer horizons is also corroborative to this view. Under a stable environment, it is reasonable to expect that the survey’s informational advantage gets thinner as the forecasting horizon increases. I.e., the predictive content of the variables neglected by our model specifications will tend to die out in the longer horizons. Hence, an appropriate assessment would focus on how these forecasts compare in the longer horizons. There, our results indicated that the learning algorithms can even outperform the surveys, as was the case of the SG
algorithm for inflation\textsuperscript{29}.

Our emphasis above on these specifications and calibrations was intentional to point out our second qualification to the results in this section. Namely, the fixed gain calibration and the restriction that agents are not allowed to switch between different learning mechanisms are features we believe to represent important constraints to the algorithms’ performance. The fixed gain, though consistent with the idea of tracking a time-varying environment, does not necessarily represent an optimal solution to this estimation problem. In fact, as our correspondences to the Kalman filter revealed, the optimal learning gain for each algorithm depends on how the variance of the disturbances evolve through time. If these variances are changing, as evidenced for instance in Sims and Zha (2006) with US data, the fixed gain calibration of the learning algorithms would be misspecified.

Regarding the restriction to a single learning algorithm, our results have also evidenced important differences on the properties of the forecasts provided by the different specifications. Allegedly, we have seen that the SG algorithm tends to provide more conservative forecasts than the LS. Hence, it would be sensible to hypothesize that a combination of these algorithms may provide the best performance in forecasting the time-varying system. Given that both of these assumptions pervade the applied literature, we take the evidence of this section as a motivation for an empirical evaluation of their alternatives in the chapters that follow.

\textbf{1.6 Concluding remarks}

In this chapter we have studied the different formulations of the LS and the SG learning algorithms, as well as their corresponding origins. Whereas the LS has been initially proposed from a non-recursive estimation perspective,\textsuperscript{29}Another interpretation consistent to this evidence is that focusing in the longer horizons we are comparing the forecasts unconditional predictive ability.

\textsuperscript{29}Another interpretation consistent to this evidence is that focusing in the longer horizons we are comparing the forecasts unconditional predictive ability.

56
the SG is originally recursive. In spite of this distinction, we show that each algorithm can be interpreted under both formulations.

We have then established an estimation framework for the computational implementation of both of these learning algorithms. This framework unifies the learning algorithms through correspondences with the Kalman filter associated to a time-varying state space model. Two important contributions in our approach relate to the generality of our correspondences, covering the cases of unrestricted time-varying sequences of gains for both algorithms, and to their broader validity in an exact computational sense. Most of the previous correspondences in the literature have been found to hold only approximately in a long-run sense, where any transient phase affecting the algorithm’s estimates has already died out. Therefore, we argue that our exact approach in drawing these correspondences favors both the computational implementation of the learning algorithms as well as their employment over out-of-equilibrium paths.

We have also proposed in this chapter the evaluation framework that we will use throughout this thesis to empirically investigate different approaches to the use of the adaptive learning algorithms in applied macroeconomics. The idea behind this framework is an attempt to mimic the real-time process of expectation formation using learning-to-forecast exercises. These exercises put together three ingredients required by our estimation framework: the learning algorithms; a model specification, for which we use VARs with time-varying coefficients; and data, where we use a real-time dataset on US output growth and inflation.

Using this framework, we conducted a first assessment on the performance of the forecasts associated to the learning algorithms relative to the forecasts obtained from surveys. We argued that the plausibility on the use of the learning algorithms as representative of how agents form their expectations requires evidence of equal predictive ability between these mechanisms and the
surveys. However, our results in this exercise suggest a pessimistic view over the learning approach: the learning algorithms are found to be outperformed by the surveys in most of the cases considered, particularly at the shorter forecasting horizons.

Nevertheless, we conclude this chapter arguing that this would be a foregone conclusion on this key question related to the empirical relevance of adaptive learning, discussing some qualifications to these negative results. First, we show that it is very likely that informational issues may be affecting our results. Namely, the learning algorithms are in disadvantage relative to the set of prediction-relevant information available to the survey’s respondents. Second, we argue that the particular specifications of the learning mechanism adopted in this exercise can represent important constraints to the algorithms performance. This motivates the empirical analysis we undertake in the chapters that follow, leaving an ultimate conclusion on the plausibility of learning open until we have dealt with these issues.
Chapter 2

The initialization of the learning algorithms

2.1 Introduction

Reopening a long standing debate on how should expectations be modeled in macroeconomic models, we have argued that the heuristics provided by learning algorithms come at the cost of introducing new degrees of freedom into the analysis. One open node relates to how these recursive mechanisms should be initialized in order to be representative of agents’ learning-to-forecast behavior. In this chapter we investigate this issue with particular attention at the applied literature in macroeconomics.

Here applied is taken to encompass both theoretical simulations as well as exercises of empirical estimation and calibration. Examples can be found in Sargent (1999); Marcet and Nicolini (2003), or more recently in Eusepi and Preston (2011); Milani (2011), between many others cited throughout this chapter. The main distinctive feature of these works consists in the replacement of unrealistic assumptions implying an instantaneous adjustment of agents expectations, inherent in the rational expectations hypothesis, with a characterization of agents as adaptive learners of their own environment. More generally, our study will be relevant for scholars interested in the actual implementation of the learning algorithms here considered.

We start by reviewing the literature in order to pool together the initialization methods previously adopted into an archetypal classification. In spite of the obvious relevance of such issue, surprisingly, we did not find many other

---

1Most results in this chapter have been previously presented in Berardi and Galimberti (2012a).
attempts to systematically assess these methods. In economics, one exception
is provided by Carceles-Poveda and Giannitsarou (2007), although their con-
tribution is to a great extent restricted to theoretical applications. With the
freedom to develop our own assessment framework, we compare initialization
methods on the basis of two main criteria. First, we argue that a good
initialization should provide estimates coherent to the long run behavior of
the algorithm; second, achieving this coherence must be feasible within the
usual data availability restrictions in macroeconomics. Our review indicates
that none of the initialization methods found in the literature is able to satisfy
both these criteria.

Motivated by this critical finding, we propose a new method of initialization
based on smoothing within a sample of training data. Our point of de-
parture is the unified framework established in the previous chapter, under
which the main learning algorithms considered in the literature are obtained
as special cases of the Kalman filter associated to a time-varying parameters
model of the economy. Recalling, we have shown how to extend the asymp-
totic correspondences between the LS and the SG algorithms to hold exactly
in transient phases too, hence allowing for a unified approach to initializa-
tions. From these correspondences, long standing Kalman smoothing results
can be readily translated into smoothing routines for the estimates obtained
from each of the above learning algorithms, and we develop our routine using
these premises.

We then evaluate our procedure in comparison with two of the reviewed
alternative methods with respect to their convergence performance in a sim-
ulation exercise. We show that our approach is able to deal with the conflict
between COHERENCE and FEASIBILITY, present in the other methods, at the
same time that it has the advantage of being implementable in any algorithm
that can be encompassed into the Kalman unifying framework. This solution,
however, comes at the cost of an increased computational burden. To further
enhance our understanding on the relevance of these different initialization methods for applied macroeconomics, we also present an empirical exercise of learning-to-forecast. Using US inflation and growth data, results are again found to favor our new smoothing routine. An ultimate judgment on the relevance of these results, however, would require going beyond our simplified application, and we leave this issue open for future research.

The remainder of this chapter proceeds as follows. In section 2.2 we introduce the statistical approach we adopt to the analysis of the learning algorithms initializations. There we also present a discussion on what is required from an initial estimate for these algorithms, so as to provide the criteria through which we can critically evaluate the methods we review from the literature. This review is presented in section 2.3, where we also describe our own proposal of a new smoothing-based routine. We then proceed to present a simulation exercise, in section 2.4, and an empirical application, in section 2.5, both aiming at a comparative analysis between different methods of initialization. Finally, we conclude this chapter with some remarks in section 2.6.

2.2 Statistical analysis of learning algorithms

2.2.1 Learning and statistical rationales

Once a learning mechanism is specified, one of the main issues in the theoretical literature on adaptive learning has been to single out the conditions under which this learning process converges towards a target equilibrium (see Marcet and Sargent, 1988, for an earlier overview). This equilibrium is often well defined by the RE hypothesis and it is represented by a fixed-point in the dynamics of a self-referential structural model, where agents expectations play a role in determining the economy’s outcomes. The key feature of such analysis lies on the use of a stochastic approximation approach in order to assess the asymptotic dynamics of the stochastic system through a deterministic
differential equation (see Evans and Honkapohja, 2009, for a recent account).

Different questions have emerged from the applied side of this literature, where scholars have tried to understand how much of macroeconomic persistence can actually be attributed to learning (Orphanides and Williams, 2005b; Milani, 2007), and what part of business cycle fluctuations can be explained by a model with learning and expectational shocks (Eusepi and Preston, 2011; Milani, 2011). In contrast to the theoretical literature, where the interest is on stability and the eventual convergence of the learning algorithm to a given equilibrium, here the key feature is represented by the actual behavior of the learning algorithms in finite samples. Such shift in interest motivates a statistical analysis of these algorithms in what they stand as estimators.

Recursive estimation algorithms are statistically characterized by undergoing through two main distinct phases: a transient and a steady state one. In defining the criteria we use to assess initializations, further below, we suggest that the purpose of an initialization method, from an empirical perspective, is to provide estimates as close as possible to the algorithm’s steady state operation, since such beliefs should reflect the continuation of an estimation process that was already running prior to the sample beginning.

The separating frontier between the transient and the steady state phases, nevertheless, is not clear-cut. To obtain an assessment, it is common practice (see Haykin, 2001, p. 266) to focus on a statistical measure of interest and construct the algorithm’s learning curves, which represent how that measure evolves through time. Roughly, one can then visually lay up bare these phases by identifying the steady state when the statistic settles down. There are two main measures adopted for this purpose and they highlight different and complementary aspects of the tracking problem. Namely, the Mean-Square Deviation (MSD) between the actual vector of coefficients, $\theta_t$, and the algorithms estimates, $\hat{\theta}_t$, is intended to capture the (average) accuracy of the algorithm’s estimates.
Definition 1 (MSD). The MSD between the actual vector of coefficients in (1.1), $\theta_t$, and an algorithm’s estimates, $\hat{\theta}_t$, is given by

$$D_t = E \left[ \Delta_t^2 \right], \quad (2.1)$$

where $\Delta_t = \| \theta_t - \hat{\theta}_t \|$ stands for the Euclidean norm of the vector of coefficients deviations.

Its evolution through time is also associated with the speed with which the algorithm is able to adjust its estimates to the time-varying system.

The second measure is the Mean-Square Error (MSE) of estimation, which is intended to capture the (average) uncertainty surrounding the algorithm’s estimates.

Definition 2 (MSE). The MSE associated to an estimator $\hat{\theta}_t$ of the coefficients in (1.1) is given by

$$E_t = E \left[ |e_t|^2 \right], \quad (2.2)$$

with $e_t = y_t - x_t' \hat{\theta}_{t-1}$.

Although this latter measure also incorporates any biases present into the coefficient estimates, as captured by the MSD, its relationship to the definition of the true stochastic disturbance, $\varepsilon_t$ in (1.1), renders to the MSE an interpretation as a measure of estimation efficiency. In this sense, notice that $\sigma_t^2$ stands as a reference level to the MSE measure.

As noted by Benveniste et al. (1990, Part I, Chapters 1 and 4), tuning recursive algorithms through control of the gain parameter gives rise to a trade-off between the tracking speed and the accuracy/efficiency of estimates. We shall observe such a trade-off in the learning curves that we compute for various gain calibrations, but notice that these are steady state features, while our main interest lies on the transient behavior that follows the initializations of the algorithms\(^2\). In our context, hence, the MSD and MSE measures serve to

2Also notice that the MSD is a second moment measure, such that its convergence to differ-
the purpose of defining metrics that will be the basis of our main evaluation criterion of initializations.

**Definition 3 (MSD-MISALIGNMENT).** The MISALIGNMENT of an algorithm estimates at period \( t \), with respect to its MSD, can be measured by

\[
\mathcal{M}_t^D = \frac{|D_t - \overline{D}_t|}{\overline{D}_t},
\]

where \( \overline{D}_t = \lim_{t \to \infty} D_t \) stands for the steady state level of the algorithm’s MSD, and \( \overline{D}_t = \sqrt{E \left[ (\Delta_t^2 - D_t) \right]} \) stands for its standard deviation.

**Definition 4 (MSE-MISALIGNMENT).** The MISALIGNMENT of an algorithm estimates at period \( t \), with respect to its MSE, can be measured by

\[
\mathcal{M}_t^E = \frac{|E_t - \overline{E}_t|}{\overline{E}_t},
\]

where \( \overline{E}_t = \lim_{t \to \infty} E_t \) stands for the steady state level of the algorithm’s MSE\(^3\), and \( \overline{E}_t = \sqrt{E \left[ (|e_t|^2 - E_t) \right]} \) stands for its standard deviation.

Clearly, our measures of MISALIGNMENT have the appealing interpretation of representing the distance between the algorithm’s current MSD/MSE and its steady state level in terms of standard deviations. For simulation purposes, (2.1), (2.2), (2.3), and (2.4) can be readily evaluated by computing their (repeated) sample counterparts.

### 2.2.2 Requirements on initializations

From the recursive form of both learning algorithms, the initializations clearly take the form of estimates for \( \hat{\theta}_0 \) and \( R_0 \), although the latter is dispensed with in the SG case. To keep up with the generality of our analysis here we focus levels for different gains is not in conflict with the algorithm’s convergence in distribution to estimates around the true values of the coefficients.\(^3\) Although the the variance of the true stochastic disturbance, \( \sigma_t^2 \), provides a reference to an algorithm’s MSE, the steady state level of the latter need not equate to the former.
solely on the initialization of the coefficients estimates, $\hat{\theta}_0$, common to both algorithms\textsuperscript{4}.

Within the context of learning and expectations, the estimates provided by the learning algorithms are taken to represent agent’s beliefs about the economy. With this in mind, inquiring about the values to assign to $\hat{\theta}_0$ should lead to the question: (i) what were agent’s beliefs at the beginning of our sample of data? It is to answer to this question that an initialization method is purposefully designed. But from a statistical point of view, whatever the initial estimate we assume, it would be just a matter of time until the algorithm achieves convergence. Then the relevant question becomes: (ii) how long will it take for the algorithm to converge to an appropriate estimate of agent’s beliefs?

In the assessment that follows, we associate two evaluation criteria to the above questions in order to qualify the initialization methods previously adopted in the literature. The first is the \textit{coherence} of initial estimates relative to the steady state behavior associated with the gain used to calibrate the algorithm. As we have argued, however, at an applied level it becomes difficult to distinguish between transient and steady state dynamics. Our assessment on this criterion, therefore, is restricted to a relative scaling of how the different methods perform in terms of their \textit{misalignment}.

\textbf{Criterion 3 (\textit{Coherence}).} An initialization method is said to provide more coherent initial estimates than another method if the \textit{misalignment} of the former initial estimates, as measured by (2.3) or (2.4), is smaller than the \textit{misalignment} of the latter.

The second criterion is the \textit{feasibility} of the initialization method in the context of macroeconomic data. Even though in this case we could provide a definition in absolute terms, such as establishing a limit on the amount of data required by a feasible method, we opt for another relative form that is suitable for the context.

\textsuperscript{4}Moustakides (1997) provides a study on how to optimally initialize $R_0$ in the LS algorithm, proposing a simple rule based on the data signal-to-noise ratio. When it comes to our applied exercises we shall be precise about this rule and how we use it.
for our later comparative exercises.

**Criterion 4 (Feasibility).** An initialization method is said to be more feasible than another method if the amount of data required by the former to obtain the aimed initial estimates is smaller than the amount of data required by the latter to achieve the same purpose.

### 2.3 Initialization methods

A summary of how different initialization methods compare with each other is proposed in Table 2.1 and a discussion is provided in the following section. Some of the works referred in Table 2.1 applied more than one form of initialization for robustness purposes, and so figure in more than one of the classifications. Also notice that most of the works classified under the same classification are not, strictly speaking, identical with respect to their initialization method(s), but share the main characteristic defining the corresponding classification. In order to address the shortcomings we find in the methods tradition ally used in the literature, we then propose a new initialization routine based on smoothing within a training sample of data.

#### 2.3.1 A review of previous methods

From an applied perspective, initialization methods can be classified between two extreme ends depending on their suitability for simulation or empirical purposes. Their distinction in that respect reflects the amount of information available about the system to which the algorithms are applied. While in simulation studies the true system is known by the researcher, in empirical applications most knowledge incorporated into an initialization represents the assumptions that qualify the study in its own.

Starting from the theory-guided methods, one first way to initialize learning algorithms for simulation studies is that obtained from the use of full
Table 2.1: Comparative of initialization methods.

<table>
<thead>
<tr>
<th>Initialization methods</th>
<th>Benefit(s)</th>
<th>Drawback(s)</th>
<th>Previously applied in economics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exact</strong>: initials obtained from theoretical (known) REE.</td>
<td>- Most theoretically-grounded between these methods.</td>
<td>- Unsuitable for empirical applications, unless accurate previous estimates are available.</td>
<td>Sargent (1999); Marcet and Nicolini (2003); Bullard and Eusepi (2005)^1; Orphanides and Williams (2005b); Carceles-Poveda and Giannitsarou (2007)</td>
</tr>
<tr>
<td></td>
<td>- Low computational cost.</td>
<td>- May lead to incoherent estimates under non-stationary setups.</td>
<td></td>
</tr>
<tr>
<td><strong>Ad-hoc</strong>: initials hand-picked by researcher. Usually taking values closer to those from the Exact method.</td>
<td>- Useful for sensitivity analysis.</td>
<td>- Unsuitable for empirical applications.</td>
<td>Milani (2007); Carceles-Poveda and Giannitsarou (2007, 2008).</td>
</tr>
<tr>
<td></td>
<td>- Low computational cost.</td>
<td>- May be incoherent with the algorithm/gain of interest.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- High degree of subjective freedom.</td>
<td></td>
</tr>
<tr>
<td><strong>Diffuse-track</strong>: initials estimated from a training sample of actual/simulated data, using the algorithm/gain of interest.</td>
<td>- Provides initials coherent with algorithm/gain of interest.</td>
<td>- May require unfeasible amount of training obs. for convergence, specially for small gain values.</td>
<td>Milani (2007, 2008)^2; Huang et al. (2009); Pfajfar and Santoro (2010)^3; Eusepi and Preston (2011); Milani (2011).</td>
</tr>
<tr>
<td></td>
<td>- Easiness of implementation, as it follows directly from the algorithm's usual operation.</td>
<td>- Subjective determination of amount of training obs.</td>
<td></td>
</tr>
<tr>
<td><strong>Diffuse-ordinary</strong>: same as Diffuse-track, but using a decreasing gain algorithm, e.g., the ordinary least squares.</td>
<td>- Accelerated convergence.</td>
<td>- May converge to an estimate incoherent with the algorithm/gain.</td>
<td>Williams (2003); Orphanides and Williams (2005a); Carceles-Poveda and Giannitsarou (2007, 2008).</td>
</tr>
<tr>
<td></td>
<td>- Appropriate for a time-invariant context, e.g., a REE path.</td>
<td>- Subjective determination of amount of training obs.</td>
<td></td>
</tr>
<tr>
<td><strong>Smoothing</strong>: initials estimated from a training sample by smoothing the Diffuse-track estimates backwards repeatedly.</td>
<td>- Provides initials coherent with algorithm/gain of interest.</td>
<td>- Highest computational cost between these methods.</td>
<td>The present work.</td>
</tr>
<tr>
<td></td>
<td>- Feasible due to exchange of training obs. for computational cost.</td>
<td>- Trickier implementation.</td>
<td></td>
</tr>
</tbody>
</table>

(1) In Bullard and Eusepi (2005) the initialization method is not explicitly stated, but it is understood that the estimates are set to depart from the values implied by the model REE. (2) In Milani (2007, 2008) a Diffuse- (under our terminology) initialization is used, but there is no explicit reference to whether this follows the track or the ordinary method. The use of the former is later acknowledged by the same author in Milani (2011), and for this reason we classify all these papers under the Diffuse-track method. (3) In Pfajfar and Santoro (2010) the initialization is determined jointly with the calibration of the gain parameter using the whole sample of data instead of a pre-sample.
knowledge about the law of motions generating the data. This method is referred as \textit{Exact} in Table 2.1, and it first\textsuperscript{5} appeared into the seminal applied contribution of Sargent (1999). Its usage has since been prominent in studies that replace the assumption of frictionless RE equilibrium with the sticky process of expectations formation through adaptive learning. For lack of a better guess, this method proposes to take the coefficient values corresponding to the RE equilibrium as initial estimates for the algorithm’s recursion.

Clearly, the main benefits of the \textit{Exact} method of initialization relates to its theoretical support, as well as its exemption from pre-forecasting data requirements. These advantages, however, come at the cost of its unsuitability for empirical applications, where the information about the true system under analysis is often the object of study.

One closer alternative is provided by the method we denote as the \textit{Ad-hoc} initialization in Table 2.1, where the initials are hand-picked by the researcher. When taking the \textit{Exact} initials as reference, this method provides a way to validate the sensitivity of results obtained under the former. It lacks, however, the objectivity of the previous method given that there is usually no guidance on the magnitude of variations on the initials, and the researcher’s degrees of freedom increase rapidly with the system’s dimension. Both the \textit{Exact} and the \textit{Ad-hoc} methods may, furthermore, lead to initials still incoherent to agents’ beliefs in terms of the underlying algorithm and its calibration, given that both provide initials incorporating information from the system, but not from the learning algorithm itself.

Shifting now our focus to approaches less theoretically-grounded, but which in contrast are favored for their empirical suitability, there are two main methods adopted in the literature to initialize the estimates of recursive learning algorithms. In the engineering literature (see Ljung and Soderstrom, 1983, pp. 299-303, e.g.), it is often suggested that the coefficients should be initialized

\textsuperscript{5}Earlier simulation works, such as Bray and Savin (1986), also followed a similar approach, but in the context of a decreasing-gain LS algorithm.
with the value of zero (known as a diffuse prior) and an initial sample should be left aside to let the algorithm adjust its estimates according to the underlying calibration. This is especially recommended for the cases where there is not enough previous knowledge about the system under estimation so as to allow an educated guess. It is referred in Table 2.1 as the Diffuse-track method.

Clearly, under the Diffuse-track method of initialization the criterion of coherence is satisfied. As long as the algorithm and its calibration are appropriate for the application, it can be expected that convergence will eventually take place, and, therefore, estimates representing the steady state behavior of the algorithm can be obtained as initials. However, two problems arise with this method. First, it is up to the researcher’s wisdom to recognize how many observations are needed to get past the initial transient phase, an aspect that increases the method’s subjectivity. The second problem with the Diffuse-track method relates to its feasibility, which may become critical in the macroeconomic context where availability of data is usually restricted as compared to the engineering context under which this method was originally proposed. This last drawback is of special relevance for the case of learning gains calibrated to small values, as usual in economic applications, where the algorithms tend to show rather slow rates of convergence. We shall return to this point later in our simulation experiments, but notice that for simulation purposes, as in Huang et al. (2009); Eusepi and Preston (2011), this point is not restrictive as long as enough observations are generated for the algorithm to achieve convergence within the pre-forecasting initialization sample.

A second empirically-grounded method of initialization involves the use of the decreasing gain LS block estimation counterpart, namely the OLS estimator, within a pre-specified initial sample of data. Essentially, this method corresponds to an initialization of the coefficients from zero, and then updating the estimates within an initial sample using the LS estimator with decreasing
gains given by $\gamma_t = 1/t$. It therefore represents an hybrid of the Diffuse-track method, implemented with the LS algorithm under a decreasing gain. This method also seems to be the quite popular in empirical studies on learning in macroeconomics, perhaps due to the prominence of the LS algorithm and its popularity between applied researchers. In Table 2.1 we denote it by the Diffuse-ordinary method.

As with the previous method, the Diffuse-ordinary initialization suffers from the same lack of objectivity regarding the determination of how much of the available sample of data should be set aside for the initialization routine. Nevertheless, the fact that a relatively higher gain value is used in the first iterations of the initialization tends to improve the convergence speed considerably, and so favoring the feasibility of this method. The Diffuse-ordinary initialization can, therefore, be expected to require a lower number of initial observations to achieve convergence in relation to the Diffuse-track method.

The most important drawback of the Diffuse-ordinary initialization method, however, is its lack of coherence with respect to the algorithm’s gain calibration. Different gain values engender different steady state behaviors of the algorithm’s estimates. So, if the initialization for a given gain calibration is obtained by using a different gain value, this initial estimate will tend to be biased in relation to the algorithm’s steady state estimates. By using a decreasing gain the Diffuse-ordinary method provides the same initial estimate irrespective of the gain calibration for which this is required. Thus, even though it tends to attain a quicker convergence, the estimates to which this initialization method converges may be incoherent with the subsequent performance of the algorithm, a fact that ends up requiring further adaptations of the algorithm outside of the initialization sample in order to get to its steady state.

---

6 To prevent instabilities into the first estimates the decreasing gains are usually set as $\gamma_t = \bar{\gamma}/t$.

7 Also notice that this method can be adjusted to use the SG algorithm with a decreasing gain.
2.3.2 A new method based on smoothing

The main difficulty that the initialization methods reviewed above face for their use in empirical applications relates to the trade-off between their feasibility and the coherence they can achieve. These criteria can be seen to represent antagonistic requirements due to the effects that the size of the training sample of data has over them. Namely, while devoting additional data to the initialization procedure tends to favor coherence, by expanding the room for the algorithm’s convergence to play, the method’s feasibility becomes impaired.

We now propose a new method aimed at mitigating this trade-off through an increase in the computational burden required for the initialization. The main idea draws upon the use of a smoothing procedure within a training sample of data.

In order to understand the concept of smoothing, it is important to first define the concept of filtering, from which the former departs. In our context, filtered estimates are those obtained from the (forward) recursions associated to the learning algorithms in (1.2)-(1.3) and (1.4). To facilitate exposition in this section, we add another subscript to our previous notation: \( \hat{\theta}_{t|k} \), where \( t \) indicates the period the estimate stands for and \( k \) indicates the information period on which the estimate is based. Then, the filtered estimates are given by \( \hat{\theta}_{t|t} \equiv \hat{\theta}_t \).

An illustration of the time evolution of this filtering process is presented in Figure 2.1a. At each point in time, new filtered estimates are obtained as an update to the previous period estimates using the newly available observations. Clearly, as the arrow in Figure 2.1a indicates, the recursive nature of this estimation process introduces persistence to the effects of the initials.

The smoothed estimates, on the other hand, stand for (backward-looking) updated inferences on the filtered estimates, i.e., \( \hat{\theta}_{t|k} \) with \( k \geq t \). An illustration of this smoothing process is also presented in Figure 2.1b in connection
Figure 2.1: Illustration of filtering, smoothing, and the initialization routine.

(a) Filtering: forward-looking estimation.

\[ \hat{\theta}_{0|0} \quad \hat{\theta}_{1|1}(x_1, y_1, \hat{\theta}_{0|0}) \quad \hat{\theta}_{2|2}(x_2, y_2, \hat{\theta}_{1|1}) = \hat{\theta}_{2|2}(x_1, x_2, y_1, y_2, \hat{\theta}_{0|0}) \]

(b) Smoothing: backward-looking estimation.

\[ \hat{\theta}_{0|2} \quad \hat{\theta}_{0|1} \quad \hat{\theta}_{0|0} \quad \hat{\theta}_{1|1}(x_1, y_1, \hat{\theta}_{0|0}) \quad \hat{\theta}_{2|2}(x_1, x_2, y_1, y_2, \hat{\theta}_{0|0}) \]

(c) Initialization routine: repeatedly smoothing.

\[ \hat{\theta}_{0|N} \]
\[ \hat{\theta}_{0|0} = 0 \]
\[ \hat{\theta}_{0|0} = \hat{\theta}_{0|N} \]

Notation: \( \hat{\theta}_{t|k} \) stands for an estimate of the parameters vector \( \theta_t \) using information available up to period \( k \). The superscript in \( \hat{\theta}_{t|N}, \hat{\theta}_{0|N}, \ldots \) stands for the repetition index in the smoothing initialization routine.
to the filtering representation. There we can see that a new (smoothed) update to the initials (arrows pointing to the left) can be associated to every new filtered estimates, i.e., to every observation that becomes available after setting the initials. Clearly, while the filtered estimates stand for the inferences made on the basis of information available at the period the estimates stand for, the smoothed estimates are inferences obtained as new information about the system becomes available.

To obtain the smoothed initials associated to the learning algorithms, we make use of the parallel drawn in the previous chapter between these algorithms and the Kalman filter applied to the estimation of a time-varying parameters model. More specifically, we start from the exact correspondences derived in section 1.3 for both the LS and the SG algorithms under an unifying state-space framework. Then, we obtain the smoother associated to each of these algorithms by direct substitution of their correspondences with the Kalman filter into the Kalman fixed-point smoother of Anderson and Moore (1979, pp. 170-6). Essentially, these authors have shown how the fixed-point smoothing problem can be solved through the application of the standard Kalman filtering expressions to the original state space model augmented with a state appropriately initialized to represent the fixed-point smoothed estimates.

Due to the use of more information, one can expect the smoothed estimates to be more accurate than the filtered ones, and this is the reason we propose their usage for the initialization of learning algorithms. This gain in accuracy, however, comes at the cost of more computations and a delay incurred by waiting for the arrival of new observations. Such a delay, obviously, prevents

---

8Smoothing is usually carried out into one of three forms: (i) as fixed-point, fix $t$, and update the estimates of $\hat{\theta}_{i|k}$ as $k$ increases; (ii) as fixed-lag, set $k = t + l$, with $l$ fixed, and obtain $\hat{\theta}_{i|t+l}$ as $t$ increases; and, (iii) as fixed-interval, fix the information set $k$, and obtain $\hat{\theta}_{i|k}$ for $t \leq k$. For our purposes only (i) and (iii) are sensible, but given that our interest rests solely on an initial estimate we adopt the former, avoiding the need of a “backward pass” as in the latter.

9Key steps in this derivation are reproduced for convenience in appendix A.6.
the use of the smoothed estimates for learning-to-forecast applications, given that these estimates make use of information not available to real-time learning agents. Therefore, a sample of initial data, say of $N$ observations, is required to be set aside for the smoothing initialization procedure, just as it is required by the diffuse initialization methods.

Within this initial sample of data, then, one can start the computation of the learning algorithms from a diffuse prior, such as $\hat{\theta}_{0|0} = 0$, and obtain not only the algorithm’s filtered estimates up to $\hat{\theta}_{N|N}$, but also its smoothed estimates of $\hat{\theta}_{0|N}$. With these latter at hand, then, one re-starts the estimation process, within the same sample of data, but now assigning the initial in accordance to the smoothed estimate, i.e., $\hat{\theta}_{0|0}^i = \hat{\theta}_{0|N}^{i-1}$, where the superscript stands for the iteration index of this smoothing routine. A new sequence of filtered and smoothed estimates is in this way obtained, and this process can be repeated a few more times until a given convergence criterion is met. For this latter, here we adopted an $\epsilon$-convergence\textsuperscript{10} criterion based on the Euclidean distance between the smoothed estimates resulting from each iteration, under which the above process is repeated until $\|\hat{\theta}_{0|N}^i - \hat{\theta}_{0|N}^{i-1}\| < \epsilon$.

This initialization routine is illustrated in Figure 2.1c. Every such a repetition is represented by a sequence of filtered estimates and of smoothed updates to the initials. Clearly, within each of these repetitions, our interest is mainly on the last smoothed initial estimate we can obtain on the basis of the information available in our fixed training sample. But it is important to note that these repetitions are not worthless with respect to their computational cost. Even though the same set of data is supplied to the algorithms throughout these repetitions of filtering/smoothing estimations, the information provided is not the same. Namely, by changing the initial estimates, $\hat{\theta}_{0|0}$, the whole stream of subsequent filtered estimates is affected, and so is the information on the system dynamics that is incorporated into the smoothed estimates.

\textsuperscript{10}To avoid halting the computations for longer than necessary, we also imposed a stopping limit to the number of these repetitions.
2.4 Comparative simulation analysis

In this section we present simulation evidence comparing two of the initialization methods reviewed above, namely the diffuse-track and the diffuse-ordinary, to our own new method based on smoothing. The comparison is made with respect to their initial transient behavior in estimating the parameters of a given (known) non-stationary environment.

2.4.1 Setup

Our purpose here is to construct the (averaged) learning curves of the algorithms during their initial transient phase and evaluate how their statistical properties are affected by the initializations adopted. Consistent to our discussion in section 2.2.1, our focus is on the MSD and MSE measures of the algorithms’ performance, as given by (2.1) and (2.2), respectively. Given the stochastic environment under which these algorithms operate, in simulation studies these curves are computed as an average over repeated samples of artificially generated data.

The artificial data is generated according to a linear (1st order) autoregression of the form

\[ y_t = \theta_t y_{t-1} + e_t, \]  

(2.5)

where the autoregressive parameter evolves according to

\[ (\theta_{t+1} - \bar{\theta}) = \beta (\theta_t - \bar{\theta}) + w_{t+1}, \]  

(2.6)

and the random disturbances \( e_t \) and \( w_{t+1} \) are zero mean mutually independent distributed as Gaussian\(^{11}\) with variances given by \( \sigma_e^2 \) and \( \sigma_w^2 \), respectively.

\(^{11}\)See footnote 3 in chapter 1.
Note that, in spite of simplifying our analysis by focusing into a one-parameter specification, we are adopting a time-varying model similar to that of Doan et al. (1984) and Hamilton (1994, pp. 400-3). In particular, notice that if $|\beta| < 1$, then $\bar{\theta}$ may be viewed as the steady-state value of the autoregressive coefficient in (2.5). Yet, in order to avoid too quick variations in the statistical properties of the data, the value of $\beta$ is usually assumed to be very close to unity. In spite of resembling a random walk, this assumption prevents the dynamics of the autoregressive coefficient to be dominated by the noise variations in its stochastic disturbances.

For the calibration of $\sigma^2_e$, $\sigma^2_w$, $\bar{\theta}$, and $\beta$, we take the recommendations of Hamilton (1994, pp. 401-3) as a reference, though adjusting them to our context. One of these adjustments refers to the use of a higher $\sigma^2_w$ in order to accentuate the variations in the estimation environment, and further justify the use of constant-gain algorithms. For the parameters determining the variance and the dynamic persistence of the artificial series, $\sigma^2_e$ and $\bar{\theta}$, respectively, we attempt to obtain a calibration that mimics the statistical properties of inflation data. Using standard econometric tools, then, we fitted an univariate fixed-coefficient AR(1) process to this series (the data shall be described in the next section), and obtained calibrations for it as $\sigma^2_e = 0.9\hat{\sigma}^2$ and $\bar{\theta} = \hat{\theta}$, where $\hat{\sigma}^2$ stands for the above estimation residuals variance, and $\hat{\theta}$ is the associated autoregressive coefficient estimate. These calibrations are summarized in Table 2.2.

For this given calibration, we drew 1,000 different samples of the random disturbances and used them with the data generating process given by (2.5)-(2.6) for the generation of artificial series with a time dimension of 1,250 observations. We discarded the first 250 of these observations for each sample to avoid sensitivity to the series initializations, for which we used $y_0 = 0$ and $\bar{y}_0 = 0$.

---

12 We have also carried out a sensitivity analysis over the values assumed for $\sigma^2_e$ and $\bar{\theta}$, so as to mimic data on output growth too. Overall, our main conclusions were not affected, but we shall point out any relevant difference in what follows. For convenience these results are presented in appendix C.
Table 2.2: Calibration of parameters for simulation with inflation-like data.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Calibrations values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma^2_{e}$</td>
<td>Variance of $e_t$ in (2.5).</td>
<td>2.25</td>
</tr>
<tr>
<td>$\sigma^2_{w}$</td>
<td>Variance of $w_{t+1}$ in (2.6).</td>
<td>$7 \times 10^{-5}$</td>
</tr>
<tr>
<td>$\bar{\theta}$</td>
<td>Steady-state value of $\theta_t$.</td>
<td>0.80</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Persistence of deviations from $\bar{\theta}$.</td>
<td>0.999</td>
</tr>
<tr>
<td>(b) For algorithms:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\bar{\gamma}_1$</td>
<td>LS “low” constant learning gains.</td>
<td>0.02</td>
</tr>
<tr>
<td>$\bar{\gamma}_2$</td>
<td>LS “high” constant learning gains.</td>
<td>0.10</td>
</tr>
<tr>
<td>$\bar{\mu}_1$</td>
<td>SG “low” constant learning gain.</td>
<td>0.001</td>
</tr>
<tr>
<td>$\bar{\mu}_2$</td>
<td>SG “high” constant learning gain.</td>
<td>0.0205</td>
</tr>
<tr>
<td>(c) For initialization methods:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td>Size of initial sample of training data.</td>
<td>75</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Convergence tolerance for smoothed initials.</td>
<td>0.01</td>
</tr>
<tr>
<td>$S$</td>
<td>Maximum number of smoothing repetitions.</td>
<td>100</td>
</tr>
</tbody>
</table>

The calibrations for the artificial series follow those recommended by Doan et al. (1984) and Hamilton (1994, pp. 400-3), (roughly) adjusted to the variables of interest. The learning gain calibrations are first set for the LS, and then adjusted for the SG according to $\bar{\mu}_i = 2\bar{\gamma}_i / \left(\frac{\sigma^2}{1-\bar{\theta}}\right)^2$ in order to account for the scale dependency of this latter to the data variance, but for illustrative purposes $\bar{\mu}_2$ is set based on $\bar{\gamma}_2' = 0.40$ instead of $\bar{\gamma}_2$.

$\theta_1 = \bar{\theta}$. An example of such artificially generated series of data is presented in Figure 2.2. Remarkably, the inflation-like characterization of this series is clear from its high degree of persistence.

Apart from these calibrations for the artificial series, we also need to specify how we calibrated the algorithms’ learning gains. Here we first define a set of different values for the LS, which is not sensitive to the scale of the data, and then adjust these gains for the SG case. In order to do this conversion, we need to compute estimates for the upper bounds on the gain calibrations that still ensure stability for each algorithm. The main issue here lies in the determination of this upper bound for the SG algorithm, which is known to be sensitive to the scale of the data (see Evans et al., 2010). Without extending any longer on this issue here, we follow the recommendations of Haykin (2001, pp. 258-74) and compute the SG upper bound as $\bar{\mu}_{\text{max}} = 2 / \lambda_{\text{max}}$, where $\lambda_{\text{max}}$ stand for the maximum eigenvalue of the regressors covariance matrix, which for the case of (2.5) is simply given by the variance of $y_t$. The LS gain

Figure 2.2: Example of inflation-like artificially generated series.

The dashed line in panel (a) represents the coefficients’ steady-state values, $\bar{\theta}$, according to Table 2.2. The dashed lines in the panel (b) represent $\pm 2$ standard deviations bands around zero, as computed from (2.5) assuming $\theta_t = \bar{\theta}$, and the calibrations in Table 2.2.

calibrations, specified in Table 2.2 by $\gamma_1$ and $\gamma_2$, are then converted to the SG as $\bar{\mu}_i = \pi_{max} \left( \bar{\gamma}_i / \sigma_y^2 \right)$, $i = 1, 2$, where the variance of $y_t$ is approximated taking the autoregressive coefficient as fixed to its long run value, $\theta_t = \bar{\theta}$.

For the case of the LS algorithm, it also remains to specify how we proceed to initialize the matrix of moments associated to (1.3). We follow Moustakides (1997) rule, under which

$$R_0 = \gamma_0^\alpha \Theta_\alpha^2$$

where $\Theta_\alpha^2$ is the variance (matrix) of the regressors in (1.1) and $\alpha$ is a parameter to be calibrated according to the signal-to-noise ratio of (1.1). For our purposes we simply set $\alpha = 1$, and compute $\Theta_\alpha^2$ on the basis of (1.1) as an autoregression with $\theta_t$ fixed to its (expected) long run value.

The set of generated series, totalizing 1,000, were then supplied to different combinations of algorithm/gain-initialization to obtain the associated $\hat{\theta}_t$ estimates. These latter were then used for the computation of the MSD and MSE statistics, following (2.1) and (2.2), respectively, averaged over the one thousand replications of data samples.
2.4.2 Simulation results

The criterion under which each initialization is evaluated is the MISALIGNMENT of the initial estimates from their corresponding algorithm/gain long run behavior, as defined in (2.3) and (2.4). Here, firstly, we evaluate this criterion visually leaving its quantification to the end. The MSD\textsuperscript{13} learning curves obtained from the application of the LS and the SG algorithms to inflation-like data are presented in Figure 2.3. For each of the methods considered, we have fixed the number of observations taken for training to the first 75, which are highlighted in the figures by shaded areas.

The initial MSD-MISALIGNMENT incurred by each initialization method depends on the gain calibration, and the same dependency is observed with respect to the algorithm’s MSD steady state level, as expected. This is evident in Figures 2.3a and 2.3b by the jumps undertaken by the MSD estimates from their after-initialization level to their stable long run level. These jumps are more remarkable for the diffuse-ordinary initializations, an observation that corroborates our previous point that the diffuse-ordinary method tends to violate the requirement on the initials’ COHERENCE.

The diffuse-track also seems to perform poorly with respect to that COHERENCE criterion, but this is more clearly evident from its application to the SG algorithm: the lack of a normalization step in the operation of this algorithm seems to be reflected in its slowly rate of convergence to steady state. The number of observations left aside for the diffuse-track initialization of the SG algorithm is clearly too small to permit convergence under the smaller gain calibration, corroborating our statement that this method lacks on FEASIBILITY.

The only method that seems to be performing consistently for all the above criteria and throughout the different algorithms and gain calibrations is our

\textsuperscript{13}MSE learning curves do not bring much content to this analysis as they present a faster convergence than MSD measures and their visualization is more polluted by noise.
Figure 2.3: MSD learning curves for inflation-like artificial data.

(a) Least Squares.

(b) Stochastic Gradient.

The shaded areas indicate the portion of observation left aside for use by the initialization methods. \( D_t \) stands for the sample correspondent to the mean-square deviation (MSD) as defined in (2.1). The vertical axis is on logarithmic scale. See the text for further details on how these curves were computed, and Table 2.2 for the gain calibrations.
own smoothing procedure. In any of the cases considered in Figures 2.3a and 2.3b, this method tended to provide initial estimates that were closer to the algorithm/gain steady state behavior compared to those provided by the other two methods. The smoothing procedure has, in special, presented a better performance for the cases where the other methods have failed, namely: (i) for higher gain calibrations in the LS, where resulting estimates were less accurate; and (ii) for lower gain calibrations in the SG, where the rate of convergence tended to be slower.

Regarding the relevance of these findings, although it is hard to judge results for the SG algorithm, for the LS it is usually the case that in the literature the learning gain is calibrated to values close to our “low” gain, \( \gamma_1 \). Results in Figure 2.3a indicate that for such an economically meaningful calibration of the LS algorithm, the three methods of initialization here evaluated tend to lead to similar results in terms of initial MSD-misalignments to its steady state behavior. This is remarkably true for the diffuse-track and the smoothing methods, while for the diffuse-ordinary one there is some evidence indicating a higher degree of initial MSD-misalignment. Notice, however, that the use of less than 75 observations for the initialization of this algorithm/calibration would unequivocally drive the initial estimates obtained from the diffuse-track and the diffuse-ordinary methods into incoherent estimates. Hence, the feasibility of these methods may still be impaired in cases of tight data availability.

To add precision to the observations made above, we complement the visual analysis with a look over the associated statistics. For this purpose we present in Table 2.3 average MSD statistics for both the LS and the SG algorithms. For each of these tables, the averaged statistics are segmented in several sub-samples after the initializations, in an attempt to obtain short run measures corresponding to the transient phase undertaken by the algorithms after the initials.

We can see that our main observations from the visual inspection are here
Table 2.3: Average MSDs after initializations on inflation-like data.

(a) Least Squares.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>76-100</td>
<td>101-150</td>
</tr>
<tr>
<td>( \gamma_1 = 0.02 )</td>
<td>Diffuse-track</td>
<td>0.0054</td>
<td>0.0047</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[4.3]</td>
<td>[0.4]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0060</td>
<td>0.0047</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[7.9]</td>
<td>[0.6]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0051</td>
<td>0.0043</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2.6]</td>
<td>[-1.7]</td>
</tr>
<tr>
<td>( \gamma_2 = 0.10 )</td>
<td>Diffuse-track</td>
<td>0.0175</td>
<td>0.0175</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-1.3]</td>
<td>[-1.4]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0064</td>
<td>0.0139</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-11.5]</td>
<td>[-4.7]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0174</td>
<td>0.0174</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-1.3]</td>
<td>[-1.4]</td>
</tr>
</tbody>
</table>

(b) Stochastic Gradient.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>76-100</td>
<td>101-150</td>
</tr>
<tr>
<td>( \mu_1 = 0.001 )</td>
<td>Diffuse-track</td>
<td>0.1407</td>
<td>0.0944</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[680.5]</td>
<td>[448.5]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0790</td>
<td>0.0544</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[358.4]</td>
<td>[240.0]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0061</td>
<td>0.0057</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[7.6]</td>
<td>[5.7]</td>
</tr>
<tr>
<td>( \mu_2 = 0.0205 )</td>
<td>Diffuse-track</td>
<td>0.0224</td>
<td>0.0223</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.1]</td>
<td>[-0.1]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0481</td>
<td>0.0240</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[7.4]</td>
<td>[0.3]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0180</td>
<td>0.0193</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.5]</td>
<td>[-0.1]</td>
</tr>
</tbody>
</table>

The average statistics refer to the mean-square deviation of coefficient estimates from their true counterparts, as defined in (2.1). The second line of headers indicate the samples of observations used to compute the average statistics. The values in round brackets, (...), are standard deviations of the statistic from the corresponding steady state average. The values in square brackets, [...], refer to the number of (steady state) standard deviations by which the corresponding short run average deviates from the steady state average, and correspond to our definition of MSD-MISALIGNMENT in (2.3). Emphasis is given in **bold** to those short run averages that deviate by more than two standard deviations from the corresponding steady state average.
corroborated: (i) the \textit{diffuse-ordinary} method is overall outperformed by the others, being the only method for which the magnitude of initial \textsc{misalignments} persist to affect the first short run measures, for each algorithm and gain calibration; (ii) the SG rate of convergence is slower than the one attained by the LS, and the \textit{smoothing} method is the only one providing initializations closer to the algorithm/calibrations steady states\textsuperscript{14}.

Finally, we now shift our focus to the results for the MSE statistic presented in Table 2.4. Here the initial \textsc{misalignments} associated to each method of initialization are smaller than those observed for the MSD measure: the MSE measures are found to converge closer to their long run levels shortly after initialization for most cases presented. But we can also observe some cases under which the \textit{diffuse-track} and the \textit{diffuse-ordinary} initials do not converge to long run behavior as quickly as the \textit{smoothing} initials do, especially for the SG algorithm under smaller gain calibration.

Another interesting observation that can be drawn from the results on the MSE statistics relates to their long run levels. From their definition, in (2.2), the natural reference for this measure is the variance of the disturbance term $\varepsilon_t$ in (1.1). For the case of inflation, $\sigma^2_\varepsilon = 2.25$, and hence results presented in Table 2.4 indicate that the estimation errors associated to both algorithms were misaligned downwards in relation to the true disturbances affecting the system. This is in contrast to the case of growth-like data, presented in Table C.3 in the appendix, for which the misalignment tends upwards.

One explanation for this difference relates to the lower signal-to-noise ratio of the growth-like series. This is clear from the higher variance of $\varepsilon_t$ and the lower (steady-state) autoregressive coefficient $\bar{\theta}$ that characterize the growth-like series, compared to inflation. Other things equal, therefore, the above results indicate that the MSEs associated to both the LS and SG algorithms tend to decrease, relative to the true stochastic disturbances affecting the system, as

\textsuperscript{14}These observations were also corroborated for the growth-like data, although with some deterioration of the short run MSD measures. See appendix C.
Table 2.4: Average MSEs after initializations on inflation-like data.

(a) Least Squares.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>76-100 101-150 151-200 201-250 251-300 750-1000</td>
<td></td>
</tr>
<tr>
<td>( \gamma_1 = 0.02 )</td>
<td>Diffuse-track</td>
<td>1.6416 1.6571 1.6243 1.6511 1.6425</td>
<td>1.6431</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.0] [0.2] [-0.2] [0.1] [-0.0]</td>
<td>(0.0811)</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>1.6332 1.6497 1.6150 1.6433 1.6367</td>
<td>1.6366</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.0] [0.2] [-0.3] [0.1] [0.0]</td>
<td>(0.0813)</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>1.6356 1.6532 1.6203 1.6482 1.6401</td>
<td>1.6412</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.1] [0.1] [-0.3] [0.1] [-0.0]</td>
<td>(0.0811)</td>
</tr>
<tr>
<td>( \gamma_2 = 0.10 )</td>
<td>Diffuse-track</td>
<td>1.6510 1.6771 1.6403 1.6772 1.6615</td>
<td>1.6653</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.2] [0.1] [-0.3] [0.1] [-0.0]</td>
<td>(0.0851)</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>1.5818 1.6581 1.6378 1.6749 1.6592</td>
<td>1.6632</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-1.0] [-0.1] [-0.3] [0.1] [-0.0]</td>
<td>(0.0851)</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>1.6433 1.6693 1.6327 1.6698 1.6543</td>
<td>1.6578</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.2] [0.1] [-0.3] [0.1] [-0.0]</td>
<td>(0.0851)</td>
</tr>
</tbody>
</table>

(b) Stochastic Gradient.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>76-100 101-150 151-200 201-250 251-300 750-1000</td>
<td></td>
</tr>
<tr>
<td>( \mu_1 = 0.001 )</td>
<td>Diffuse-track</td>
<td>\textbf{2.4229} \textbf{2.1771} \textbf{1.8720} 1.7974 1.7259</td>
<td>1.6461</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[9.5] [6.5] [2.8] [1.9] [1.0]</td>
<td>(0.0815)</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>\textbf{2.0404} \textbf{1.9353} 1.7512 1.7235 1.6815</td>
<td>1.6390</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[4.9] [3.6] [1.4] [1.0] [0.5]</td>
<td>(0.0818)</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>1.6419 1.6635 1.6264 1.6479 1.6386</td>
<td>1.6424</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.0] [0.3] [-0.2] [0.1] [-0.0]</td>
<td>(0.0813)</td>
</tr>
<tr>
<td>( \mu_2 = 0.0205 )</td>
<td>Diffuse-track</td>
<td>1.5765 1.5599 2.0410 1.4939 1.7183</td>
<td>1.5803</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.0] [-0.1] [1.5] [-0.3] [0.4]</td>
<td>(0.0307)</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>\textbf{2.2243} 1.5836 \textbf{2.2459} 1.5119 1.7430</td>
<td>1.5968</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[2.0] [-0.0] [2.1] [-0.3] [0.5]</td>
<td>(0.3071)</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>1.2622 1.3607 1.8352 1.2869 1.5248</td>
<td>1.3778</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.4] [-0.1] [1.5] [-0.3] [0.5]</td>
<td>(0.3027)</td>
</tr>
</tbody>
</table>

The average statistics refer to the mean-square error of estimation, as defined in (2.2). The second line of headers indicate the samples of observations used to compute the average statistics. The values in round brackets, (...), are standard deviations of the statistic from the corresponding steady state average. The values in square brackets, [...], refer to the number of (steady state) standard deviations by which the corresponding short run average deviates from the steady state average, and correspond to our definition of MSE-MISALIGNMENT in (2.4). Emphasis is given in bold to those short run averages that deviate by more than two standard deviations from the corresponding steady state average.
the data signal-to-noise ratio (SNR) increases\textsuperscript{15}.

### 2.5 Empirical application

In this section we attempt to check the evidence about different initialization methods that was derived in the previous section from simulations, by means of an empirical application with US macroeconomic data on inflation and growth. It is not our purpose to be exhaustive in this application, but only to provide an assessment of how these initialization methods perform, comparatively, under a simplistic context of applied macroeconomics. Before going through the results, we begin with a brief description of the data and model specification to be used in conjunction to the LS and the SG algorithms to assess the effects that three different initial estimates have over the properties of the forecasts for inflation\textsuperscript{16}. For consistency, we evaluate the same methods of initialization, namely, the \textit{diffuse-track}, the \textit{diffuse-ordinary}, and the \textit{smoothing} methods.

#### 2.5.1 Data, model\textsuperscript{17}, and algorithms’ calibrations

We use quarterly data on the US real GDP and its price index from 1947q1 to 2011q4. Our data on this series comes from the Philadelphia’s Fed Real-Time Data Research Center from which we used the series observed at the vintage of 2012q1. For simplicity, in this chapter we are neglecting real-time data issues by focusing on an unique snapshot of the realization of these series. We also simplify our estimation by subtracting the mean of each series, precluding the need of intercepts in our model specification. This gives us a total of 259

\textsuperscript{15}We shall return to this observation when interpreting results associated to periods of instability in the next chapter.

\textsuperscript{16}Again, our presentation will focus on results for inflation, even though we have also used data on growth for VAR estimations. For the results on growth forecasts, please refer to appendix C.

\textsuperscript{17}The data and model specification we adopt in this section follow the definitions given in section 1.4.2, though we briefly restate them here for completeness.
observations for inflation and output growth, which we again denote by $\pi_t$ and $g_t$, respectively.

We then use a simple unrestricted VAR(1) specification to model these series, where coefficients are updated recursively for each new observation made available through time. These estimates are obtained separately for each algorithm, i.e., the LS and the SG, and also for each different initialization method. The model specification we are using can then be expressed as

$$
\begin{bmatrix}
\pi_t \\
g_t
\end{bmatrix} =
\begin{bmatrix}
\hat{\phi}_{\pi,t} & \hat{\phi}_{g,t} \\
\hat{\phi}_{\pi,t} & \hat{\phi}_{g,t}
\end{bmatrix}
\begin{bmatrix}
\pi_{t-1} \\
g_{t-1}
\end{bmatrix} +
\begin{bmatrix}
\hat{\epsilon}_{1,t} \\
\hat{\epsilon}_{2,t}
\end{bmatrix},
$$

(2.8)

where $\hat{\phi}$ and $\hat{\phi}$ stand for the coefficients’ estimates associated to the equations having inflation and growth as endogenous, respectively, and their subscripts indicate the explanatory variable to which they are attached and the period from which their estimates are made. The residual terms $\hat{\epsilon}_{1,t}$ and $\hat{\epsilon}_{2,t}$ stand for estimation errors.

The initializations are computed on the basis of a training sample of 75 initial observations, as we did in our simulation exercise. In terms of the time span of our sample, this means we left aside data from 1947q2 to 1965q4, which is (approximately) in line with the previous applied literature on adaptive learning in macroeconomics. Orphanides and Williams (2005a), for example, use data up to 1965q4 to initialize the learning algorithms, while Branch and Evans (2006) and Milani (2011) expand this information set with a few more observations, using data up to 1969q4 and 1968q4, respectively.

It remains to specify how we calibrate the learning gain of the LS and the SG estimation algorithms. In the spirit of the calibration we used for the simulation exercises, we first define the gain for the LS and adjust this for the data-dependent context of gain calibration for the SG algorithm. To be consistent with the previous literature, the reference calibration we adopt for the LS is given by $\bar{\gamma} = 0.025$. For the SG algorithm, we use expressions similar
to those we adopted for the simulation exercises, though with the regressors covariance matrix being estimated using the initialization data\textsuperscript{18}. Doing that we obtain the following calibration for the SG gain\textsuperscript{19}: $\bar{\mu} = 0.0046$.

### 2.5.2 Empirical results

Our analysis is based on statistics of 1-period ahead forecasts that can be straightforwardly computed from the VAR specification in (2.8) for inflation, together with coefficients estimates obtained from each combination of algorithm, gain calibration, and initialization. Our focus on forecasts is naturally due to the learning-to-forecast rationale we are adopting. Also, our lack of knowledge about the true system coefficients prevents us from evaluating MSD measures directly.

It is instructive to have a look at the coefficient estimates that each algorithm provides for the two-variables VAR(1) model of (2.8), and how these estimates are affected by the method used to initialize the algorithms’ operations. Given our focus on inflation forecasts, we present in Figure 2.4 the estimates for the inflation persistence coefficient, $\hat{\phi}_{\pi,t}$, obtained from the LS and SG algorithms, respectively.

Corroborating our simulation findings, the differences between the initial estimates presented by each method of initialization are again more pronounced for the diffuse-ordinary method. As expected, these initial differences tend to die out as observations accumulate over time, with only small differences remaining after say observation 150, which in our sample stands for

\textsuperscript{18}Other than for the calibration of the SG gains, we also used the covariance matrix estimated on the basis of initialization data for the initialization of $\mathbf{R}_0$ in the LS algorithm. See Equation 2.7. Our use of only initialization data is an attempt to prevent any form of “cheating in forecasting” by the learning algorithms in what they stand as representative of agents’ expectations.

\textsuperscript{19}The higher SG gain value here, compared to those in the other chapters, stems both from the use of a smaller sample in the estimation of the regressors covariance matrix and the use of a slightly different method for the computation of the upper bound to the SG gain; compare the expressions in sections 1.3.4 and 2.4.1.
Figure 2.4: Estimates of inflation autoregressive coefficient.

(a) Least Squares.

(b) Stochastic Gradient.

The plotted estimates refer to those obtained with a unique learning gain for each algorithm, namely $\hat{\gamma} = 0.025$ and $\hat{\mu} = 0.0046$, as specified in the text. The shaded area indicates the portion of observations left aside for use by the initialization methods.
A deeper understanding of these differences in estimates can be obtained through statistics computed over the forecasts associated to them. Here our focus goes to their variances and MSFE, presented in Table 2.5. While the MSFE stands for a useful measure of performance comparisons, it is mainly in the forecasts’ variance that lies our interest. Learning-to-forecast behavior provides the channel through which shocks may persist over time, rather than perishing instantaneously as implied by RE. Recent studies have found support to this view by incorporating forecasts generated through learning algorithms as an additional explanatory variable in business cycle modeling (see, e.g., Orphanides and Williams, 2005a; Milani, 2011, between others cited above). Hence, it is important to assess whether a given initialization method is distorting the variance of the forecasts associated to each learning algorithm in relation to its long run behavior.

In accordance with the previous applied literature we can further narrow our focus to the LS case. Focusing on the first sub-samples we see that the diffuse-ordinary method rendered forecasts with lower variances than the other methods. Compared to the smoothing method, for example, the forecasts obtained departing from the diffuse-ordinary initials presented variance around 18% lower during the first sub-sample (1966q1-1971q4). It should be emphasized that the result that the diffuse-ordinary method delivers forecasts with a lower variance than the others does not constitute a point in favor of this method, and neither, necessarily, against it. Bear in mind that we are interested in assessing the initials coherence to the algorithm’s long run behavior, rather than in a minimization of forecasts variance or their associated errors\textsuperscript{21}.

Before drawing a conclusion on this point, one additional observation is

\textsuperscript{20}Another observation is that the SG estimates appear to be more volatile than those of the LS, which could well be explained either as a difference between the algorithms behavior, or as a gain calibration feature.

\textsuperscript{21}In that respect, notice that the diffuse-ordinary presented slightly higher MSFEs than the other methods during the first two sub-samples under LS estimation, and also during the second sub-sample under SG estimation.
Table 2.5: Statistics on inflation forecasts.

<table>
<thead>
<tr>
<th>Initializations</th>
<th>Samples after initializations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>76-100</td>
</tr>
<tr>
<td>(a) Least Squares estimates with $\gamma = 0.025$:</td>
<td></td>
</tr>
<tr>
<td>Diffuse-track</td>
<td>0.82</td>
</tr>
<tr>
<td>(2.09)</td>
<td>(2.69)</td>
</tr>
<tr>
<td>Diffuse-ordinary</td>
<td>0.61</td>
</tr>
<tr>
<td>(2.17)</td>
<td>(2.78)</td>
</tr>
<tr>
<td>Smoothing</td>
<td>0.74</td>
</tr>
<tr>
<td>(2.14)</td>
<td>(2.78)</td>
</tr>
<tr>
<td>(b) Stochastic Gradient estimates with $\mu = 0.0046$:</td>
<td></td>
</tr>
<tr>
<td>Diffuse-track</td>
<td>0.62</td>
</tr>
<tr>
<td>(1.93)</td>
<td>(2.99)</td>
</tr>
<tr>
<td>Diffuse-ordinary</td>
<td>0.35</td>
</tr>
<tr>
<td>(1.94)</td>
<td>(3.28)</td>
</tr>
<tr>
<td>Smoothing</td>
<td>0.68</td>
</tr>
<tr>
<td>(1.95)</td>
<td>(2.93)</td>
</tr>
</tbody>
</table>

The statistics presented here refer to variances of forecasts (first row for each algorithm/initial) and mean squared forecast error (in round brackets at second row for each algorithm/initial). The second line of headers indicates the samples of observations used to compute these statistics.

useful. A smaller difference is observed between the variances of the forecasts obtained from the *diffuse-track* and the *smoothing* methods. The forecasts’ variance from the former initialization procedure was found to be around 10% and 15% higher than that from the latter, during, respectively, the first and the second sub-samples.\(^{22}\)

In contrast to our previous simulation exercise, though, our knowledge about the true nature of the series under estimation is restricted at the empirical level. This prevents us from obtaining a clear-cut answer to which of the initialization methods provides a higher degree of coherence between the initial estimate and the algorithm’s long run behavior. Nonetheless, our simulation results suggested that the *diffuse-ordinary* method tends to be more distortionary than the other methods. Together with the empirical results found

\(^{22}\)These quantitative results were more sensitive to the variable being forecasted, and the gain calibration used, but still such increased sensitivity had no effect over our conclusions. Please refer to appendix C for the results on growth.
here, then, our study indicates that in order to initialize learning algorithms: (i) there is not much difference between using the diffuse-tracking or the smoothing methods to initialize learning algorithms at usual calibrations, although the latter has the advantage of being robust to changes in algorithms/gain calibrations; (ii) using the diffuse-ordinary method results in initial forecasts with a lower variance than those obtained from the other methods, and our simulations indicate that this difference represents a distortion in relation to the algorithm’s long run operation.

2.6 Concluding remarks

In this chapter we provided a critical review on the several methods previously proposed in the literature of learning and expectations in macroeconomics in order to initialize its learning algorithms either for simulation or for empirical purposes. Most importantly, we have also provided one of the first attempts in the literature to evaluate how these methods compare to each other, and how their performance may be evaluated with respect to their learning and expectations rationale. To delineate the scope of our analysis, we focused on two of the main algorithms found in this literature, namely, the Least Squares (LS) and the Stochastic Gradient (SG) algorithms.

Before pooling the initialization methods in a classification exercise, we provided a discussion on what it is required from them, arguing for the use of two main criteria. First, an initialization should provide initial estimates coherent to the algorithm’s long run behavior. Second, such a coherence must be feasible within the data availability restrictions of usual macroeconomic applications. Our finding is that none of the previous methods reviewed is able to pass the sieve of both criteria, and this motivated us to propose a new method.

Departing from exact correspondences between the learning algorithms and the Kalman filter associated to a time-varying hypermodel, as drawn in
chapter 1, we proposed the use of a smoothing routine to obtain the initial estimates for each algorithm. This routine makes use of a sample of initial training data, and it is designed to satisfy the above requirements of coherence and feasibility in exchange for additional computational costs. In order to evaluate its success, we undertook both a simulation exercise and an empirical application, comparing our new smoothing-based initialization to two of the methods found in the previous applied literature.

From the simulation exercise, the main conclusion we can draw is that our method is successful in satisfying what we required from an initialization while the previous ones achieved only partial success in that respect. Namely, our smoothing-based routine was the only method performing consistently throughout the various applications we have explored, under different algorithms, calibrations and statistical environments. We interpret this finding as a natural result from the unified design we adopted for the derivation of our smoothing initialization method.

A different question, however, is how much the differences we found across methods are relevant for actual applications of these algorithms into macroeconomic contexts. To shed some light on this issue, we compared the initialization methods in a simplified empirical application of learning-to-forecast US data on inflation and output growth. Using a sample of quarterly data from 1947 to 2011, where data up to 1965 is left aside for the initialization of the algorithms, our results indicate that the effects of the different initialization methods last no longer than mid-1980s. For the preceding sample periods, nonetheless, our results indicate that the initialization method can distort the variances of the forecasts constructed using the learning algorithms. Even though we have quantified these effects for our simplified application, their actual relevance will depend on the issue under scrutiny.
Chapter 3

The determination and calibration of the learning gains

3.1 Introduction

One recurrent issue in the application of adaptive learning algorithms in order to mimic the process through which agents form their expectations refers to the calibration of these algorithms. The computational operation of these algorithms requires the pre-specification of a sequence of learning gain values, or of a mechanism through which these gains are determined in real-time. This chapter is devoted to investigate this issue empirically, thus producing renewed estimates of the learning gains. We hope these results will serve as guidance for the calibration of the learning algorithms in applied macroeconomics.

We also introduce in this chapter an insightful distinction between two assumptions on the rationale given to the learning gain. The first view is that the learning gain is determined as a choice by the agents adopting a given learning algorithm to update their expectations. The second view, in contrast, is to assume that the learning gain stands as a primitive parameter of agents learning-to-forecast behavior. In spite of their evident dissimilarity, an understanding of the effects of these different assumptions over the resulting gain calibrations seems to have been neglected in the previous literature.

In order to shed some light on this point we adopt the evaluation framework proposed in section 1.4. Recall we defined there the learning-to-forecast exercises, which attempt to mimic the real-time process of expectation formation.

---

1Some results in this chapter have been previously presented in Berardi and Galimberti (2012b).
tion. Using such framework we evaluate empirically the performance of the distinct approaches to the calibration of the learning gains, again focusing on both the LS and the SG algorithms. We analyze the quality of the forecasts produced by each calibration along the two dimensions we have previously defined: their forecasting accuracy, and their resemblance to surveys.

We carry out these exercises using real-time quarterly data on US inflation and output growth covering a broad post-WWII period of time, from 1947q2 to 2011q4. Our results provide strong evidence in favor of the gain as a primitive approach, although the performances of particular calibrations of the learning algorithms, applied to forecast different variables, are found to be rather sensitive to the samples used for selection of the gains.

The remainder of this chapter proceeds as follows. In section 3.2 we relate our work to the previous literature, also emphasizing the relevance of the learning gain and its interpretations. We then proceed, in section 3.3 with a precise specification of the different learning gain calibrations we evaluate, leaving the presentation and discussion of their comparative results to section 3.4. Finally, we conclude this chapter with some remarks in section 3.5.

3.2 Theoretical background

Applied research requires going beyond the incipient debate between the use of a decreasing or a constant sequence of gains, and at this point the researcher is usually left with a decision between two calibration approaches. The first is to assume that the learning gain is determined as a choice by the agents adopting a given learning algorithm to update their expectations. The second, in contrast, is to assume that the learning gain stands as a primitive parameter of agents learning-to-forecast behavior.

Recognizing the prominent role that the learning gain has in determining the statistical properties of the forecasts associated to the learning algorithms, it is clear that these alternative assumptions are relevant on both theoretical
and empirical grounds. In this section we review the theoretical literature to relate each approach of gain calibration to the different perspectives the theory of adaptive learning has been proposed to model expectations in macroeconomics.

### 3.2.1 Gain as a choice

One important debate related to the traditional assumption of decreasing gains was initiated by an intriguing observation made by Bray and Savin (1986). Adaptive learning attempts to capture agents’ expectations off the equilibrium path, which implies that until convergence is reached agents’ perceived law of motion (PLM) is changing. Thus, even if the structural model is assumed to be time-invariant, learning itself introduces time-variation in the economy’s actual law of motion (ALM). Statistical theory, however, indicates that decreasing gains estimators are not appropriate for estimation of models with time-varying parameters. Therefore, it would be reasonable to expect that even if agents commence learning with decreasing gains, they would eventually have collected enough evidence to prompt a change towards tracking-consistent gain calibrations.

The natural follow up to this compelling argument by Bray and Savin (1986) was the incorporation of more sophisticated mechanisms into the analysis of learning convergence. Assuming agents learn using Kalman filtering techniques Margaritis (1990) was able to show convergence to RE in a model with time-varying parameters, though still requiring that the gains tend to zero. Relaxing this latter assumption, however, Bullard (1992) found that convergence results are lost if agents’ PLMs are not restricted to become (asymptotically) time-invariant; hence there would be no reason, on rationality grounds, to expect vanishing learning gains. Finally, a reconcilement between these results was proposed by McGough (2003, p.120): “for convergence to a RE equilibrium to occur, the agents must believe that the conditional variance of
the time-varying parameters decreases to zero.” If the only source of time-
variation comes from learning, convergence to RE will indeed lead the esti-
mates of this conditional variance to zero².

This seemingly circular reasoning actually reflects much deeper implica-
tions on the internal consistency between adaptive learning and convergence
to RE: either that agents initially believe in such a convergence, thus not aban-
doning the decreasing gain during out-of-equilibrium learning for its poor
performance; or that agents are aware of this trade-off between tracking and
convergence, and attempt to guarantee the latter at the same time that tun-
ing their gain calibration to improve on the former during transient dynamics.
Clearly, this last upshot from this debate provides a rationale for the view that
the learning gains are determined as a choice by the agents.

Finally, we can also relate the approach of gain calibration as a choice with
alternative approaches to bounded rationality. One popular alternative has
been the adaptively rational expectations approach of Brock and Hommes
(1997), where agents are assumed to select between a set of predictors to form
their expectations. The selection is determined according to a discrete-choice
model, which is microfounded in a random utility framework, depending on
fitness measures associated to each predictor’s past performance. One appealing
feature of this approach is that a diversity of predictors may be jointly in
use at a given period depending on how the intensity of choice is regulated.

Another related approach is the “expectation calculation” approach of Evans
and Ramey (1998), where agents are assumed to face a calculation decision
pondering between the benefits and the costs of extra cognitive efforts to form
expectations. Then, in a similar vein to the optimizing behavior we assume
to impose discipline on agents choice of gains, their agents are assumed to op-
timize on their choice of calculation intensity. More in line with our above

²As McGough (2003) appropriately recognizes, the conditional variance of the system’s
time-varying coefficients (i.e., the Ω in Eq. 1.19) has to decrease at a sufficiently high rate for
this argument to hold.
discussion about internal consistency, Marcet and Nicolini (2003) propose a threshold rule under which agents learn with decreasing gains during stable periods, but switch to a constant gain when some instability is detected through large prediction errors.

3.2.2 Gain as a primitive

From a theoretical point of view, the learning gain is determinant both of whether convergence to a RE equilibrium takes place and of its transitional dynamic properties. In order to obtain positive convergence results, the traditional analysis of learning usually places strong restrictions in the sequence of gains. Examples of these restrictions include the requirement of decreasing gains or that of a “small” constant gain to guarantee weaker (in distribution) convergence results (see Evans and Honkapohja, 2001, Ch.7). Furthermore, under the constant gain specification a phenomenon known as “escape dynamics,” recurrently pushing the economy away from its equilibrium, has been found to have its frequency of occurrence associated to the value of the learning gain (Cho et al., 2002).

An understanding of the interplay between these features is provided by Sargent and Williams (2005), who establish a Bayesian interpretation to the LS algorithm, later extended to the SG case by Evans et al. (2010). Such derivations are obtained along similar lines to those we explored in establishing our unified framework in chapter 1: assuming data is generated by a model with time-varying parameters governed by a random walk, it is well known that the Kalman filter provides the Bayes optimal estimator; drawing correspondences between the learning algorithms and the Kalman filter is, thus, equivalent to finding the conditions under which the learning algorithms approximate the Bayesian estimator.

Under the Bayesian interpretation, however, these correspondences define particular priors imputed as agents beliefs about the statistical properties of
the data. Namely, recall from chapter 1 that our correspondences required particular specifications of the variances of the stochastic disturbances affecting both the signal ($\sigma_t^2$) and the parameter drifting ($\Omega_t$) equations. Besides, other than relating the learning algorithms to the Kalman filter, the priors also carry implications about the specific calibrations of the learning gains within each algorithm. Given that in the Bayesian tradition these priors are specified in advance to estimation, this interpretation provides an appealing rationale for the view that the learning gains stand as primitive parameters of agents learning-to-forecast behavior.

Another useful interpretation given to the learning gains is that of determining the memory of the learning mechanism (see, e.g., Barucci, 1999; Honkapohja and Mitra, 2006). Recalling our results from section 1.2.2, we have seen that the gains determine the weights the learning algorithms assign to each observation in updating their estimates. Namely, the higher the gain, the higher the emphasis given to more recent observations; hence, higher gains imply a lesser memory of the past observations of the data in the current estimates provided by the learning algorithms. Recognizing that it would be forcible to think of an agent’s memory as a choice, this interpretation provides an alternative rationale to the calibration of gains as a primitive.

To conclude this section notice that irrespective of the rationale adopted to justify the gain as a primitive approach, the fact that the learning gain is taken as a deep parameter of agents characteristics does not preclude that these gains may differ depending on the variable, or through time. In fact, it is quite reasonable to think that the gains will present such variations given their relevance in determining the quality of the inferences obtained through the learning algorithms.
3.3 Learning gain calibrations

3.3.1 Approach overview

In this section we describe the different learning gain calibrations we evaluate. Our empirical strategy is to adopt the learning-to-forecast exercises we proposed in section 1.4. Particularly, we will construct series of forecasts for US inflation and output growth, using the LS and the SG algorithms together with real-time data on those variables to estimate VAR model specifications, as described in section 1.4.2. The distinguishing feature of these forecasts is, of course, the use of different gain calibrations.

To approach empirically the gain determination issue, we associate to the alternative assumptions distinct measures of gain selection. Under the first alternative, additional discipline is required to model the decision problem by assuming that agents would be willing to optimize on their choice of a gain. Given that the ultimate purpose of the learning algorithms is to represent agents process of learning-to-forecast, a natural measure to select the gain calibrations under this assumption is the accuracy of their associated forecasts.

As for the second alternative, taking the learning gain as a primitive obviously does not require any additional behavioral assumption. Nevertheless, an appropriate measure of fit under this alternative requires the actual observation of agents behavior of learning-to-forecast. Consistent to our previous analysis, we once again use data of survey forecasts as a proxy. Therefore, in order to obtain a calibration of the learning gains as agents primitive parameters we can adopt a measure of resemblance of their associated forecasts to those forecasts observed from actual agents collected through surveys.

The main challenge to the calibration of the learning gains, however, is about how these measures are evaluated in order for a particular gain to be selected for the algorithm’s recursive computations. This issue clearly represents an important obstacle for the view of the gains as an agent’s choice; namely, one
has to be careful about the conciliation of this choice with the real-time informational assumptions underlying the learning process.

Another related issue is about the specification of the range, or the options, of gain values of interest. We follow two computational approaches to obtain the gain values. In the first we begin by constructing a grid of admissible values, and then proceed imposing selection rules that will represent our different assumptions on the role of the gains. The second approach involves the use of an outer mechanism to adaptively adjust the gain in response to changes in the recent performance of each algorithm.

### 3.3.2 Grid-based

Under the grid-based approach we first need to establish which options we consider available to agents, and then discuss how the gains are selected among the available alternatives. Following our discussion in Section 1.3.4, we construct a grid of gain values by setting upper bounds on their admissible values so as to ensure the algorithm’s stability. We use a grid of 100 values for this purpose, meaning that our estimation routine is applied to each algorithm with 100 different constant gain values.

When it comes to represent agents learning-to-forecast behavior, though, an unique gain value is required for each time an iteration on the recursive algorithms is performed. Under the gain as a choice assumption we are interested in finding the gain value that would be representative of agents’ optimization of each algorithm’s forecasting accuracy. In other words, we can uncover what would have been an agent’s choice of a gain by minimizing the average (accuracy) loss associated to the gains available in our grid. Similarly, for the gain as a primitive assumption we can also pick the gain following such an average loss minimization, though here the forecasts losses need to be de-

\[ \gamma_t = \frac{\gamma}{t} \quad \text{and} \quad \mu_t = \frac{\mu}{t} \]

We have also computed the algorithms with a decreasing gain on the form of \( \gamma_t = \gamma/t \) and \( \mu_t = \mu/t \) to benchmark our results.
fined relative to their resemblance to the surveys.

It still remains to specify under which samples these losses are computed. We explore three alternatives on this aspect. The first, denoted as full-sample, is to pick the gain yielding the minimum average loss over the full sample of forecasts that we have computed. Clearly, under the gain as a choice, this selection sample violates the restrictions of a “fair” out-of-sample forecasting exercise that we are exploring in connection to the idea of real-time learning. Nevertheless, this alternative has been adopted in some of the previous calibration attempts in the literature (see, e.g., Milani, 2007, 2008, 2011). It does not present any conflict under the gain as primitive, where it would just imply that agents hold immutable beliefs about the system they are forecasting. This alternative has also found some applications in the previous literature (see, e.g., Orphanides and Williams, 2005a; Pfajfar and Santoro, 2010).  

The second selection sample we adopt, denoted as in-sample, involves splitting the whole sample of forecasts in two parts: the first part is used as an in-sample period on which the minimization of the average loss is applied in order to pick a gain for each algorithm; the second part is then used for the evaluation, keeping fixed the gain calibration. This alternative goes in line with traditional exercises of forecast evaluation and has also been used by Branch and Evans (2006); Weber (2010) for purposes similar to ours.

The last selection sample we explore is to allow the choice of the gain to be recursive, through a minimization of the average loss over a rolling window sample of forecasts. From the real-time learning and forecasting perspective, the sequence of gains selected using this recursive approach is the extreme opposite to that proposed in the first alternative; particularly, it does not allow the use of information on the future quality of the forecasts for the calibration of the algorithms in real-time. Moreover, following to our previous discussion and the results in chapter 1, there is no reason to restrict the gains to be fixed

\footnote{We note that in all these papers the calibration has been made within a structural model and solely focusing on the LS.}
throughout the whole forecasting sample.

To make our comparative between these different selection samples conformable we set the length of the rolling window equal to the amount of data set aside for the in-sample calibration. Namely, we will adopt an in-sample/window length of 60 observations (1966q1-1980q4)\(^5\). These selection samples determine what is left as our evaluation sample: from 1981q1 to 2010q4. Moreover, given our interest in multi-horizons forecasts, we also need to specify which horizon is used in the loss evaluations. Here we allow the gain selections to differ by forecasting horizon, hence evaluating the above selection criteria for each forecasting horizon.

The grid approach is restrictive in two respects. First, by construction, the available options of gain values are a discrete approximation of the continuum interval of admissible values. Hence, the optimality of the selected gains is conditioned to this discretization. Second, in spite of allowing time variation to the determination of the learning gains, the grid approach does not allow gain variations to happen within an ongoing estimation process. Instead, it solves the selection problem allowing for switches between different algorithm/gain estimation sequences\(^6\). Our next approach is not affected by these restrictions.

### 3.3.3 Adaptive gain

An alternative approach to the selection of the learning gains in real-time is to turn the gain calibration itself into an adaptive estimation problem. The idea is to plug each learning algorithm with an outer mechanism to adjust the gain in response to changes in estimates of the algorithm’s recent performance. Such an automatic tuning of the recursive algorithms was first suggested in

---

\(^5\)Recall the learning-to-forecasts exercises take the first 75 observations (1947q2-1965q4) of our sample for the initialization of the algorithms.

\(^6\)One can circumvent this restriction by adding a time dimension to the grid, but this alternative would substantially increase the computational cost.
Benveniste et al. (1990), and later analyzed by Kushner and Yang (1995) who presented evidence favoring this approach. In economics, a recent application has been presented in Kostyshyna (2012), evidencing some quantitative improvements to model hyperinflation episodes.

The derivation of the adaptive gain learning algorithms departs from the specification of an objective function, which in our context is related to the loss functions we associate with the two alternative roles given to the learning gains. Hence, the adaptive gain approach is also split in two, with the gain as a choice represented by a mechanism responding to the algorithm’s accuracy, whereas under the gain as a primitive the adaptation is driven by the algorithm’s resemblance. To be concise, we adopt the general notation of section 1.4.1 in the derivation that follows: \( z_t \) stands for the forecast’s target (either the actuals, \( y_t \), or the survey forecasts, \( s_t \)); \( x_t \) continues to represent a vector of regressors, i.e., an unity constant and lags of inflation and growth under our VAR specification, (1.32); and \( \hat{\theta}_t \) is a vector of estimates of our model’s parameters\(^7\). Moreover, we also assume a squared loss function.

Let the objective be to select the gain so as to minimize the expected loss of forecast (comparison) errors

\[
J_t = \frac{1}{2} E \left[ (z_t - x_t' \hat{\theta}_{t-1})^2 \right].
\] (3.1)

If the true properties of the data generating process (DGP) in (1.1) and (1.19) were known beforehand, one could ideally pick each algorithm’s gain so as to optimize the above criterion. In the lack of this information, as it is the case in the learning-to-forecast situation, one alternative is to superimpose an outer adaptive scheme for the purpose of automatic tuning the gain parameter.

The general idea is to use a recursive updating scheme that corrects the gain parameter in the direction opposite to a stochastic approximation of the

\(^7\)Without loss of generality, our presentation focus in a single equation.
loss function gradient\textsuperscript{8}. Deriving\textsuperscript{9} these gradients and plugging their stochastic approximation into each algorithm’s recursion we obtain the LS and SG algorithms with adaptive gains (LSA and SGA, respectively).

**Algorithm 3 (LSA).** Under the estimation context of (1.1), the LSA algorithm assumes the form of

\[
\hat{\theta}_t^{LSA} = \hat{\theta}_{t-1}^{LSA} + \gamma_t R_t^{-1} x_t \left( y_t - x_t' \hat{\theta}_{t-1}^{LSA} \right),
\]

(3.2)

\[
R_t = R_{t-1} + \gamma_t \left( x_t x_t' - R_{t-1} \right),
\]

(3.3)

\[
\gamma_t = \left[ \gamma_{t-1} + \alpha_{\gamma} \Psi_{t-1}^{LSA} \left( z_t - x_t' \hat{\theta}_{t-1}^{LSA} \right) \right] \frac{\overline{T}_{\max}}{\overline{T}_{\min}},
\]

(3.4)

\[
\Psi_t^{LSA} = \left( I - \gamma_t R_t^{-1} x_t x_t' \right) \Psi_{t-1}^{LSA} + \left( I - \gamma_t R_t^{-1} \hat{S}_t \right) R_t^{-1} x_t \left( y_t - x_t' \hat{\theta}_{t-1}^{LSA} \right),
\]

(3.5)

\[
\hat{S}_t = (1 - \gamma_t) \hat{S}_{t-1} + x_t x_t' - R_{t-1},
\]

(3.6)

where $\alpha_{\gamma} \gamma$ is an adaptation constant, $\Psi_t^{LSA}$ stands for an estimate of $\partial \hat{\theta}_t^{LSA} / \partial \gamma$, $\hat{S}_t$ stands for an estimate of $\partial R_t / \partial \gamma$, and $[\bullet]_{\overline{T}_{\min}}^{\overline{T}_{\max}}$ is a truncation operator setting $\gamma_t$ to $\overline{T}_{\min}$ if it falls below this value, or to $\overline{T}_{\max}$ if it rises above this value. The remaining components follow from the definition of the LS algorithm.

**Algorithm 4 (SGA).** Under the estimation context of (1.1), the SGA algorithm assumes the form of

\[
\hat{\theta}_t^{SGA} = \hat{\theta}_{t-1}^{SGA} + \mu_t x_t \left( y_t - x_t' \hat{\theta}_{t-1}^{SGA} \right),
\]

(3.7)

\[
\mu_t = \left[ \mu_{t-1} + \alpha_{\mu} x_t' \Psi_{t-1}^{SGA} \left( z_t - x_t' \hat{\theta}_{t-1}^{SGA} \right) \right] \frac{\overline{\mu}_{\max}}{\overline{\mu}_{\min}},
\]

(3.8)

\[
\Psi_t^{SGA} = \left( I - \mu_t x_t x_t' \right) \Psi_{t-1}^{SGA} + x_t \left( y_t - x_t' \hat{\theta}_{t-1}^{SGA} \right),
\]

(3.9)

where $\alpha_{\mu}$ is an adaptation constant, $\Psi_t^{SGA}$ stands for an estimate of $\partial \hat{\theta}_t^{SGA} / \partial \mu$, and $[\bullet]_{\overline{\mu}_{\min}}^{\overline{\mu}_{\max}}$ is a truncation operator setting $\mu_t$ to $\overline{\mu}_{\min}$ if it falls below this value, or to $\overline{\mu}_{\max}$ if it rises above this value. The remaining components follow from the definition

\textsuperscript{8}Recall that this stochastic approximation idea has already been employed in section 1.2.3.

\textsuperscript{9}See appendix A.7.
of the SG algorithm.

An intuition for this gain adaptation mechanism follows directly from its interpretation as a numerical optimization method, browsing along the error-performance curve, (3.1), in search for an optimal gain. Here $\Psi_t$ keeps track of the algorithm’s past estimation performance, accumulating its past gradients discounted according to each algorithm’s forgetting factors\(^\text{10}\). This synthetic measure of past performance is taken as a reference in the gain update equations, (3.4) and (3.8): when the latest gradient points towards the same (a different) direction as of $\Psi_t$, the adaptive mechanism interprets this as evidence of systematic (contradictory) mistakes; hence the gain is increased (decreased) to intensify (lessen) the algorithm’s response to last period error\(^\text{11}\).

Nevertheless, the above interpretation becomes less compelling under the gain as a primitive approach, where $z_t \equiv s_t$. In this case the gain adaptation mechanism has to bear with two distinct estimation objectives: the algorithm’s accuracy performance, and its resemblance to the surveys. Following the above interpretation, that means the adaptation mechanism has to browse along two error-performance curves at the same time. Whereas the latest gradient estimates is drawn from the resemblance to surveys objective, the past gradients synthesized in $\Psi_t$ refer to the algorithm’s accuracy performance. Hence, the gain is increased to intensify the algorithm’s response to last period error if its recent performance indicates systematic mistakes on both dimensions, and vice versa.

Computation of the LSA and SGA algorithms still requires specification of their adaptation constants, $\alpha_\gamma$ and $\alpha_\mu$. According to the analysis of Kushner and Yang (1995), stability requires this parameter to be small, so as to satisfy $0 < \alpha_\mu \leq \overline{T}_\min$ and $0 < \alpha_\mu \leq \overline{\mu}_\min$, together with an appropriate upper bound

\(^{10}\)Recall the definition of $\lambda$ from section 1.2.2. For the LS this is recognized in (3.5) noting that $E[x_i x'_i] \rightarrow R_i$.

\(^{11}\)A similar intuition is given in Kostyshyna (2012), though $\Psi_t$ is interpreted as the discounted past errors due to a simpler model specification where agents are learning only the value of a constant.
to the gains. For our purposes we take the extreme gain values in the grids defined in the previous section as the bounds for the adaptive gains. Most importantly, Kushner and Yang (1995) also present simulation evidence indicating that the algorithm’s performance is not as sensitive to the calibration of \( \alpha \) as it is to the learning gain. Therefore, our calibration of these adaptation constants will precede any attempt of performance optimization in favor of a calibration presenting a sequence of gains not too jumpy, but neither constant\(^{12}\).

We end this section with a word of caution in the use of the adaptive gains approach within a learning context. In the derivation of the gradient of (3.1) we assumed that the derivative of \( z_t \) with respect to the gain is null, which may not be a realistic assumption when \( z_t \equiv y_t \) (gain as a choice): self-referentiality implies that the determination of the actuals are affected by agents expectations; if agents are learning, the gain would have an indirect effect over \( y_t \) by determining agents expectations, hence violating\(^{13}\) the assumption above. However, relaxing this assumption requires the specification of a structural model, which goes beyond our scope.

A similar argument may be made when \( z_t \equiv s_t \) (gain as a primitive), remembering that we take the surveys as proxy for agents expectations. Under our hypothesis that the learning algorithms represent agents’ learning-to-forecast behavior, the gains would have a direct effect in the determination of \( s_t \). But for this case our assumption can be readily justified: under the gain as a primitive, its evolution is exogenously determined; our use of the adaptive mechanisms of gain adjustment is motivated solely as estimators that take the survey’s data as given.

\(^{12}\)We discuss these calibrations in the next section.
\(^{13}\)Agents would also be required to be aware of their expectation’s effects for this relationship to be manifest in their loss function.
3.4 Results and discussion

In this section we present the comparative results on the different approaches to the calibration of the learning gains as outlined above. We first present the numerical results of the calibrated gains. Our evaluation begins with an overview on the statistical properties of the forecasts associated to the combinations of learning algorithms and their calibrations. We then compare the different methods of gain selection, namely, the full-sample, the in-sample, and the recursive grid-based methods, and the adaptive gains approach. Finally, we shift our focus to our main issue on the determination of the learning gains, namely, we compare the gain as a choice and the gain as a primitive assumptions.

3.4.1 Gain calibrations

In our approach a sequence of gain values is associated to each application of the learning algorithms to forecast inflation and growth. These are determined according to the combinations of gain assumptions and selection methods. In Table 3.1 we present the resulting numerical calibrations we obtained for the fixed gain calibrations, namely, the full-sample and the in-sample methods.

We make three main observations regarding these fixed calibrations. First, it is clear that the data scales substantially pushed the gain calibrations for the SG algorithm downwards relative to those for the LS, even though there is no direct proportionality between them\(^{14}\). Second, the primitive gain calibrations tended to be smaller (bigger) than those as a choice for inflation (growth). Lastly, the gains displayed a decreasing trend in relation to the VAR lag orders, but no clear relationship with the forecasting horizons.

In contrast to these fixed calibrations, the recursive and adaptive methods introduce time-variation to the sequence of gains. Their evolution through time

---

\(^{14}\)Their relative ratio (LS/SG) vary from as least as 12x (4x) up to 120x (335x) for inflation (growth).
Table 3.1: Fixed gain calibrations.

<table>
<thead>
<tr>
<th>Variable Gains as a choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain as a primitive</td>
</tr>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Full-sample</td>
</tr>
<tr>
<td>In-sample</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gain as a primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Full-sample</td>
</tr>
<tr>
<td>In-sample</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(a) Inflation: Least Squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b) Growth: Least Squares</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stochastic Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Full-sample</td>
</tr>
<tr>
<td>In-sample</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(a) Inflation: Least Squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b) Growth: Least Squares</td>
</tr>
</tbody>
</table>


The full-sample calibrations report the gain values minimizing the average squared distance between the forecasts and (i) the actual observations for the gain as choice, or (ii) the survey forecasts for the gain as a primitive, over the entire sample of data available after initialization, i.e., from 1966q1 to 2010q4, whilst for the In-sample the gains are selected over an initial sample of 60 observations, from 1966q1 to 1980q4.
The recursive calibrations of the gains represent discrete values based on the grid of gains computed for each algorithm.

is presented in Figures 3.1 and 3.2, respectively. Overall, these figures point to an agreement between the choice and primitive time-varying calibrations. However, little harmony is found comparing the gains picked by the recursive and adaptive calibrations; one difference is observed in their scales, with the adaptive gains presenting a wider range of variation. Distinct behaviors are also observed on these calibrations with respect to the variable forecasted, and the algorithm. It remains to see whether these differences have been relevant for their performances.
The series of adaptive gains were obtained using adaptation constants given by $\alpha_{\gamma} = 0.01$ and $\alpha_{\gamma} = 0.001$ for the LS on inflation and growth, respectively, and $\alpha_{\mu} = 1 \times 10^{-6}$ and $\alpha_{\mu} = 1 \times 10^{-7}$ for the SG on inflation and growth, respectively.
3.4.2 Comparative overview

We analyze the quality of the forecasts produced by each combination of algorithm/calibration along the two evaluation dimensions of our learning-to-forecast exercises: their forecasting accuracy, and their resemblance to surveys. Although we take these two criteria as mutually desirable, it is not clear whether they are compatible with each other; recall that whilst forecasting accuracy represents the goal of optimizing agents, resemblance to surveys is indicative of actual agents behavior. In spite of this conflict, there is some overlap between these criteria and their homonym measures used in this chapter’s definitions of gain determination approaches. As such forecasting accuracy would tend to favor calibrations of the gain as a choice, whereas resemblance to surveys would have a similar bias towards the gain as a primitive. Therefore, their joint usage can be justified as an attempt to prevent favoring any of these approaches.

We start looking over the forecasts associated to each algorithm and gain value included in the grid computations. In Figure 3.3 we average the performance of each algorithm’s forecasts for the different grid gain values, also presenting the algorithms’ decreasing gain performances as a benchmark. The evidence is clearly favoring the constant gain SG algorithm, but not so remarkably for the LS case. This latter result appears to be at odds with that of Branch and Evans (2006), who found that the LS with constant gain tends to outperform its decreasing gain version.

An explanation for these results is due to our use of a different evaluation sample from that used by those authors, particularly for the inclusion of data covering the recent (2007-08) financial crisis. Intriguingly, the instabilities associated to this period seems to have more negatively affected the constant gain LS than its decreasing gain version\(^{15}\). Hence, the potential tracking bene-

\(^{15}\)Inspection of results restricted to the Branch and Evans (2006) sample corroborate this point. Another difference is that we use a real-time dataset both for the computation of forecasts and for their evaluation; comparatively, this was found to make a bigger difference for the results on growth.
The MSFE plotted are computed for each gain over the evaluation sample of forecasts, 1981q1-2010q4. The gain calibration used for the LS/SG algorithms are indicated into the lower/upper horizontal axes, respectively, and correspond to those experimentally calibrated for algorithm’s stability. Notice the minimums of these curves are not equivalent to the Full-sample gain calibrations presented in Table 3.1 due to the use of different sample periods. Similar figures corresponding to the Full-sample gain calibration can be found in Berardi and Galimberti (2012b, Fig. 2).
fits provided by the constant gain specifications seem to have been overcome by the higher level of noise affecting the economy during this period.\footnote{Recall that our previous chapter simulation results (see footnote 15) revealed that the estimation MSE tends to increase as the SNR decreases.}

It is also evident in this figure how the performance of each algorithm is affected as the forecasting horizon varies. For the SG there is mainly a scale effect, whereas the LS performance is found to be rather sensitive to the gain values at longer horizons. This result is particularly relevant for a recent debate on the distinction between Euler equation and infinite horizon learning (see Branch et al., 2012). As the analysis of Eusepi and Preston (2011) reveals, introducing learning directly into the Euler equations predicted by equilibrium and rational optimization assumptions reduces the empirical relevance of the “learning explanation” of macroeconomic fluctuations. Elusively, that ends up turning the agent’s infinite horizon decision problem into a one-period-ahead expectations formation problem. Our findings indicating the sensitivity of the algorithms’ performance to variations in the forecasting horizon thus contribute to this debate favoring the multi-horizons approaches to learning.

Finally, before getting into the comparative of the gain calibrations, we present in Table 3.2 statistics for each individual series of forecasts. These statistics point towards similar observations as those we drawn in section 1.5: (i) the LS (SG) forecasts tend to be biased up(down)wards; (ii) the forecasts fail to replicate growth rates variability, whereas for inflation the SG forecasts presented variances closer to that of the actuals; (iii) between each algorithm’s calibrations there is little variation in terms of their forecasts statistical properties.

The \textit{adaptive} gain specifications seem to be the only escaping to these regularities. Forecasting growth, e.g., their variance were remarkably higher (lower) than the other calibrations for the LS (SG) algorithm. The \textit{adaptive} gains’ performances, here assessed through the statistics in the last four columns, also stand out relative to the others, though negatively. These results, however, are
Table 3.2: Data and forecasts statistics by calibration.

(a) Inflation.

<table>
<thead>
<tr>
<th>Series/Algorithm</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
<th>Var</th>
<th>AR(1)</th>
<th>CorA</th>
<th>CorS</th>
<th>MSFE</th>
<th>MSFCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuals</td>
<td>2.67</td>
<td>-0.33</td>
<td>9.39</td>
<td>2.68</td>
<td>0.63</td>
<td>1.00</td>
<td>0.81</td>
<td>0.00</td>
<td>1.03</td>
</tr>
<tr>
<td>Surveys</td>
<td>2.93</td>
<td>0.62</td>
<td>9.49</td>
<td>2.46</td>
<td>0.94</td>
<td>0.81</td>
<td>1.00</td>
<td>1.03</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Least Squares
- Full-smpl. /choice | 2.99 | -0.14| 10.30| 2.94 | 0.83  | 0.73 | 0.90 | 1.62 | 0.55  |
- In-smpl. /choice | 2.98 | -0.30| 10.35| 2.92 | 0.81  | 0.72 | 0.90 | 1.64 | 0.55  |
- Recur. /choice | 3.07 | 0.13 | 10.35| 2.80 | 0.81  | 0.72 | 0.90 | 1.69 | 0.56  |
- Adapt. /choice | 2.99 | 0.71 | 10.80| 3.00 | 0.85  | 0.71 | 0.88 | 1.86 | 0.74  |
- Full-smpl./primit. | 2.97 | -0.34| 10.32| 2.75 | 0.74  | 0.70 | 0.90 | 1.71 | 0.54  |
- In-smpl. /primit. | 2.97 | -0.40| 10.39| 2.83 | 0.76  | 0.71 | 0.90 | 1.70 | 0.55  |
- Recur. /primit. | 3.06 | 0.01 | 10.39| 2.57 | 0.78  | 0.72 | 0.89 | 1.62 | 0.55  |
- Adapt. /primit. | 3.01 | 0.52 | 10.28| 3.37 | 0.88  | 0.70 | 0.86 | 1.93 | 0.87  |

Stochastic Gradient
- Full-smpl. /choice | 2.77 | 0.12 | 11.89| 2.65 | 0.69  | 0.67 | 0.88 | 1.76 | 0.62  |
- In-smpl. /choice | 2.77 | 0.12 | 11.89| 2.65 | 0.69  | 0.67 | 0.88 | 1.76 | 0.62  |
- Recur. /choice | 2.69 | -0.83| 11.89| 2.79 | 0.74  | 0.69 | 0.89 | 1.69 | 0.63  |
- Adapt. /choice | 2.98 | 0.23 | 10.89| 2.79 | 0.76  | 0.69 | 0.89 | 1.76 | 0.58  |
- Full-smpl./primit. | 2.77 | 0.12 | 11.89| 2.65 | 0.69  | 0.67 | 0.88 | 1.76 | 0.62  |
- In-smpl. /primit. | 2.77 | 0.11 | 11.84| 2.62 | 0.68  | 0.67 | 0.88 | 1.77 | 0.62  |
- Recur. /primit. | 2.78 | -0.08| 11.84| 2.62 | 0.68  | 0.67 | 0.88 | 1.76 | 0.63  |
- Adapt. /primit. | 2.97 | 0.19 | 11.53| 2.79 | 0.74  | 0.69 | 0.89 | 1.79 | 0.56  |

(b) Growth.

<table>
<thead>
<tr>
<th>Series/Algorithm</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
<th>Var</th>
<th>AR(1)</th>
<th>CorA</th>
<th>CorS</th>
<th>MSFE</th>
<th>MSFCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuals</td>
<td>2.54</td>
<td>-6.14</td>
<td>8.67</td>
<td>6.17</td>
<td>0.50</td>
<td>1.00</td>
<td>0.78</td>
<td>0.00</td>
<td>2.56</td>
</tr>
<tr>
<td>Surveys</td>
<td>2.24</td>
<td>-5.19</td>
<td>7.01</td>
<td>3.14</td>
<td>0.72</td>
<td>0.78</td>
<td>1.00</td>
<td>2.56</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Least Squares
- Full-smpl. /choice | 3.04 | -3.66| 6.58 | 1.58 | 0.60  | 0.39 | 0.58 | 5.55 | 2.74  |
- In-smpl. /choice | 3.00 | -4.15| 6.87 | 1.90 | 0.59  | 0.37 | 0.57 | 5.72 | 2.85  |
- Recur. /choice | 3.08 | -0.95| 6.70 | 1.18 | 0.53  | 0.38 | 0.63 | 5.55 | 2.61  |
- Adapt. /choice | 2.80 | -6.79| 9.27 | 3.68 | 0.40  | 0.21 | 0.35 | 7.84 | 4.71  |
- Full-smpl./primit. | 3.01 | -2.02| 5.46 | 1.07 | 0.58  | 0.47 | 0.64 | 5.03 | 2.46  |
- In-smpl. /primit. | 3.04 | -3.66| 6.58 | 1.58 | 0.60  | 0.39 | 0.58 | 5.55 | 2.74  |
- Recur. /primit. | 2.98 | -0.95| 6.00 | 1.10 | 0.52  | 0.42 | 0.66 | 5.24 | 2.34  |
- Adapt. /primit. | 2.81 | -5.48| 7.43 | 3.47 | 0.57  | 0.30 | 0.50 | 6.90 | 3.60  |

Stochastic Gradient
- Full-smpl. /choice | 1.88 | -2.13| 4.66 | 1.32 | 0.48  | 0.43 | 0.59 | 5.44 | 2.16  |
- In-smpl. /choice | 1.82 | -2.10| 4.57 | 1.24 | 0.47  | 0.43 | 0.59 | 5.52 | 2.20  |
- Recur. /choice | 2.10 | -1.24| 4.75 | 1.32 | 0.53  | 0.40 | 0.55 | 5.38 | 2.23  |
- Adapt. /choice | 2.13 | -0.54| 4.74 | 0.90 | 0.68  | 0.14 | 0.30 | 6.51 | 3.02  |
- Full-smpl./primit. | 1.88 | -2.13| 4.66 | 1.32 | 0.48  | 0.43 | 0.59 | 5.44 | 2.16  |
- In-smpl. /primit. | 1.86 | -2.13| 4.64 | 1.29 | 0.48  | 0.43 | 0.59 | 5.47 | 2.17  |
- Recur. /primit. | 2.07 | -1.85| 4.68 | 1.46 | 0.53  | 0.41 | 0.56 | 5.37 | 2.20  |
- Adapt. /primit. | 2.09 | -0.56| 4.35 | 0.77 | 0.64  | 0.20 | 0.38 | 6.23 | 2.73  |

See footnotes to Table 1.1. The algorithms’ forecasts refer to those from the VAR(1).
representative of only the first horizon of forecasts obtained with the VAR(1) model specification. To achieve a more broad assessment of how the different gain calibrations compare, we introduce a key synthetic evaluation statistic in what follows.

3.4.3 Gain selection

Recall that in the evaluation stage of the learning-to-forecast exercises we proposed a relative assessment of the learning mechanisms by conducting paired comparisons of their forecasting performance. However, our coverage of multiple forecasting horizons and VAR lag order specifications, for robustness, requires performing a high quantity of such comparisons. Hence, for the ease of exposition we now adopt hit rate measures to synthesize these evaluations. The hit rate stands for the frequency by which the forecasts associated to a given algorithm/calibration is found to outperform those associate to its competitor(s) with respect to one of our evaluation criteria.

To illustrate the above definition consider the hit rates presented in Table 3.3 comparing the different methods of gain selection. The first comparison focus on the LS algorithm applied in forecasting inflation with gains determined as a choice; here the hit rate of 45% associated to the Full-sample selection method, e.g., indicates that this method outperformed the others in 9 out of the 20 comparisons conducted for each combination of VAR lag order and forecasting horizon. The DM-20% and GW-20% statistics associated to these hit rates then represent the frequency by which the losses associated to the outperforming method is found to be statistically different of the second-best method according to the corresponding test at a 20% level of significance. So, in our example case of the LS/choice calibrations we have that only 3 (15% of

---

17To be specific, 40 for each pair of algorithms/calibrations: 5 horizons × 4 VARs × 2 evaluation criteria. See Table C.5 in the appendix for an illustrative excerpt from our comparative evaluation routines.

18Draws are handled by counting for both tied competitors. These may occur when, e.g., different methods of gain selection pick the same gain.
of the 9 cases where the Full-sample was favored presented a GW test $p$-value below 20% when compared to the second-best method of gain selection.

The comparative evaluations of the different methods of gain selection are presented in Table 3.3 according to the variable forecasted, the algorithm, and the gain determination approach. There is notable sensitivity on the results presented along these dimensions, though the combinations of variable/algorithm seem to be a main factor: in forecasting inflation, the recursive method is favored both in terms of accuracy as in terms of resemblance for the LS, whilst a similar result is found with the full-sample method for the SG algorithm; in forecasting growth these patterns change place, with the full-sample gain specification being favored for the LS, whilst under the SG the results favor time-varying gain specifications. Thus, where the evidence is favorable to a fixed gain calibration for one algorithm, it is favorable for time-varying gains for the other\textsuperscript{19}.

In spite of some evidence favoring fixed gains, the in-sample method has found scarce evidence in its favor. This indicates the relevance of the information contained in the post in-sample observations (after 1980q4) for the calibration of the algorithms, which is not surprising given the evidence of structural breaks around this period (see, e.g., Stock and Watson, 2003). Statistical significance was also scarce in most of the comparisons presented, with some exceptions under the resemblance criterion, e.g., the LS recursive wins in forecasting inflation. This result does not come at surprise neither; our previous inspection of the forecasts statistics already indicated these calibrations presented similar properties, which probably hindered the statistical tests’ potential to distinguish between these series.

Regarding the time-varying gain specifications the most remarkable result

\textsuperscript{19}One might have erroneously expected that the recursive method, for its flexibility, should have picked the same gain as the full-sample, giving the superiority of this latter; notice, however, that the recursive method does not have access to the same information set as that given to the full-sample method.
Table 3.3: Hit rates comparing gain selection methods.

(a) Forecast accuracy.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Full-sample wins</th>
<th>In-sample wins</th>
<th>Recursive wins</th>
<th>Adaptive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Alg./Gain</td>
<td>Hit rate</td>
<td>DM-20%</td>
<td>GW-20%</td>
<td>Hit rate</td>
</tr>
<tr>
<td>Inflation - LS/choice</td>
<td>45%</td>
<td>0%</td>
<td>15%</td>
<td>5%</td>
</tr>
<tr>
<td>- LS/primitive</td>
<td>22.5%</td>
<td>0%</td>
<td>0%</td>
<td>7.5%</td>
</tr>
<tr>
<td>- SG/choice</td>
<td>45%</td>
<td>0%</td>
<td>0%</td>
<td>20%</td>
</tr>
<tr>
<td>- SG/primitive</td>
<td>55%</td>
<td>0%</td>
<td>10%</td>
<td>15%</td>
</tr>
<tr>
<td>Growth - LS/choice</td>
<td>70%</td>
<td>20%</td>
<td>10%</td>
<td>20%</td>
</tr>
<tr>
<td>- LS/primitive</td>
<td>47.5%</td>
<td>5%</td>
<td>5%</td>
<td>7.5%</td>
</tr>
<tr>
<td>- SG/choice</td>
<td>40%</td>
<td>10%</td>
<td>15%</td>
<td>10%</td>
</tr>
<tr>
<td>- SG/primitive</td>
<td>10%</td>
<td>5%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

(b) Forecast resemblance.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Full-sample wins</th>
<th>In-sample wins</th>
<th>Recursive wins</th>
<th>Adaptive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Alg./Gain</td>
<td>Hit rate</td>
<td>DM-20%</td>
<td>GW-20%</td>
<td>Hit rate</td>
</tr>
<tr>
<td>Inflation - LS/choice</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>10%</td>
</tr>
<tr>
<td>- LS/primitive</td>
<td>15%</td>
<td>0%</td>
<td>0%</td>
<td>5%</td>
</tr>
<tr>
<td>- SG/choice</td>
<td>47.5%</td>
<td>15%</td>
<td>20%</td>
<td>22.5%</td>
</tr>
<tr>
<td>- SG/primitive</td>
<td>45%</td>
<td>25%</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>Growth - LS/choice</td>
<td>70%</td>
<td>15%</td>
<td>20%</td>
<td>20%</td>
</tr>
<tr>
<td>- LS/primitive</td>
<td>50%</td>
<td>0%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>- SG/choice</td>
<td>30%</td>
<td>10%</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/primitive</td>
<td>15%</td>
<td>5%</td>
<td>10%</td>
<td>0%</td>
</tr>
</tbody>
</table>

The hit rates sum up forecasting horizons (5) and VAR lag orders (4). DM-20% and GW-20% represents the frequency within which the associated specification was found to have superior performance over the runner up specification with statistical significance below the 20% level using the Diebold and Mariano (1995) and the Giacomini and White (2006) test(s).
relates to the poorer performance of the adaptive method under the LS algorithm. An explanation for this result may be drawn looking back into how the recursive selections evolved compared to the adaptive gains, in the top panels of figures 3.1 and 3.2, respectively; clearly the adaptive gain values are generally higher than those picked by the recursive method. Hence, our results indicate that the upper bound for the LS adaptive gains may have been too loose, which is also consistent with the observation by Kushner and Yang (1995) on the relevance of this parameter for the algorithm’s performance. Clearly, this was not a problem for the SG algorithm.

### 3.4.4 Gain determination and implications

We now get to the issue on the determination of the learning gains. The hit rates comparing the gain as a choice and the gain as a primitive assumptions are presented in Table 3.4. The calculations adopted to obtain these hit rates follow in similar lines as those for Table 3.3, as explained above. The only difference is that here there are only two “competitors”. Take the case of the LS applied to forecast inflation using the Full-sample gain selection method: the 90% hit rate associated to the gain determined as a primitive indicates that this assumption outperformed the alternative determination of the gain as a choice in 18 out of the 20 comparisons combining the VAR lag orders and forecasting horizons we consider. Furthermore, we can see that in 5 (9) of these victories the DM (GW) test indicated that the series of losses associated the two methods of gain determination are statistically different at a significance level below 20%.

In contrast to our previous results, here the evidence is crystal clear: the gain as a primitive is overwhelmingly favored on both evaluation criteria. The few threats to the primitive gain dominance can be easily disqualified: (i) the LS with adaptive gains, we recall, was found to have a poor performance relative to its alternatives; (ii) under the other few calibrations where the gain as choice prevailed, i.e., the LS fixed gains for growth, statistical significance is
Table 3.4: Hit rates comparing gain determination assumptions.

(a) Forecast accuracy.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Gain as a choice wins</th>
<th>Gain as a primitive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>Inflation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- LS/Full-sample</td>
<td>10%</td>
<td>0%</td>
</tr>
<tr>
<td>- LS/In-sample</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>- LS/Recursive</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>- LS/Adaptive</td>
<td>40%</td>
<td>5%</td>
</tr>
<tr>
<td>- SG/Full-sample</td>
<td>17.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/In-sample</td>
<td>22.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/Recursive</td>
<td>40%</td>
<td>5%</td>
</tr>
<tr>
<td>- SG/Adaptive</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>Growth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- LS/Full-sample</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>- LS/In-sample</td>
<td>55%</td>
<td>10%</td>
</tr>
<tr>
<td>- LS/Recursive</td>
<td>15%</td>
<td>5%</td>
</tr>
<tr>
<td>- LS/Adaptive</td>
<td>75%</td>
<td>30%</td>
</tr>
<tr>
<td>- SG/Full-sample</td>
<td>27.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/In-sample</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/Recursive</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/Adaptive</td>
<td>20%</td>
<td>0%</td>
</tr>
</tbody>
</table>

(b) Forecast resemblance.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Gain as a choice wins</th>
<th>Gain as a primitive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>Inflation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- LS/Full-sample</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>- LS/In-sample</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>- LS/Recursive</td>
<td>40%</td>
<td>25%</td>
</tr>
<tr>
<td>- LS/Adaptive</td>
<td>35%</td>
<td>10%</td>
</tr>
<tr>
<td>- SG/Full-sample</td>
<td>17.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/In-sample</td>
<td>22.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/Recursive</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/Adaptive</td>
<td>20%</td>
<td>0%</td>
</tr>
<tr>
<td>Growth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- LS/Full-sample</td>
<td>45%</td>
<td>5%</td>
</tr>
<tr>
<td>- LS/In-sample</td>
<td>55%</td>
<td>25%</td>
</tr>
<tr>
<td>- LS/Recursive</td>
<td>10%</td>
<td>5%</td>
</tr>
<tr>
<td>- LS/Adaptive</td>
<td>65%</td>
<td>10%</td>
</tr>
<tr>
<td>- SG/Full-sample</td>
<td>2.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/In-sample</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/Recursive</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>- SG/Adaptive</td>
<td>5%</td>
<td>0%</td>
</tr>
</tbody>
</table>

See footnotes to Table 3.3.
still favoring the \textit{primitive} gain.

From an applied perspective, our findings provide some important guidance on how the learning gain can be appropriately calibrated. The gain as a \textit{primitive} is associated to a minimization of the distance between the algorithm’s forecasts and their survey’s counterparts. To select a particular sequence of gains on the basis of this measure, one needs to take into account the algorithm and variable of interest. For inflation, flexibility in the time-variations of the gain is required by the LS algorithm; the SG, on the other hand, is more appropriately calibrated with a fixed gain value. For growth, these specifications change place: the SG is the one requiring time-varying gains, whilst a fixed gain does the job with the LS.

Finally, our evidence favoring the interpretation of the learning gains as \textit{primitive} parameters of agents learning-to-forecast behavior also has important implications within the context of the theoretical literature, as we outlined in section 3.2. Our results contribute to the debate on the internal (in)consistency between the adaptive learning approach and the rationality of mechanisms of expectations formation. Namely, by favoring the gain as a \textit{primitive}, we provide support to the view that agents are essentially characterized as bounded rational; when their expectations formation process is represented by adaptive learning algorithms, even if this process is likely to converge to a RE equilibrium, it does not imply that these agents will rationally optimize on their calibration. Agents simply do not seem to take it as a \textit{choice}.

3.5 Concluding remarks

In this chapter we have studied empirically the issue of what is the most appropriate interpretation of the learning gains in adaptive algorithms, and how can a calibration strategy be developed so as to reflect it. Our main insight relates to a distinction on the rationale given to the determination of the learning gains: as a \textit{choice} of rationally optimizing agents, or as a \textit{primitive} parameter of
bounded rational agents.

We have also produced some renewed numerical calibrations of the learning gains for applications on real-time US data of inflation and output growth. Consistent to our analysis, our gain calibrations are segmented by the different methods and assumptions in the determination of these gains, and according to the lag orders of VAR models and the forecasting horizons. Significant heterogeneity was found with respect to these dimensions as well.

Our results provide strong evidence in favor of the interpretation of the learning gains as primitive parameters of agents learning-to-forecast behavior. Furthermore, our evidence also points to some heterogeneity in the time evolution of this behavior with respect to the variable forecasted and the algorithm adopted. This latter finding clearly provides further motivation for our analysis of the algorithm’s dimension in the next chapter.

Therefore, the main implication of our results is that agents’ adaptive behavior in the adjustment of their learning algorithms is better represented from a bounded rationality point of view rather than from a rationally optimizing interpretation. This finding turns out to be relevant to a long established debate on the internal (in)consistency between the adaptive learning approach and the rationality of expectations. Namely, there is no reason to expect, or question, a rational agent to optimize on the choice of a learning gain, given that our evidence suggest that it stands as a primitive parameter of agents bounded rationality.
Chapter 4

The choice of a representative learning algorithm

4.1 Introduction

The learning approach to expectations modeling in macroeconomics requires the specification of a learning algorithm to represent how agents update their beliefs about the economy. Throughout this thesis we have conducted our analysis with a parallel evaluation of the two main algorithms in the literature, namely the LS and the SG; the existence of these alternatives, though, adds to the researcher uncertainty on how to appropriately represent agents’ learning-to-forecast behavior. The usual choice of a representative algorithm in the literature has been the LS, possibly due to its widespread popularity between econometricians. But apart from this vogue there seems to be no clear justification for such choice and, taking a bounded rationality standpoint, one could for example argue in favor of the computationally simpler alternative provided by the SG algorithm.

Since the ultimate sieve of scientific research comes with empirical validation, we argue that the previous literature has neglected the need of a realistic justification in the choice of a learning algorithm. In this chapter we attempt to remedy this issue by empirically assessing the representativeness of these learning algorithms, also contributing to the literature with the formulation of new “paired” algorithms. Other than evaluating the choice between these different learning mechanisms, we also return to our main issue regarding the plausibility of learning. Namely, we evaluate whether our efforts to tune the

\footnote{Some results in this chapter have been previously presented in Berardi and Galimberti (2012b).}
learning mechanism for applied macroeconomics is successful in reverting our earlier unfavorable account, in section 1.5.

Our approach is based on learning-to-forecast exercises, simulating the real-time process of expectation formation for US data on inflation and output growth. In the choice of an algorithm for the purpose of representing agents’ learning behavior, two alternative views are possible: one that would select the algorithm that performs better, therefore emphasizing the rational aspect of agents’ choice, and the other that would instead select the algorithm that better resembles actual forecasts, in an attempt to mimic agents’ actual behavior. Consistent to our previous approach, we associate different evaluation measures to each of these views: forecasting accuracy and resemblance to surveys, respectively. We recall that it is not clear whether these two alternative approaches are compatible with each other, but we take both criteria as mutually desirable properties of a representative learning mechanism.

We obtain two sets of results. Our first results compare the LS and the SG algorithms and provide evidence favorable for the use of the former. Although this finding goes in support of most of the previous applied literature adopting the LS as a representative learning mechanism, our results also point to an interesting qualification in connection to last chapter findings. Namely, the LS dominance is weakened under calibrations of the learning gain as a choice, where the SG becomes the preferred for forecasting inflation. Most intriguingly, with the gain determined as a choice the LS is still found to provide forecasts closer to those observed from surveys for both inflation and growth variables. The puzzling aspect of this result is due to the SG superiority in forecasting inflation: while a rational choice of an algorithm would depend on the variable of interest, agents are found to behave as LS learners irrespective of the variable forecasted. Therefore, our results point to another drawback of the approach to learning assuming that agents actively decide on how to best calibrate their learning mechanisms.
Focusing on the LS as the representative algorithm we then reassess the plausibility of the learning approach by evaluating the quality of the forecasts associated to the algorithm relative to those obtained through the surveys. Recall that these survey forecasts are taken as proxies to agents’ expectations, thus providing a benchmark to judge the plausibility of the learning algorithms as representative of these agents’ learning-to-forecast behavior. Our results in this exercise indicate substantial improvements to the comparatives carried in section 1.5. Particularly, the learning algorithm is able to equilibrate the contest against the surveys if we neglect the first horizon of forecasting for inflation, and the first two for growth. Therefore, our results suggest that our tuning efforts were not in vain.

Our second set of results adds to the debate on the choice of a representative learning algorithm by proposing two alternative formulations, which we call the Hybrid and the Mixing algorithms. Both these algorithms stand as “paired” formulations of the previous algorithms, with a weight parameter regulating how close each of them behaves as the LS or as the SG. We compare these paired algorithms both between each other, and against the traditional LS and SG algorithms, again in terms of our two forecast evaluation criteria. Our results indicate that the paired algorithms successfully improve upon the performance of the traditional algorithms, though these improvements are observed mainly under fixed gain and weight calibrations. Most importantly, with the paired algorithms we can further improve our assessment on the plausibility of the learning approach: these algorithms are found to balance the contest against the surveys just after the first forecasting horizon for both inflation and growth variables.

The remainder of this chapter proceeds as follows. In section 4.2 we preceude our empirical analysis relating the issue of the choice of a representative algorithm to the relevant literature, and also review the previous works assessing the learning algorithms comparatively from a theoretical perspective. We
then proceed with our empirical analysis comparing the LS and the SG algorithms, in section 4.3. In section 4.4 we shift our focus to the presentation and evaluation of the paired algorithms. Finally, we conclude this chapter with some remarks in section 4.5.

4.2 Literature background

In spite of our applied focus, the choice of a representative learning algorithm is relevant for both the theoretical and the empirical literature on learning and expectations in macroeconomics. We begin this section with a brief review of the relevant theoretical issues, and then focus on our main concerns regarding the recent empirical literature on adaptive learning.

4.2.1 Theoretical relevance

One of the main motivations for the analysis of learning convergence has been to provide a solution to cases of RE indeterminacy (see, e.g., Evans and Honkapohja, 1994). One important tool on this issue is provided by the so-called E-stability\(^2\) conditions: drawing upon a tight link between E-stability and LS-learnability, known as the E-stability principle, the learning explanation has been adopted as the foundation for an equilibrium selection criterion. Taking advantage of this principle, the literature on equilibrium selection has been drawing correspondences between RE determinacy and E-stability.

One example is provided by McCallum (2007), which showed that for a broad class of multivariate linear models, RE determinacy is a sufficient (though not a necessary) condition for E-stability. Moreover, under RE indeterminacy McCallum argues that there is usually only one solution that is E-stable, thus indicating that the E-stability criterion may single out an unique plausible solution. However, Bullard and Eusepi (2009) pointed out that McCallum’s

\(^2\)A RE equilibrium is said to be E-stable if small displacements of agents beliefs from the equilibrium end up returning back to it (Evans, 1989).
equivalency connections were not realistic for being based on the assumption that agents have access to contemporaneous information. Ellison and Pearlman (2011) closed the dispute by showing that the correspondence between determinacy and E-stability is valid for any informational assumption as long as agents are assumed to hold the appropriate PLM\(^3\).

The main problem with this approach is in its reliance on the link between E-stability and LS-learnability, as it is now understood that the learnability of a given RE equilibrium is defined with respect to the algorithm of choice. Analysis of learning using the SG algorithm, for instance, pointed to convergence conditions distinct to those obtained under LS learning (Barucci and Landi, 1997). It is possible to find cases where the SG converges to a RE equilibrium that the LS does not (Heinemann, 2000), but also where the LS is stable but the SG is not (Giannitsarou, 2005). An assessment of the plausibility of these different learning algorithms is therefore of central relevance to the theoretical debate over learnability as a solution to RE indeterminacy.

### 4.2.2 Empirical relevance

The key feature in empirical works of adaptive learning is the replacement of assumptions implying an instantaneous adjustment of agents’ expectations with a characterization of agents as adaptive learners. Clearly, this approach establishes a new channel through which the effects of expectation shocks dissipate with lags in the dynamics of the economy. Two main strands emerged in this literature: one giving emphasis to the role of learning in explaining macroeconomic fluctuations; and the other focusing on the explanatory power of learning in modeling the behavior of policymakers.

The first studies adopting the learning approach were focused on applications within the subject of finance. Timmermann (1993) for example uses the

---

\(^3\)Ellison and Pearlman (2011) call this “saddlepath learning”, the main requirement being that agents can distinguish between predetermined and non-predetermined endogenous variables.
learning dynamics to explain anomalies encountered in the literature of asset pricing, such as the apparent predictability and the excess volatility found in stock returns data. In Garratt and Hall (1997) agents’ expectations of exchange rates are modeled as a learning process within a large scale macroeconomic model. Their analysis then focus on the implications of different kinds of structural change for the dynamics of learning, such as oil price shocks or exchange rate policy regime changes.

In the context of macroeconomics, the literature may yet be distinguished with respect to the canonical model adopted as a point of departure for the introduction of adaptive learning. Within the standard Real Business Cycle (RBC) class of models, Williams (2003) is usually referred as the first attempt to quantify the effects of learning over the dynamical properties of the model. His numerical results, however, brought a pessimistic account for the adaptive learning approach, pointing to an unlikely relevance of the learning channel to the explanatory power of those models. Nevertheless, qualifications to this conclusion were later presented by Huang et al. (2009) and Eusepi and Preston (2011); whilst in the former it was shown that learning has important amplifying effects to the technological shocks so prominent to the canonical model, it is in the later that the irrelevance finding of Williams (2003) was ultimately clarified.

According to Eusepi and Preston (2011), the difference relies on the failure to model optimal decisions conditional to preexisting beliefs. In other words, for infinite horizon decision problems it is important to consider the whole chain of extrapolative effects of agents expectations. Until recently such effects have been neglected in the literature proposing the introduction of adaptive learning at the level of the recursive solutions to the infinite horizons problem, hence stressing the relevance of expectations only for the one-period ahead forecasting horizon\textsuperscript{4}.

\textsuperscript{4}Recall our discussion about the Euler equation versus the infinite horizon learning approach in the previous chapter.
Within the tradition of new Keynesian models, and with the advent of the so-called Dynamic Stochastic General Equilibrium (DSGE) models, the incorporation of adaptive learning was even more challenging. The difficulties are due to the greater diversity of imperfect competition and market failure aspects already at play in these models in order to close the gap between the model and real market conditions, and to improve its fit to macroeconomic data (see, e.g., Christiano et al., 2005).

Milani (2007) provides some first estimates of a DSGE model with adaptive learning together with the possibility of consumption habits and price indexation; using US data his estimates suggest that the learning inertial effects prevail as the data’s preferred channel in explaining observed persistence in macroeconomic variables. Leaving the other sources of rigidities aside, Milani (2011) then shows that expectation shocks account for about half of business cycles fluctuations, the other half being attributed to traditional effects of shocks to aggregate demand.

The second strand of applied studies employs adaptive learning to improve our understanding of how policymakers behave. Here the central hypothesis is that policy can be represented by an agent’s pursuit of pre-established targets, though this agent lacks perfect knowledge of the economy’s full structure and its current state. Hence, the policymaker uses statistical methods to learn adaptively about structural changes taking place in the economy, at the same time that it is affecting the determination of this economy’s outcomes. The seminal contribution to this approach comes from Sargent (1999), where the concept of self-confirming equilibria was also introduced to represent the possible outcomes of purposeful interactions among a collection of adaptive learners.

Most studies following this approach of modeling policymakers as adaptive learners carry out counterfactual analysis focused on periods of recognized (in hindsight) structural change. Marcet and Nicolini (2003), for exam-
ple, emphasize the role of adaptive learning in explaining the recurrence of hyperinflation episodes during the 1980s, mainly observed in Latin American countries. Milani (2008) reassess the influential results of Clarida et al. (2000) about the destabilizing effects of pre-1979 US monetary policy, finding that the policy instability result does not hold when the RE hypothesis is relaxed to allow for an adaptive learning policymaker.

Another collection of studies searches for an explanation to the period of Great Inflation (1970s) in the US, based on the hypothesis that policymakers misperceived ongoing structural changes in the economy of that period: in Bullard and Eusepi (2005) the misperception is about a productivity slowdown, whereas Orphanides and Williams (2005a) emphasize the consequences of real-time difficulties to detect changes in the natural rate of unemployment. With the central bank represented as an adaptive learner, these studies provide an explanation for the slowness within which the monetary authority reacted to the ongoing structural change, thus prolonging its harmful effects.

Finally, our work is also linked to the literature investigating how different learning rules perform in forecasting macroeconomic data and in matching forecasts from surveys. Such literature includes the works of: Branch and Evans (2006), with an application to US data on inflation and growth, recently extended to other variables by Markiewicz and Pick (2013); Weber (2010), which provided a similar application to European data; and Ormeño and Molnár (2013) showing the usefulness of information contained in survey data for the estimation of structural models with adaptive learning. Although adopting an approach similar to ours, these works did not provide a comparison between the LS and SG algorithms, but mainly focused on the LS and its different calibrations.

Common to most of these studies is the use of the LS algorithm as representative of agents learning-to-forecast behavior. Thus, by inquiring about

\[^{5}\text{There were some few exceptions, as in Marcet and Nicolini (2003) and Bullard and Eusepi (2005).}\]
the choice of this representative algorithm our study attempts to provide robustness to the findings in the previous applied literature.

4.2.3 Theoretical comparison of the algorithms

In what follows we briefly review the statistical literature comparing the LS and the SG algorithms from an \textit{a priori} perspective, i.e., in theory. We focus on the main figures of merit considered in this literature. As the previous results in this literature reveal, there is no clear dominance of any of these algorithms in most of the evaluation measures, with data conditions being the ultimate determinant of their relative performances.

\textbf{Computational complexity}

The SG algorithm requires a lower number of computations for a complete iteration of adaptation than the LS, given that this latter requires the inversion of the matrix of moments, an operation for which computational complexity grows exponentially with the number of regressors. To be specific, while an SG iteration requires only $2K + 1$ multiplications and $2K$ additions, a LS iteration requires $K^2 + 5K + 1$ multiplications, $K^2 + 3K$ additions, and 1 division, with $K$ standing for the number of regressors in $x_t$ (see Sayed, 2008, pps. 166, 200-201).

\textbf{Rate of convergence}

There are many factors affecting the rate of convergence of the LS and the SG algorithms, such as the level of noise and the eigenvalues of the regressors covariance matrix (data-driven factors); the magnitude and the direction of the initial misalignment in the coefficients estimates that first ignited the transient phase (misalignment factors); and, the magnitude of the learning gain for each algorithm (calibration factors).

Eweda (1999) provides an analysis of the effects that these factors have over the algorithms’ convergence time by focusing on simplified cases, i.e., by im-
posing restrictive assumptions on the data environment. One important factor is the learning gain calibration, which is found to determine the speed with which the algorithm estimates are adjusted in face of large misalignments: the higher is the learning gain, the quicker is the adjustment. A second observation from Eweda’s results is that there might be cases when the SG algorithm converges faster than the LS algorithm. Specifically, under the case of uncorrelated regressors the SG transient performance tends to improve with increased regressors variance. Such a result is, however, dependent on the assumption of uncorrelated regressors, which is very restrictive for macroeconomic contexts. Under the case of correlated regressors, simulation evidence tends to favor the LS due to its data orthogonalization feature (see also Haykin, 2001, pps. 285-291, 367-371, 454-457).

Tracking performance

The assessment of an algorithm’s tracking performance focuses on its steady state behavior, i.e., after passing the transient phase. The main measures adopted for such an assessment are the MSD and the MSE, previously defined in (2.1) and (2.2), respectively. Recall that whereas the MSD captures the (average) accuracy of the algorithm’s estimates, the MSE provides a measure of estimation efficiency. Moreover, the evolution through time of these measures provides useful pictures of the speed at which an algorithm is able to adjust its estimates to the time-varying system.

Optimization of tracking performance is in general associated to a minimization of the MSD and the MSE, mainly through control of the gain parameter. In attempting to do so, however, one is confronted with a well known trade-off between speed and accuracy in the estimates provided by the algorithms: on one extreme, tracking can be slower than the system actual time variations, but with less noisy estimates; on the other extreme, tracking can be made as rapid as the time-varying context, but with estimates much more
contaminated by noise (see e.g. Benveniste et al., 1990, Part I, Chapters 1 and 4).

Comparative analysis on the tracking performance of the LS and the SG algorithms can be found in Eweda (1994) and Haykin (2001, pp. 643-659), and their results indicate the preeminence of data conditions in the determination of which algorithm outperforms the other. To be more specific, the comparison between the LS and the SG algorithm in terms of tracking performance depends on how the covariance matrices of the regressors (\(x_t\) in Eq. 1.1) and of the disturbances affecting the time-varying coefficients (i.e., \(\omega_t\) in Eq. 1.19) relate to each other.

**Robustness**

In a context of model misspecification, an estimator is said to be robust if it does not magnify the effect of modeling errors on estimation errors, and the SG algorithm is known to be the maximally robust algorithm in this sense (see Hassibi et al., 1996; Evans et al., 2010, pp. 240-242). Loosely speaking, in a worst case of misspecification the magnitude of the prediction errors obtained from the SG estimation will never exceed the magnitude of the true model disturbances. The LS algorithm, in turn, does not have this same robust interpretation, a result that can be associated to its squared norm formulation (Hassibi and Kailath, 2001). In short, the robustness of the LS algorithm, as measured by the (median) level of the interval of possible factor values at which modeling errors are propagated into estimation errors, depends on statistics of the regressors data.

### 4.3 Choice between single algorithms

In this section we present the results of our empirical comparison between the LS and the SG algorithms in forecasting inflation and output growth, using US macroeconomic data.
4.3.1 Approach

Our approach is based on the learning-to-forecast exercises we proposed in section 1.4, which attempts to mimic the real-time environment faced by an economic agent in his/her process of expectations formation. Consistent to our previous analysis we keep most of the specifications of these exercises fixed.

Regarding the gain calibrations, we follow our previous chapter distinction regarding their determination as a choice or as a primitive parameter of agents’ behavior. On the method used for the selection of the gain values in real-time, however, we simplify our exposition by focusing in two of the main calibrations we explored in the previous chapter. Namely, we adopt the full-sample and the recursive methods of gain selection, hereinafter denoted as fixed and a time-varying gain calibrations. Both are based on a grid of 100 admissible gain values.

Our design unfolds into a three-stages routine to generate the forecasts associated to each learning algorithm: initialization, estimation and forecasting, and evaluation. We then use the LS and the SG algorithms to obtain recursive estimates of the parameters of VAR model specifications (1 to 4 lag orders) applied to real-time quarterly data on US inflation and output growth, from 1947q2 to 2011q4. The first 75 observations (up to 1965q4) are used for the smoothing-based initialization of the algorithms, and the next 60 observations (from 1966q1 to 1980q4) are used for the algorithms’ (first) time-varying calibration. Therefore, our evaluation sample corresponds to the period from 1981q1 to 2010q4.

We conduct three evaluation exercises. The first two compare the learning algorithms against each other, and are designed to reflect distinct perspectives on their use. One is to take the point of view of an economic agent, who has to build forecasts of variables relevant for his/her economic decisions; thus, what matters for this agent is the accuracy of such forecasts, as defined by their dis-
tance with actual future realizations. Our second exercise assumes the point of view of the researcher, who is interested in uncovering which mechanism better represents the behavior of the economic agents being modeled; thus, what matters for the researcher is instead the resemblance of the forecasts produced by the algorithms to the observed forecasts as revealed from surveys.

Our third exercise attempts to address the key issue on the plausibility of the adaptive learning approach. Namely, we compare the forecasting performance of the algorithm favored as representative in the previous two exercise to the accuracy of the survey forecasts. Our argument is that for the learning approach to be a plausible representation of how agents form their expectations, this exercise must bring evidence that the learning mechanism forecasts at least as good (or bad) as the surveys.

4.3.2 Results

We start looking over the forecasts associated to each algorithm and gain value included in the grid computations. Figure 4.1 presents surfaces of average past performance for each algorithm and variable, showing their evolution through time and for the different gain values. Two main observations arise:\(^6\): (i) the behavior of each algorithm depends on the variable being forecasted, whereas for a given variable the LS and SG algorithms behave differently; (ii) the magnitudes of forecast errors were relatively higher during the first decade in our sample, irrespective of the variable forecasted and the algorithm used. Rather than an indication of algorithms’ instability, this latter observation can be associated with the period of greater volatility that preceded the Great Moderation in the US economy (see Stock and Watson, 2003).

Shifting our focus to the results comparing the LS and SG algorithms, we present in Table 4.1 the hit rates associated to our evaluation measures. In

---

\(^6\) Inspection of such surfaces for different forecasting horizons and for the resemblance to surveys, bring up similar observations.
Figure 4.1: Evolution of algorithms’ forecasting accuracy through time.

The MSFE plotted are computed on the basis of the 60 quarters backwards to the dates indicated into the NW-SE axis. The gain calibrations used for each algorithm are indicated into the SW-NE axis, and correspond to those experimentally calibrated for the algorithms’ stability. Forecast errors refer to mean-aggregated errors over 5 horizons of forecasts.
Table 4.1: Hit rates comparing the algorithms.

(a) Forecast accuracy.

<table>
<thead>
<tr>
<th>Variables</th>
<th>LS wins</th>
<th>SG wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Gains</td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td><strong>Inflation, gains as a choice</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>20%</td>
<td>0%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>35%</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Inflation, gains as a primitive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>65%</td>
<td>0%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>60%</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Growth, gains as a choice</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>95%</td>
<td>75%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>75%</td>
<td>30%</td>
</tr>
<tr>
<td><strong>Growth, gains as a primitive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>100%</td>
<td>70%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>100%</td>
<td>55%</td>
</tr>
</tbody>
</table>

(b) Forecast resemblance.

<table>
<thead>
<tr>
<th>Variables</th>
<th>LS wins</th>
<th>SG wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Calibrations</td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td><strong>Inflation, gains as a choice</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>80%</td>
<td>45%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>100%</td>
<td>75%</td>
</tr>
<tr>
<td><strong>Inflation, gains as a primitive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>95%</td>
<td>55%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>100%</td>
<td>75%</td>
</tr>
<tr>
<td><strong>Growth, gains as a choice</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>75%</td>
<td>70%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>65%</td>
<td>40%</td>
</tr>
<tr>
<td><strong>Growth, gains as a primitive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>80%</td>
<td>70%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>75%</td>
<td>65%</td>
</tr>
</tbody>
</table>

See footnotes to Table 3.4. The fixed and the time-varying gain specifications here stand for the full-sample and the recursive methods of chapter 3.
terms of accuracy a clear dominance of the LS is observed in forecasting growth. For inflation, the results depend on the approach adopted to the determination of the gains: while the SG is preferred under the gain as a choice, the LS tends to outperform the SG when the gain is taken as a primitive. Notice, however, that the tests for equal predictive ability rarely find statistical significance for the LS victories in forecasting inflation. Hence, our reading of these accuracy results is that they represent mixed evidence in support of each of the algorithms.

The results on the forecasts’ resemblance to surveys, in contrast, indicate an overwhelming dominance of the LS algorithm. Other than beating the SG resemblance with a higher frequency, the victories of the LS are often found with statistical significance by the tests of equal predictive ability. Also notice that, if the SG presents any threat to the LS dominance, this would not be on the grounds of their resemblance to inflation but to the growth survey forecasts. Clearly, this is a disturbing observation given that in our previous exercise the slightly favorable evidence for the SG was observed in forecasting inflation, but not for growth.

The majority of the results obtained in the comparison between the LS and the SG algorithms is, nevertheless, pointing to the former algorithm as the best choice to serve as an unique representative. Hence, for our third exercise we compare the forecasting performance of the LS algorithm to that of the survey forecasts. Following to our analysis in chapter 3, we also restrict our attention to the calibrations of this algorithm taking the gain as a primitive. The hit rates associated to this evaluation are presented in Table 4.2. As a reference we also present the LS hit rates associated to our preliminary assessment on this issue in section 1.5.

Our results evidence some remarkable improvements brought by our calibration efforts to the quality of the forecasts of the LS learning algorithm relative to the surveys. In forecasting inflation, the LS is found to match the performance of the surveys if the first horizon of forecasts is neglected, which can
Table 4.2: Hit rates comparing LS vs. survey forecasts accuracy.

(a) Inflation.

<table>
<thead>
<tr>
<th>Gains</th>
<th>LS wins</th>
<th>Survey wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>Constant gain (from chapter 1):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- h=0..4</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=1..4</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=2..4</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Fixed gains as a <em>primitive</em>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- h=0..4</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=1..4</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=2..4</td>
<td>67%</td>
<td>0%</td>
</tr>
<tr>
<td>Time-varying gains as a <em>primitive</em>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- h=0..4</td>
<td>40%</td>
<td>15%</td>
</tr>
<tr>
<td>- h=1..4</td>
<td>50%</td>
<td>19%</td>
</tr>
<tr>
<td>- h=2..4</td>
<td>67%</td>
<td>25%</td>
</tr>
</tbody>
</table>

(b) Growth.

<table>
<thead>
<tr>
<th>Gains</th>
<th>LS wins</th>
<th>Survey wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>Constant gain (from chapter 1):</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- h=0..4</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=1..4</td>
<td>6%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=2..4</td>
<td>8%</td>
<td>0%</td>
</tr>
<tr>
<td>Fixed gains as a <em>primitive</em>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- h=0..4</td>
<td>30%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=1..4</td>
<td>37.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=2..4</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>Time-varying gains as a <em>primitive</em>:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- h=0..4</td>
<td>30%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=1..4</td>
<td>37.5%</td>
<td>0%</td>
</tr>
<tr>
<td>- h=2..4</td>
<td>50%</td>
<td>0%</td>
</tr>
</tbody>
</table>

The hit rates vary on the horizons included. Those for the constant gain can be computed directly from Tables 1.2 and 1.3. See also footnotes to Tables 3.4 and 4.1.
be reasonably justified due to informational issues affecting the comparative\(^7\). The results are less favorable, though still improving, for the LS in forecasting growth, where exclusion of the first two horizons from the evaluation is required to obtain a balance between the learning algorithm and the survey. Finally, notice that the tests of equal predictive ability were found most of the times favoring the surveys victories; statistical significance in favor of the LS algorithm was found only in forecasting inflation, with its frequency increasing with the horizon of the forecasts.

4.3.3 Discussion

An understanding of our results on the algorithms’ forecasting performances can be obtained by considering the different statistical properties of the variables being forecasted. Compared to inflation rates, output growth is known to have a lower degree of dynamic persistence and a higher degree of volatility. These characteristics naturally make output growth a variable harder to forecast than inflation (see, e.g., Patton and Timmermann, 2011). Hence, it seems reasonable to expect that a more sophisticated method would be favored in forecasting growth.

Our results corroborate this idea: the “sophisticated” LS tended to outperform the computationally simpler SG algorithm in forecasting growth. Still, it is instructive to see that, in some cases, the SG was able to outperform the LS in forecasting inflation. These results thus confirm our preliminary view that the relative suitability of these algorithms depends mainly on the statistical properties of the data environment to which they are applied. Taking the surveys as a reference, nevertheless, pointed to an overall dominance of the LS in providing forecasts closer to those representing actual agents’ forecasts. Clearly, this result provides support for the usually unquestioned choice of the LS learning scheme in the applied literature on adaptive learning in macroeconomics.

\(^7\text{Recall our discussion of this issue in section 1.5.}\)
However, the joint interpretation of these resemblance results with those for the algorithms’ accuracy reveals a puzzling conflict. The motivation we gave to these two exercises came from what one can think of as two distinguished pragmatic perspectives on the choice of a learning algorithm. For an economic agent, the requisite is of a mechanism capable of providing accurate inferences and, assuming this agent behaves rationally, such a choice would be guided by an optimization of forecasting performance. For a researcher, in contrast, the choice of a learning algorithm reflects his quest for the true mechanism representative of the former agent’s behavior. Rephrasing our findings in these interpretative terms, in order to form expectations for output growth and inflation: (i) neither the LS nor the SG algorithms dominate as what would have been the agents’ rational choice; but, (ii) the LS was found to be the closest representative of agents’ learning mechanism for both variables.

A key aspect for the comprehension of this finding relates to the distinct approaches to the determination of the gains, either as a choice or as a primitive parameter of agents behavior. The above conflict seems to be driven mainly by the results on the performance of the algorithms with gains calibrated as a choice. Our discussion in chapter 3 already pointed to the existence of conflicts to the internal consistency of this approach to gain determination within a rational optimizing framework. Therefore, our results seem to indicate that taking both the gain and the algorithm as agents’ choices also leads to evidence in conflict with actual agents forecasting behavior, as observed from surveys. When the gain is taken as a primitive parameter, our results on agents’ preferred learning algorithm are found roughly coherent to the algorithms forecasting performance.

Our ultimate conclusion on the choice of a learning algorithm representative of agents learning-to-forecast behavior is therefore in favor to the LS with primitive gain(s). Furthermore, our evaluation of this algorithm’s forecasting accuracy relative to the survey forecasts provided an assessment favorable for
the plausibility of the learning approach. Our argument at this point required that the forecasts associated to the learning approach should perform no better, nor worse, than the forecasts observed from actual agents through the surveys.

Accordingly, our results at longer forecasting horizons, which are less prone to informational issues, pointed to a balanced performance between the algorithm and the survey’s forecasts. Other than returning a positive assessment on the choice of the LS as a representative algorithm, this finding is also suggestive of the value of our calibration efforts. Nevertheless, the statistical tests of equal predictive ability tended to favor the surveys, weakening the evidence in favor of the plausibility of the learning approach; i.e., the plausibility of learning requires the algorithm’s and the survey’s forecasts to be statistically indistinguishable. An attempt to improve on this assessment is provided in the next section.

4.4 Learning with paired algorithms

In this section we introduce and evaluate the Hybrid (HY) and the Mixing (MX) algorithms. Our evaluation will follow along similar lines as the previous section approach. We compare the new learning mechanisms both between them, and against the best performing algorithms from the previous section. The best performing paired algorithms are finally used for an assessment of the plausibility of learning.

4.4.1 Hybrid and Mixing algorithms

Our previous analysis on the performance of the LS and the SG algorithms suggested that there might be cases favoring the use of both of them, depending on the properties of the data. One natural way to exploit the features particular to each of these algorithms is to employ weighted averaging rules. This is the central idea behind our Hybrid and Mixing algorithms, which stand as
paired formulations of the previous algorithms.

The rationale behind this approach can be traced back to early research on the use of combined forecasts in order to attenuate the potential biases present in the individual series of forecasts (see Granger and Ramanathan, 1984). Thus, agents’ use of the paired algorithms can be motivated as a result of their own interest in finding ways to merge the distinct forecasting methodologies at their disposal. Besides, just as the LS was motivated in the learning literature for its widespread knowledge between econometricians, the use of forecast combinations also has a long tradition in the literature (see Timmermann, 2006).

In our Hybrid and Mixing algorithms a weight parameter regulates how close each of them behaves as the LS or as the SG. Their main difference relates to the level at which the weighting is applied: whereas in the HY the LS and the SG are combined at the level of their updating recursions, in the MX the combination is at the coefficients estimates level.

**Algorithm 5** (HY). Under the estimation context of (1.1), the Hybrid algorithm assumes the form of

\[ \hat{\theta}_t^{HY} = \hat{\theta}_t^{HY} + H_t x_t \left( y_t - x_t' \hat{\theta}_t^{HY} \right), \quad (4.1) \]

\[ H_t = \omega_t \gamma_t R_t^{-1} + (1 - \omega_t) \mu_t I, \quad (4.2) \]

where \( \omega_t \) is the weight given to the LS algorithm in the mixture gain matrix \( H_t \), and the remaining components follow from the definitions of the LS and SG algorithms.

Noticing that the SG results as a special case of the LS when \( R_t = I \) and \( \gamma_t = \mu_t \), it is clear that the Hybrid algorithm encompasses both of the traditional algorithms\(^8\): when the weight equals to unity, 4.1-4.2 reduces to 1.2; when the weight is null, 4.1-4.2 reduces to 1.4. However, it is important to

\(^8\)We also experimented with a nonequivalent alternative replacing \( H_t \) in 4.1 by \( h_t^{-1} \), with \( h_t = \omega_t \gamma_t^{-1} R_t^{-1} + (1 - \omega_t) \mu_t^{-1} I \), but its performance was slightly inferior.
note that, unless \( \omega_t = 1 \) or \( \omega_t = 0 \) for all \( t \), the stream of coefficients estimates obtained with the Hybrid algorithm will rarely coincide with any of the estimates provided by the LS or the SG alone. In other words, even if at a given period \( t \) the Hybrid can behave exactly as the LS (\( \omega_t = 1 \)) or as the SG (\( \omega_t = 0 \)) would, their estimates are not supposed to coincide, given that their previous estimates may differ.

A simpler alternative is provided by the Mixing algorithm, where the final coefficients estimates are represented by a weighted average of those obtained with the LS and the SG algorithms.

**Algorithm 6 (MX).** Under the estimation context of (1.1), the Mixing algorithm assumes the form of

\[
\hat{\theta}_t^{MX} = w_t \hat{\theta}_t^{LS} + (1 - w_t) \hat{\theta}_t^{SG}.
\]

where \( w_t \) is the weight given to the estimates obtained from the LS algorithm, and the remaining components follow from the definitions of the LS and SG algorithms.

In contrast to the Hybrid, the Mixing algorithm does not “suffer” from a dependence on its previous period estimates. By weighting the LS and SG algorithms directly from their coefficients estimates, the MX algorithm is more closely related to standard econometric approaches of forecasts combination (see Timmermann, 2006).

The computational implementation of the paired algorithms follows the usual process adopted for the previous single algorithms, the LS and the SG. The projection facility, for example, may also be invoked for the paired algorithms if their estimates fall outside the stability region. For the initialization of the paired algorithms we adopt a weighted average of the initializations used for the single algorithms.
4.4.2 Calibration of weights

The paired algorithms require the specification of a sequence of weights parameters. For the calibration of these weights we follow an approach similar to that we have used for the learning gains, matching both their determination and their selection methods with a grid of 101 weight values (0, 0.01, \ldots, 0.99, 1). Particularly, we adopt four alternative calibrations: \textit{fixed} gains/weights and \textit{time-varying} gains/weights, both determined as a \textit{choice} and as a \textit{primitive}. It is important to note, however, that for the calibration of the paired algorithms’ weights we take the learning gains for each single algorithm as given\textsuperscript{10}.

In Table 4.3 we present the resulting \textit{fixed} weight calibrations for the HY and MX algorithms, for varying forecasting horizons and model specifications. Overall we observe that the LS dominance we found in our previous analysis is reflected in the weight calibrations for the paired algorithms; the average weight taking both algorithms together is equal to 0.69 for inflation and 0.61 for growth. Additionally we note that: the MX weights tend to be higher than those associated to the HY algorithm; and, the \textit{primitive} weights tend to be smaller than those obtained as a \textit{choice}, specially in the first forecasting horizon.

A different picture emerges with the \textit{time-varying} weights calibrations, presented in Figure 4.2 focusing in the selections for the first horizon of forecast. There we can see that the weights on the LS algorithm tend to decrease with time, although this trend reverts during the last decade for growth. Recall that our sample covers the Great Moderation period (between 1985 and 2006, see Stock and Watson, 2003), which has been characterized by low levels of growth rates volatility. Hence, the patterns we observe in the evolution of the time-varying weights suggest that the SG is favored during periods of macroe-

\textsuperscript{9}It would also be possible to explore different combinations of gains and weights calibrations, e.g., fixed weights pairing a LS with time-varying gains and a SG with fixed gains.

\textsuperscript{10}Alternatively, one could have used a two-dimensional grid to perform a joint selection of both gains and weights, though this would represent a substantial increase in the computational burden of our routines.
Table 4.3: Fixed weight calibrations.

(a) Inflation.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Weight as a choice</th>
<th></th>
<th></th>
<th>Weight as a primitive</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h=0</td>
<td>h=2</td>
<td>h=4</td>
<td>h=0</td>
<td>h=2</td>
<td>h=4</td>
</tr>
<tr>
<td>Hybrid:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- VAR(1)</td>
<td>0.68</td>
<td>1.00</td>
<td>1.00</td>
<td>0.60</td>
<td>0.76</td>
<td>0.74</td>
</tr>
<tr>
<td>- VAR(2)</td>
<td>0.95</td>
<td>0.54</td>
<td>0.55</td>
<td>0.66</td>
<td>0.68</td>
<td>0.66</td>
</tr>
<tr>
<td>- VAR(3)</td>
<td>0.72</td>
<td>0.33</td>
<td>0.30</td>
<td>0.55</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>- VAR(4)</td>
<td>0.64</td>
<td>0.55</td>
<td>0.57</td>
<td>0.52</td>
<td>0.62</td>
<td>0.62</td>
</tr>
<tr>
<td>Mixing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- VAR(1)</td>
<td>0.86</td>
<td>0.84</td>
<td>0.86</td>
<td>0.86</td>
<td>0.92</td>
<td>0.85</td>
</tr>
<tr>
<td>- VAR(2)</td>
<td>0.93</td>
<td>0.61</td>
<td>0.64</td>
<td>0.79</td>
<td>0.83</td>
<td>0.81</td>
</tr>
<tr>
<td>- VAR(3)</td>
<td>0.77</td>
<td>0.47</td>
<td>0.44</td>
<td>0.62</td>
<td>0.75</td>
<td>0.76</td>
</tr>
<tr>
<td>- VAR(4)</td>
<td>0.71</td>
<td>0.72</td>
<td>0.73</td>
<td>0.57</td>
<td>0.69</td>
<td>0.69</td>
</tr>
</tbody>
</table>

(b) Growth.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Weight as a choice</th>
<th></th>
<th></th>
<th>Weight as a primitive</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h=0</td>
<td>h=2</td>
<td>h=4</td>
<td>h=0</td>
<td>h=2</td>
<td>h=4</td>
</tr>
<tr>
<td>Hybrid:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- VAR(1)</td>
<td>0.71</td>
<td>0.16</td>
<td>0.11</td>
<td>0.23</td>
<td>0.47</td>
<td>0.45</td>
</tr>
<tr>
<td>- VAR(2)</td>
<td>0.72</td>
<td>0.76</td>
<td>0.64</td>
<td>0.58</td>
<td>0.94</td>
<td>1.00</td>
</tr>
<tr>
<td>- VAR(3)</td>
<td>0.48</td>
<td>0.78</td>
<td>0.72</td>
<td>0.23</td>
<td>0.69</td>
<td>0.66</td>
</tr>
<tr>
<td>- VAR(4)</td>
<td>0.33</td>
<td>0.58</td>
<td>0.72</td>
<td>0.15</td>
<td>0.49</td>
<td>0.57</td>
</tr>
<tr>
<td>Mixing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- VAR(1)</td>
<td>0.74</td>
<td>0.56</td>
<td>0.39</td>
<td>0.55</td>
<td>0.72</td>
<td>0.83</td>
</tr>
<tr>
<td>- VAR(2)</td>
<td>0.78</td>
<td>0.84</td>
<td>0.70</td>
<td>0.47</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>- VAR(3)</td>
<td>0.55</td>
<td>0.81</td>
<td>0.74</td>
<td>0.31</td>
<td>0.73</td>
<td>0.70</td>
</tr>
<tr>
<td>- VAR(4)</td>
<td>0.45</td>
<td>0.67</td>
<td>0.75</td>
<td>0.27</td>
<td>0.63</td>
<td>0.70</td>
</tr>
</tbody>
</table>
Figure 4.2: Evolution of time-varying weight calibrations through time.

(a) HY on inflation.

(b) HY on growth.

(c) MX on inflation.

(d) MX on growth.

The recursive calibrations of the weights represent discrete values based on the grid of weights computed for each algorithm.

Economic stability.

Another interesting observation from Figure 4.2 relates to a widening difference between the weights determined according to the different approaches from around the middle of our sample. Particularly, in the second part of our sample the weights determined as a choice tended to favor the SG for inflation and the LS for growth more strongly than those determined as a primitive. From their construction, these results suggest a disagreement between the learning mechanisms agents were using during this period and their corresponding forecasting performance.

The above observation seems consistent to the evidence of state-dependent informational rigidities recently presented by Coibion and Gorodnichenko (2012).
Imperfect information models (see Mankiw and Reis, 2010, for a survey) relax the assumption that agents have full information about the economy, as inherent to the RE approach, by the introduction of informational rigities, or limitations to agents informational processing capabilities. One important implication of these models is that high levels of economic volatility can be associated to an increase of the efforts agents devote to the collection and processing of information. Coibion and Gorodnichenko (2012) use this feature to estimate the endogenous responses of informational rigidities during the Great Moderation period. The upshot from their analysis is that the lower levels of macroeconomic volatility during that period has induced a growing degree of information rigidity, which reflected into a greater complacency of agents to forecasting errors, hence providing an explanation for our observation from Figure 4.2.

Following our approach of adaptive gains in the previous chapter we have also extended the paired algorithms to have their weights updated adaptively\(^{11}\). However, these mechanisms showed a poorer performance than the calibrations above. Results comparing these gain calibrations are presented in the appendix C. Thus, for conciseness, we exclude the paired algorithms with adaptive weights from our results presentation.

### 4.4.3 Results and discussion

We begin our evaluation of the paired algorithms by looking at how they compare to each other. The hit rates relevant to this comparative are presented in Table 4.4. In terms of accuracy a clear dominance of the MX is observed in forecasting inflation. For growth, the results are not so clear as there is balanced evidence in favor of both algorithms. The results for the time-varying calibration of gains/weights as a choice seem to be particularly favorable to the HY algorithm in forecasting the growth variable; we note, however, that the sta-

\(^{11}\)See appendix A.8 for the derivations.
Table 4.4: Hit rates comparing the paired algorithms.

(a) Forecast accuracy.

<table>
<thead>
<tr>
<th>Variables</th>
<th>HY wins</th>
<th></th>
<th>MX wins</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
<td>GW-20%</td>
<td>Hit rate</td>
</tr>
<tr>
<td>Inflation, gains/weights as a choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>5%</td>
<td>0%</td>
<td>0%</td>
<td>95%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>10%</td>
<td>0%</td>
<td>5%</td>
<td>90%</td>
</tr>
<tr>
<td>Inflation, gains/weights as a primitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>5%</td>
<td>0%</td>
<td>0%</td>
<td>95%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Growth, gains/weights as a choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>45%</td>
<td>0%</td>
<td>10%</td>
<td>55%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>50%</td>
<td>15%</td>
<td>5%</td>
<td>50%</td>
</tr>
<tr>
<td>Growth, gains/weights as a primitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>25%</td>
<td>5%</td>
<td>5%</td>
<td>75%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>45%</td>
<td>10%</td>
<td>15%</td>
<td>55%</td>
</tr>
</tbody>
</table>

(b) Forecast resemblance.

<table>
<thead>
<tr>
<th>Variables</th>
<th>HY wins</th>
<th></th>
<th>MX wins</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
<td>GW-20%</td>
<td>Hit rate</td>
</tr>
<tr>
<td>Inflation, gains/weights as a choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>70%</td>
<td>60%</td>
<td>45%</td>
<td>30%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>75%</td>
<td>40%</td>
<td>40%</td>
<td>25%</td>
</tr>
<tr>
<td>Inflation, gains/weights as a primitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>100%</td>
<td>85%</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>100%</td>
<td>65%</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>Growth, gains/weights as a choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>30%</td>
<td>5%</td>
<td>25%</td>
<td>70%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>15%</td>
<td>0%</td>
<td>0%</td>
<td>85%</td>
</tr>
<tr>
<td>Growth, gains/weights as a primitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed</td>
<td>40%</td>
<td>15%</td>
<td>30%</td>
<td>60%</td>
</tr>
<tr>
<td>- Time-varying</td>
<td>30%</td>
<td>0%</td>
<td>20%</td>
<td>70%</td>
</tr>
</tbody>
</table>

See footnotes to Table 3.4.
tistical significance results on the tests for equal predictive ability under this particular comparison still tended to favor the MX algorithm.

The results on the forecasts’ resemblance to surveys, in contrast, indicate these algorithms exchange their positions: whereas the HY presents a closer resemblance to the survey forecasts for inflation more frequently than the MX, it is this latter algorithm’s forecasts that better resembles to the survey forecasts for the growth variable. But again, the results on the statistical significance of these comparisons are less conclusive for the case of the growth variable. In spite of these uncertainties, our reading from these results on the comparison between our two alternative formulations of paired algorithms reinforces our previous finding of a conflict between our two evaluation criteria.

It is interesting in that respect to investigate how these paired algorithms perform relative to our previous single algorithms. The hit rates comparing the best performing specifications of each of these classes of algorithms are presented in Table 4.5. Recall from Table 4.1 that the LS was found to beat the SG in most cases of calibrations and variables, except in forecasting inflation with the gain determined as a choice. Hence, this is the only case where the SG is taken as representative of the single algorithms approach in Table 4.5. Similar arguments are used to the specification of the algorithm representative of the paired approach, in accordance to our discussion above.

Overall the results indicate that the paired algorithms are successful at improving upon the performance of the single specifications. This is particularly true for the resemblance to surveys evaluation criterion, where, for example, the forecasts of the HY with a fixed gain/weight calibrated as a primitive are found more closely resembling to the survey forecasts for inflation at every comparison with the LS. Interestingly, in some cases of time-varying calibrations the LS is found to outperform the competing paired algorithm. Nevertheless, a closer inspection of the performance of the paired algorithms gain/weight selection methods (see Table C.7 in the appendix) indicates that these were cases where
Table 4.5: Hit rates comparing best single and paired algorithms.

(a) Forecast accuracy.

<table>
<thead>
<tr>
<th>Variables</th>
<th>LS wins Hit rate</th>
<th>SG wins Hit rate</th>
<th>HY wins Hit rate</th>
<th>MX wins Hit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DM-20%</td>
<td>GW-20%</td>
<td>DM-20%</td>
<td>GW-20%</td>
</tr>
<tr>
<td>Inflation, gains/weights as a choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed 10%</td>
<td>0%</td>
<td>5%</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>- Time-varying 60%</td>
<td>0%</td>
<td>30%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inflation, gains/weights as a primitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed 25%</td>
<td>0%</td>
<td>10%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Time-varying 30%</td>
<td>0%</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Growth, gains/weights as a choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed 40%</td>
<td>10%</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Time-varying 60%</td>
<td>0%</td>
<td>10%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Growth, gains/weights as a primitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed 15%</td>
<td>5%</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Time-varying 55%</td>
<td>20%</td>
<td>25%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) Forecast resemblance.

<table>
<thead>
<tr>
<th>Variables</th>
<th>LS wins Hit rate</th>
<th>SG wins Hit rate</th>
<th>HY wins Hit rate</th>
<th>MX wins Hit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DM-20%</td>
<td>GW-20%</td>
<td>DM-20%</td>
<td>GW-20%</td>
</tr>
<tr>
<td>Inflation, gains/weights as a choice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed 30%</td>
<td>10%</td>
<td>0%</td>
<td>20%</td>
<td></td>
</tr>
<tr>
<td>- Time-varying 15%</td>
<td>0%</td>
<td>15%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inflation, gains/weights as a primitive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Fixed 10%</td>
<td>0%</td>
<td>10%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Time-varying 25%</td>
<td>0%</td>
<td>25%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The algorithms picked for each comparison reflect the results presented in tables 4.1 and 4.4.
Table 4.6: Hit rates comparing paired algs. vs. survey forecasts accuracy.

(a) Inflation.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Paired algorithm wins</th>
<th>Survey wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>MX - fixed gain/weight as a choice:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h=0.4</td>
<td>30%</td>
<td>0%</td>
</tr>
<tr>
<td>h=1.4</td>
<td>37.5%</td>
<td>0%</td>
</tr>
<tr>
<td>h=2.4</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>MX - fixed gain/weight as a primitive:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h=0.4</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td>h=1.4</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>h=2.4</td>
<td>67%</td>
<td>0%</td>
</tr>
</tbody>
</table>

(b) Growth.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Paired algorithm wins</th>
<th>Survey wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>HY - fixed gain/weight as a choice:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h=0.4</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td>h=1.4</td>
<td>50%</td>
<td>0%</td>
</tr>
<tr>
<td>h=2.4</td>
<td>67%</td>
<td>0%</td>
</tr>
<tr>
<td>HY - fixed gain/weight as a primitive:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h=0.4</td>
<td>35%</td>
<td>0%</td>
</tr>
<tr>
<td>h=1.4</td>
<td>44%</td>
<td>0%</td>
</tr>
<tr>
<td>h=2.4</td>
<td>58%</td>
<td>0%</td>
</tr>
</tbody>
</table>

See footnotes to Table 4.2.

the time-varying calibration did not perform well.

The improved forecasting performance brought by the paired algorithms provides an important addition to our understanding for the recurring observations of a conflict between our two evaluation criteria, where: the forecasts’ accuracy is taken as representative of the interests of a rational agent; and the forecasts’ resemblance is taken to represent the behavior of actual agents. We have seen that, relative to our evidence based on the single algorithms, the paired algorithms reinforce our findings of such a conflict. The improved performance of the latter, therefore, indicates that the conflict reflects an inefficiency on agents’ side rather than a failure of the learning algorithms as representative of agents learning-to-forecast behavior.

152
One way to deepen our understand in this issue is to compare the paired algorithms forecasting accuracy to that of the survey forecasts, focusing on the specifications emerging as the best representatives from the analysis above. The hit rates associated to these comparisons are presented in Table 4.6. These results evidence improvements similar to what we have found for the single algorithms. For inflation, the MX with a fixed gain/weight determined as a primitive is found again to match the performance of the surveys leaving only the first horizon of forecasting out of the evaluation.

But perhaps the most important result obtained with the paired algorithms relates to the improvements they brought to the forecasts for the growth variable. Our previous evaluation of the plausibility of the learning approach indicated that the main difficulty of the single algorithms was with respect to its forecasting performance for this variable. Here the exclusion of only the first forecasting horizon is required for the HY algorithm to attain a balance on the comparative to the surveys in forecasting growth, a result similar to what we have found for the inflation variable.

Therefore, the results presented in this section provide evidence favorable to the plausibility of the learning approach as representative of how agents form their expectations for applied macroeconomics purposes. In agreement to our previous findings in this thesis, our discussion took note on these results to argue in favor of the bounded rationality interpretation of adaptive learning. Namely, our argument sustains that agents inefficient adjustment of their learning mechanisms, to deal with the statistical changes in the macroeconomy, reflects limitations to their rationality.

4.5 Concluding remarks

In this chapter we have provided an empirical assessment on the choice of a learning algorithm to stand as representative of agents learning-to-forecast behavior. We did that by first comparing the LS to its simpler competing al-
alternative, the SG algorithm. Our findings provided support to most of the previous applied literature adopting the LS as a representative learning mechanism. We then proposed and evaluated two new formulations of algorithms that combine the previous two, denoted as the Hybrid and the Mixing algorithms. These algorithms were found successful to improve upon the performance of the traditional algorithms, though they have also highlighted the existence of a conflict on the use of two plausible selection criteria. Namely, our comparative assessments were based on both the algorithms’ forecasting accuracy and their resemblance to survey forecasts.

We have motivated the use of these two evaluation criteria in terms of the pragmatic views of an agent and of a researcher. While the former is thought of as the ultimate user of the learning algorithms, the latter seeks to uncover which of the above algorithms provides the closer representation for agents’ behavior. Assuming the economic agents are rationally optimizing in their choice for a learning mechanism, we have argued that from their perspective the best choice of an algorithm is given by the one with a superior forecasting performance. From a researcher perspective, however, the most plausible option is the one that more closely resembles what is observed as agents’ behavior, here represented by survey forecasts. Combining these two perspectives, we obtained conflicting results on the choice of a representative learning algorithm, where the algorithm selected using one criterion was not the same picked by the other.

We interpreted these findings as favorable to the plausibility of the learning approach though. Assessing the quality of the forecasts provided by the learning algorithms relative to those obtained from the surveys, we found that they get close to becoming statistically indistinguishable. We argue that considerations with respect to the informational restrictions involved in such a comparison are essential for the validity of this finding. Taking these aspects into account, we conclude that our results provide important qualifications to
the rationality of the adaptive learning approach. Particularly, our results re-
ject the view where agents self-interest drive their adaptive behavior towards
a statistical optimization of their learning mechanisms. Instead, our evidence
favors a bounded rationality interpretation of agents learning-to-forecast be-
havior.
Conclusions and future directions

In this thesis we have empirically assessed the adaptive learning approach to the modeling of expectations for applied macroeconomics. The need of more realistic behavioral assumptions in the modeling of agents expectations in a macroeconomic context has pushed the literature towards the adoption of bounded rationality approaches. Adaptive learning is one of such alternatives, proposing the use of recursive mechanisms to depict the evolution of agents’ beliefs over time.

The lack of systematic assessments of the empirical plausibility of the adaptive learning approach may be considered an important obstacle to its broader adoption. Here we have provided an attempt to fill this gap by evaluating three dimensions of the applied adaptive learning problem: (i) the initialization of the learning algorithms; (ii) the determination and calibration of the learning gains; and, (iii) the choice of a representative learning mechanism. Our main contribution is to show that a proper assessment of the adaptive learning approach requires going beyond the previous views in the literature about these issues.

We have also contributed with the proposal of a renewed estimation and evaluation framework for the analysis of the issues above. Digressing into the origins of the two main algorithms considered in the literature of adaptive learning, namely Least Squares (LS) and Stochastic Gradient (SG), we have found a connection between their non-recursive forms and their interpretation within a state-space unifying framework. Based on such connection, we extended the correspondence between the LS and the Kalman filter recursions to a formulation with time-varying gains of the former, and also presented a similar correspondence for the case of the SG. Our correspondences hold ex-
actly, in contrast to previous approximate results into the literature, and we have argued that this has favored our computational implementations.

In our analysis of the initialization of the learning algorithms, we have provided a critical review on the methods previously adopted into the literature either for simulation or empirical purposes. We found that none of these methods was able to pass the sieve of both criteria of coherence to the algorithm long run behavior and of feasibility within the data availability restrictions for macroeconomics. We then proposed a new smoothing-based initialization routine, derived from the correspondences above, and showed through simulations that our method meets both those criteria in exchange for a higher computational cost. A simple empirical application is also presented to demonstrate the relevance of initialization for beginning-of-sample inferences.

Another important insight we provide in this thesis relates to a distinction on the rationale given to the determination of the learning gains. We discuss and empirically evaluate two alternatives: the gain as a choice of rationally optimizing agents, and the gain as a primitive parameter of bounded rational agents. Our results provide strong evidence in favor of the latter interpretation, hence favoring the use of surveys data for their calibration. Furthermore, our evidence also points to some heterogeneity in the time evolution of this behavior with respect to the variable forecasted and the algorithm adopted.

We finally provided an empirical assessment on the choice of a representative learning algorithm. Comparing first the traditional LS and SG algorithms, our findings provided support to most of the previous applied literature adopting the former as the representative learning mechanism. Challenging the view that learning should be represented by only one of the above algorithms, we then proposed and evaluated new formulations of algorithms that combine the previous two. These latter were found to successfully improve upon the performance of the traditional algorithms.

One recurrent issue throughout this thesis relates to a tension between the
bounded rationality aspect involved in the adaptive learning approach and what would be representative of agents self-interest to adapt towards a statistical optimization on the use of the learning mechanisms. Our main finding regarding this issue is that it is the bounded rationality point of view that prevails in what we observe from the data. Particularly, we have found that the expectations we evaluated, accessed through surveys of forecasts collected from actual agents, do not always reflect the optimal use of the learning algorithms. In spite of this conflict, we were able to obtain evidence favorable to the plausibility of the learning approach.

There are several possible extensions to our approach that we foresee contributing even further to our empirical understanding of the role of adaptive learning in applied macroeconomics. Given the empirical nature of this thesis, one first natural extension comes with the use of alternative sets of data to test for the robustness of our findings: within the context of US data, one can extend our analysis to incorporate alternative sources of survey forecasts, such as the Michigan’s Surveys of Consumers; widening the scope, one can also extend our applications to international sets of data, though the real-time requirements may constitute an important restriction. Another alternative would be to resort to data obtained from laboratory learning-to-forecast experiments, as in Hommes (2011).

Perhaps the main weakness of our approach relates to the lack of a structural model, which prevented us to obtain a complete characterization of the self-referentiality of expectations. Another natural extension would therefore involve exchanging the generality of our non-structural approach for the potential insights arising from an analysis of self-referentiality within particular structural models. Particular extensions to our analysis of the issues above include: an assessment of the relevance of the learning algorithms initializations for historical analysis of periods in the beginning of our usual samples of macroeconomic data; an analysis, both theoretical and empirical, of the effects
of algorithm’s misspecification over the predictions of macroeconomic models incorporating adaptive learning.

We also seek to extend our evaluation framework to cover traditional analysis of expectations rationality through tests for forecast bias, an approach initiated by Mincer and Zarnowitz (1969), and tests for the efficient use of information, following the Nordhaus (1987) approach. Both techniques develop into a regression-based evaluation of forecasts, which can be adapted to an analysis of the forecasts associated to the learning algorithms. Particularly, we note that our algorithm specification mixing the LS and the SG may be given a regression representation mixing the forecasts of the traditional algorithms.

Finally, in spite of highlighting several possibilities for future extensions, the general message conveyed in this thesis can be summarized as an optimistic assessment on the plausibility of adaptive learning. Our main findings point to an empirical support for the use of the adaptive learning approach as a representative of agents bounded rational behavior in their process of expectations formation. Nevertheless, our analysis also indicates that this conclusion is conditioned by important issues that have been neglected in the literature for the empirical implementation of this approach. We hope to have provided some guidance on the appropriate treatment of these issues.
Appendix A

Detailed derivations

A.1 Correspondence between non-recursive and recursive LS

First let

\[ R_t = \gamma_t \sum_{i=1}^{t} \beta(t, i) x_i x'_i. \]  \hfill (A.1)

from which we find that

\[ R_t = \gamma_t \sum_{i=1}^{t-1} \beta(t, i) x_i x'_i + \gamma_t x_t x'_t, \]  \hfill (A.2)

\[ = (1 - \gamma_t) \gamma_{t-1} \sum_{i=1}^{t-1} \beta(t-1, i) x_i x'_i + \gamma_t x_t x'_t, \]  \hfill (A.3)

\[ = (1 - \gamma_t) R_{t-1} + \gamma_t x_t x'_t, \]  \hfill (A.4)

\[ R_t = R_{t-1} + \gamma_t (x_t x'_t - R_{t-1}), \]  \hfill (A.5)

where in the second step we used

\[ \beta(t, i) = \frac{\gamma_{t-1}}{\gamma_t} (1 - \gamma_t) \beta(t-1, i), \]  \hfill (A.6)

which comes directly from (1.9).
For the vector of coefficients, note that using (A.1) in (1.7) we have that

\[ \hat{\theta}_{LS}^t = \gamma_t R_{t-1} \sum_{i=1}^{t} \beta(t,i) x_i y_i, \]  
(A.7)

\[ = \gamma_t R_{t-1} \left[ x_t y_t + \gamma_{t-1} (1 - \gamma_t) \sum_{i=1}^{t-1} \beta(t-1,i) x_i y_i \right], \]  
(A.8)

\[ = \gamma_t R_{t-1} \left[ x_t y_t + \frac{\gamma_{t-1}}{\gamma_t} (1 - \gamma_t) \sum_{i=1}^{t-1} \beta(t-1,i) x_i y_i \right], \]  
(A.9)

\[ \hat{\theta}_{LS}^t = \gamma_t R_{t-1} \sum_{i=1}^{t-1} \beta(t-1,i) x_i y_i, \]  
(A.10)

where in the third step we have again made use of (A.6). Now, note that lagging (A.7) one period and pre-multiplying it by \( R_{t}^{-1} R_{t-1} \) we have that

\[ R_{t}^{-1} R_{t-1} \hat{\theta}_{LS}^{t-1} = \gamma_{t-1} R_{t}^{-1} \sum_{i=1}^{t-1} \beta(t-1,i) x_i y_i, \]

which can then be substituted into (A.10) leading us to

\[ \hat{\theta}_{LS}^t = \gamma_t R_{t-1} x_t y_t + (1 - \gamma_t) R_{t-1} \hat{\theta}_{LS}^{t-1}, \]  
(A.11)

\[ = \gamma_t R_{t-1} x_t y_t + (1 - \gamma_t) R_{t-1} \ldots \]

\[ \ldots \left[ R_{t} (1 - \gamma_t)^{-1} - \gamma_t (1 - \gamma_t)^{-1} x_t x_t' \right] \hat{\theta}_{LS}^{t-1}, \]  
(A.12)

\[ = \gamma_t R_{t-1} x_t y_t + \hat{\theta}_{LS}^{t-1} - \gamma_t R_{t-1} x_t x_t' \hat{\theta}_{LS}^{t-1}, \]  
(A.13)

\[ \hat{\theta}_{LS}^t = \hat{\theta}_{LS}^{t-1} + \gamma_t R_{t}^{-1} x_t \left( y_t - x_t' \hat{\theta}_{LS}^{t-1} \right), \]  
(A.14)

where in the second step we made use of the expression preceding (A.5) above. Note that (A.5) and (A.14) assume exactly the same form as of (1.3) and (1.2), respectively, thus establishing the correspondence at scrutiny.
A.2 Correspondence between non-recursive and recursive SG

Let's first take out the last term of the summation in (1.10) resulting in

\[ \hat{\theta}_{SG}^{t} = \mu_{t} x_{t} y_{t} + \sum_{i=1}^{t-1} \beta (t, i) x_{i} y_{i}. \]  

(A.15)

Now, notice that from (1.6) and (1.11) we have that

\[ \beta (t, i) = \mu_{i} \prod_{k=i+1}^{t} (I - \mu_{k} x_{k} x_{k}',) \]  

(A.16)

\[ = (I - \mu_{t} x_{t} x_{t}') \mu_{i} \prod_{k=i+1}^{t-1} (I - \mu_{k} x_{k} x_{k}'), \]  

(A.17)

\[ \beta (t, i) = (I - \mu_{t} x_{t} x_{t}') \beta (t - 1, i). \]  

(A.18)

Substituting (A.18) into (A.15) we then have

\[ \hat{\theta}_{SG}^{t} = \mu_{t} x_{t} y_{t} + (I - \mu_{t} x_{t} x_{t}') \sum_{i=1}^{t-1} \beta (t - 1, i) x_{i} y_{i}, \]  

(A.19)

\[ = \mu_{t} x_{t} y_{t} + (I - \mu_{t} x_{t} x_{t}') \hat{\theta}_{SG}^{t-1}, \]  

(A.20)

\[ \hat{\theta}_{SG}^{t} = \hat{\theta}_{SG}^{t-1} + \mu_{t} x_{t} \left( y_{t} - x_{t}' \hat{\theta}_{SG}^{t-1} \right), \]  

(A.21)

which has exactly the same form as of (1.4), thus establishing the correspondence at scrutiny.

A.3 Correspondence with general Kalman filter

Following Hamilton (1994, pp. 399-400) notation, we consider a general state-space model with stochastically varying coefficients given by

\[ \xi_{t+1} = F(x_{t}) \xi_{t} + v_{t+1}, \]  

(A.22)

\[ y_{t} = a(x_{t}) + [H(x_{t})]' \xi_{t} + w_{t}, \]  

(A.23)
where $\xi_t$ is a vector of unobserved coefficients (states), $y_t$ is a vector of observable variables, $x_t$ is a vector of exogenous or predetermined variables, $F (x_t)$ and $H (x_t)$ are matrix-valued functions of $x_t$, and $a (x_t)$ is a vector-valued function $x_t$, all with conformable dimensions. The vectors of noises $v_{t+1}$ and $w_t$ are assumed to be mutually independent and distributed according to a Gaussian distribution, conditionally on $I_t = (x_t, x_{t-1}, \ldots, x_1, y_{t-1}, \ldots, y_1)$, with mean zero and variances given by $Q (x_t)$ and $R (x_t)$, respectively. Assuming further that the initial state $\xi_1 \sim N (\hat{\xi}_1 | 0, P_1 | 0)$, the optimal estimates of states $\xi_t$ are obtained through the Kalman filter equations given by

$$
\hat{\xi}_{t | t} = \hat{\xi}_{t | t-1} + K_t \left( y_t - a (x_t) - [H (x_t)]^\prime \hat{\xi}_{t | t-1} \right), \quad \text{(A.24)}
$$

$$
P_{t | t} = \left[ I - K_t [H (x_t)]^\prime \right] P_{t | t-1}, \quad \text{(A.25)}
$$

$$
K_t = P_{t | t-1} H (x_t) \left( [H (x_t)]^\prime P_{t | t-1} H (x_t) + R (x_t) \right)^{-1}, \quad \text{(A.26)}
$$

$$
\hat{\xi}_{t+1 | t} = F (x_t) \hat{\xi}_{t | t}, \quad \text{(A.27)}
$$

$$
P_{t+1 | t} = F (x_t) P_{t | t} F (x_t)^\prime + Q (x_t). \quad \text{(A.28)}
$$

where the subscripts indicate the timing of information associated to each estimate, e.g., $t + 1 | t$ means the inference standing for period $t + 1$ of the associated variable is made on the basis of data observed through period $t$.

To show that (1.20)-(1.22) represents the Kalman solution to the estimation of the time-varying coefficients of the model in (1.1) and (1.19), first let $\xi_t \equiv \theta_t$, $F (x_t) \equiv I$, $v_t \equiv \omega_t$, $y_t \equiv y_t$, $a (x_t) \equiv 0$, $H (x_t) \equiv x_t$, $w_t \equiv \epsilon_t$, $Q (x_t) \equiv \Omega_t$, and $R (x_t) \equiv \sigma_t^2$. Substituting these in (A.24)-(A.28) we get

$$
\hat{\theta}_{t | t} = \hat{\theta}_{t | t-1} + K_t \left[ y_t - x_t^\prime \hat{\theta}_{t | t-1} \right], \quad \text{(A.29)}
$$

$$
P_{t | t} = \left[ I - K_t x_t^\prime \right] P_{t | t-1}, \quad \text{(A.30)}
$$

$$
K_t = P_{t | t-1} x_t \left[ x_t^\prime P_{t | t-1} x_t + \sigma_t^2 \right]^{-1}, \quad \text{(A.31)}
$$

$$
\hat{\theta}_{t+1 | t} = \hat{\theta}_{t | t}, \quad \text{(A.32)}
$$

$$
P_{t+1 | t} = P_{t | t} + \Omega_t. \quad \text{(A.33)}
$$
Substituting (A.29) and (A.30) into (A.32) and (A.33) we get

\[
\hat{\theta}_{t+1|t} = \hat{\theta}_{t|t-1} + K_t \left[ y_t - x_t' \hat{\theta}_{t|t-1} \right], \tag{A.34}
\]

\[
P_{t+1|t} = \left[ I - K_t x_t' \right] P_{t|t-1} + \Omega_t, \tag{A.35}
\]

which is evidently equivalent to the recursions in (1.20)-(1.22) with \( \hat{\theta}_t \equiv \hat{\theta}_{t+1|t} \) and \( P_t \equiv P_{t+1|t} \).

**A.4 Correspondence between Kalman-based and ad hoc LS**

Let's begin by rearranging terms in (1.3) and using the matrix inversion lemma,

\[
(A + UCV)^{-1} = A^{-1} - A^{-1} U \left( C^{-1} + VA^{-1} U \right)^{-1} VA^{-1}, \tag{A.36}
\]

to find that

\[
\begin{align*}
\gamma_t \gamma_t^{-1} R_t &= \gamma_t^{-1} \left( 1 - \gamma_t \right) R_{t-1} + \frac{x_t}{U} \left( C \frac{1}{V} \right) x_t', \tag{A.37} \\
\gamma_t R_t^{-1} &= \frac{\gamma_t}{1 - \gamma_t} R_{t-1} - \frac{\gamma_t}{1 - \gamma_t} R_{t-1} x_t \ldots \\
&\ldots \left( 1 + x_t' \frac{\gamma_t}{1 - \gamma_t} R_{t-1} x_t \right)^{-1} x_t' \frac{\gamma_t}{1 - \gamma_t} R_{t-1}, \tag{A.38} \\
\frac{\gamma_t}{\gamma_{t-1}} R_t^{-1} &= \frac{\gamma_t}{\gamma_{t-1}} \left( 1 - \gamma_t \right) \left( I - \frac{R_{t-1} x_t x_t'}{1 - \gamma_t} + x_t' R_{t-1} x_t \right) R_{t-1}, \tag{A.39} \\
P_t^{LS} &= \frac{\gamma_t}{\gamma_{t-1}} \left( 1 - \gamma_t \right) \left( I - \frac{P_t^{LS} x_t x_t'}{x_t' P_{t-1} x_t + \gamma_{t-1} (1 - \gamma_t)} \right) P_{t-1}^{LS}, \tag{A.40}
\end{align*}
\]

where in the last line we let \( P_{t-1}^{LS} \equiv \gamma_t R_{t-1} \Rightarrow P_{t-1}^{LS} = \gamma_{t-1} R_{t-1}^{-1} \).

For the coefficients estimates recursion in (1.2), let \( K_t^{LS} \equiv \gamma_t R_t^{-1} x_t = P_t^{LS} x_t \)}
to then obtain

\[
K_{t}^{LS} = \frac{\gamma_{t}}{\gamma_{t-1} (1 - \gamma_{t})} \left( I - \frac{P_{t-1}^{LS} x_{t} x_{t}'}{x_{t} P_{t-1}^{LS} x_{t} + \frac{\gamma_{t-1}}{\gamma_{t}} (1 - \gamma_{t})} \right) P_{t-1}^{LS} x_{t}, \tag{A.41}
\]

\[
= \frac{\gamma_{t}}{\gamma_{t-1} (1 - \gamma_{t})} P_{t-1}^{LS} x_{t} \left( 1 - \frac{x_{t} P_{t-1}^{LS} x_{t}'}{x_{t} P_{t-1}^{LS} x_{t} + \frac{\gamma_{t-1}}{\gamma_{t}} (1 - \gamma_{t})} \right), \tag{A.42}
\]

\[
= \frac{\gamma_{t}}{\gamma_{t-1} (1 - \gamma_{t})} P_{t-1}^{LS} x_{t} \left( \frac{\gamma_{t-1}}{\gamma_{t}} (1 - \gamma_{t}) \right) \left( x_{t} P_{t}^{LS} x_{t} + \frac{\gamma_{t-1}}{\gamma_{t}} (1 - \gamma_{t}) \right), \tag{A.43}
\]

\[
K_{t}^{LS} = \frac{P_{t-1}^{LS} x_{t}}{x_{t} P_{t-1}^{LS} x_{t} + \frac{\gamma_{t-1}}{\gamma_{t}} (1 - \gamma_{t})}. \tag{A.44}
\]

Note that (A.44) and (A.40) assume exactly the same form as of (1.26) and (1.27), respectively, thus establishing the correspondence at scrutiny.

### A.5 Correspondence between Kalman filter and SG

First, let's show how (1.28) is obtained from substitution of (1.11) into (1.23). To see that, first notice that an scalar $\lambda_{t}$ equivalent to $\lambda_{t}$ in (1.11) can be obtained by solving the eigenvalue problem

\[
(I - \mu_{t} x_{t} x_{t}') z = \lambda_{t} z, \tag{A.45}
\]

for an arbitrary $K \times 1$ vector $z$. As usual, this problem can be solved by finding the $\lambda_{t}$ that makes the following determinant to be equal to zero,

\[
\det [(1 - \lambda_{t}) I - \mu_{t} x_{t} x_{t}'] = 0. \tag{A.46}
\]

Making use of the matrix determinant lemma, i.e.,

\[
\det [A + uv'] = \det [A] \left( 1 + v' A^{-1} u \right), \tag{A.47}
\]
we then find that

\[
\det \begin{bmatrix}
(1 - \lambda_t) \mathbf{I} - \mu_t \mathbf{x}_t \mathbf{x}_t' \\
\mathbf{A} u \\
\mathbf{v}'
\end{bmatrix} = (1 - \lambda_t)^K \left(1 - \mathbf{x}_t' \left(1 - \lambda_t\right)^{-1} \mathbf{I} \mu_t \mathbf{x}_t\right)
\]

(A.48)

\[
= (1 - \lambda_t)^K - (1 - \lambda_t)^{K-1} \mu_t \mathbf{x}_t' \mathbf{x}_t.
\]

(A.49)

Equating this last expression to zero and solving then we find that

\[
\lambda_t = 1 - \mu_t \mathbf{x}_t' \mathbf{x}_t,
\]

(A.50)

which is the scalar version of \(\lambda_t\) in (1.11) that we were looking for substitution into (1.23), together with \(\alpha_t\) from (1.11) as well. This results in

\[
\sigma_t^2 = (1 - \mu_t \mathbf{x}_t' \mathbf{x}_t) \mu_t^{-1},
\]

(A.51)

\[
= \mu_t^{-1} - \mathbf{x}_t' \mathbf{x}_t,
\]

(A.52)

which is recognizably equal to (1.28) as we wanted to show.

Now, to show the correspondence between the Kalman filter and the SG algorithm, start by substituting (1.28) into the Kalman gain formulae, (1.21), to find that

\[
\mathbf{K}_t = \frac{\mathbf{P}_{t-1} \mathbf{x}_t}{\mathbf{x}_t' \mathbf{P}_{t-1} \mathbf{x}_t + \mu_t^{-1} - \mathbf{x}_t' \mathbf{x}_t},
\]

(A.53)

\[
= \frac{\mu_t \mathbf{P}_{t-1} \mathbf{x}_t}{\mu_t \mathbf{x}_t' \mathbf{P}_{t-1} \mathbf{x}_t - \mu_t \mathbf{x}_t' \mathbf{x}_t + 1},
\]

(A.54)

where clearly we need \(\mathbf{P}_{t-1} = \mathbf{I}\) in order to (1.30) hold true. To achieve this, we can use (1.29) into (1.22) to find that

\[
\mathbf{P}_t = \left(\mathbf{I} - \frac{\mathbf{P}_{t-1} \mathbf{x}_t' \mathbf{x}_t'}{\mathbf{x}_t' \mathbf{P}_{t-1} \mathbf{x}_t + \sigma_t^2}\right) \mathbf{P}_{t-1} + \mathbf{I} \ldots
\]

\[
\ldots - \left(\mathbf{I} - \frac{\mathbf{P}_{t-1} \mathbf{x}_t' \mathbf{x}_t'}{\mathbf{x}_t' \mathbf{P}_{t-1} \mathbf{x}_t + \sigma_t^2}\right) \mathbf{P}_{t-1},
\]

(A.55)

\[
\mathbf{P}_t^{SG} = \mathbf{I}.
\]

(A.56)
Substituting this result into the previous formulae for the Kalman gain specialized to the SG case we find

$$K_{SG}^t = \mu_t x_t, \quad (A.57)$$

thus confirming the correspondence between the Kalman filter and the SG algorithm under the assumptions of (1.28) and (1.29).

### A.6 Kalman fixed-point smoother

Following Anderson and Moore (1979, pp. 170-6), consider replacing the state-space framework of (1.1) and (1.19) by

$$y_t = \begin{bmatrix} x_t' & 0 \end{bmatrix} \begin{bmatrix} \theta_t \\ \theta_a^t \end{bmatrix} + \varepsilon_t, \quad (A.58)$$

$$\begin{bmatrix} \theta_t \\ \theta_a^t \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{t-1} \\ \theta_a^{t-1} \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} \omega_t, \quad (A.59)$$

with the state vector at a fixed \( t = j \) satisfying

$$\begin{bmatrix} \theta_j' \\ \theta_a^j \end{bmatrix} = \begin{bmatrix} \theta_j' \\ \theta_a^j \end{bmatrix}. \quad \text{Thus, we are essentially augmenting the former system with an additional state vector which, due to the assumed "initialization" at period } j, \text{ will satisfy } \theta_a^t = \theta_j, \quad \forall t \geq j. \text{ It follows from this latter observation and the definition of conditional estimates that } \hat{\theta}_{i|t-1} = \hat{\theta}_{j|t-1}, \hat{\theta}_{i+1|t} = \hat{\theta}_{j|t}, \text{ and so on. The coefficients in the right hand side of these equalities are clearly in accordance to what we have defined as fixed-point smoothed estimates in the main text (see footnote 8), i.e., keeping } j \text{ fixed we evaluate how the coefficients estimates get updated as time goes on and new observations become available. Furthermore, the state-space system in (A.58)-(A.59) is conformable to the application of the Kalman filter, where the updating recursions for } \hat{\theta}_t = \hat{\theta}_{t+1|t} \text{ will still be given by (1.20)-(1.22), and those for } \hat{\theta}_a^t = \hat{\theta}_{t+1|t} \text{ will represent the fixed-point smooth-}$$

168
ing recursions of $\hat{\theta}_{j|t}$. These latter are found, from Anderson and Moore (1979), to be given by

$$\hat{\theta}_{j|t} = \hat{\theta}_{j|t-1} + K_t^a (y_t - x_t' \hat{\theta}_{t-1}) ,$$  \hspace{0.5cm} (A.60)

$$K_t^a = \frac{\Sigma_{t-1} x_t}{x_t' P_{t-1} x_t + \sigma_t^2},$$  \hspace{0.5cm} (A.61)

$$\Sigma_t = \Sigma_{t-1} (I - K_t x_t'),$$  \hspace{0.5cm} (A.62)

$$P_{j|t} = P_{j|t-1} - \Sigma_{t-1} x_t K_t^a',$$  \hspace{0.5cm} (A.63)

where $\Sigma_j = P_j$, and the conditional covariance matrix of the coefficients smoothed estimates errors is here given by (A.63), i.e., $P_{j|t} = E \left[ (\theta_j - \hat{\theta}_{j|t}) (\theta_j - \hat{\theta}_{j|t})' \right]$. It is also important to note the use of terms from the filtering recursions, $K_t$ and $P_{t-1}$. The smoother associated to each learning algorithm, thence, follows automatically from what the different assumptions on (1.24)-(1.25) and (1.28)-(1.29) imply for these recursions.

A.7 Adaptive gain algorithms

To derive the LSA and the SGA expressions in (3.2)-(3.6) and (3.7)-(3.9), we start by defining the gains adaptation recursions, as given by

$$\gamma_t = \gamma_{t-1} - \alpha_\gamma \hat{\nabla}_t^\gamma,$$  \hspace{0.5cm} (A.64)

$$\mu_t = \mu_{t-1} - \alpha_\mu \hat{\nabla}_t^\mu,$$  \hspace{0.5cm} (A.65)

where $\alpha_\gamma$ and $\alpha_\mu$ represent small adaptation constants, and $\hat{\nabla}_t^\gamma$ and $\hat{\nabla}_t^\mu$ stand for estimates of the gradients $\nabla_t^\gamma = \frac{\partial J_t}{\partial \gamma}$ and $\nabla_t^\mu = \frac{\partial J_t}{\partial \mu}$, for the LS and the SG, respectively. The key step then is to find the relevant gradients and plug their stochastic approximation in the above recursions.

We first derive the LSA. Taking the first derivative of $J_t$ with respect to $\gamma$
we obtain the gradient
\[ \nabla_{\gamma_i} \ell = \frac{\partial J_{LSA}^t}{\partial \gamma} = -E \left[ x_i' \frac{\partial \hat{\theta}_{LSA}^{t-1}}{\partial \gamma} \left( z_t - x_i' \hat{\theta}_{LSA}^{t-1} \right) \right], \quad (A.66) \]
which is stochastically approximated as
\[ \hat{\nabla}_{\gamma_i} = -x_i' \hat{\Psi}_{LSA}^{t-1} \left( z_t - x_i' \hat{\theta}_{LSA}^{t-1} \right), \quad (A.67) \]
where \( \hat{\Psi}_{LSA}^{t} \) stands for a recursive estimate of \( \frac{\partial \hat{\theta}_{LSA}^{t}}{\partial \gamma} \). Differentiating (1.2) and (1.3) we obtain
\[ \frac{\partial \hat{\theta}_{LSA}^{t}}{\partial \gamma} = \frac{\partial \hat{\theta}_{LSA}^{t-1}}{\partial \gamma} + R_t^{-1} x_t \left( y_t - x_i' \hat{\theta}_{LSA}^{t-1} \right) \ldots \]
\[ -\gamma_t R_t^{-1} \frac{\partial R_t}{\partial \gamma} R_t^{-1} x_t \left( y_t - x_i' \hat{\theta}_{LSA}^{t-1} \right) - \gamma_t R_t^{-1} x_t x_t' \frac{\partial \hat{\theta}_{LSA}^{t-1}}{\partial \gamma}, \quad (A.68) \]
\[ \frac{\partial R_t}{\partial \gamma} = \frac{\partial R_t^{-1}}{\partial \gamma} + x_t x_t' - R_t^{-1} - \gamma_t \frac{\partial R_t^{-1}}{\partial \gamma}. \quad (A.69) \]
Letting \( \hat{S}_t \) stand for the recursive estimate of \( \frac{\partial R_t}{\partial \gamma} \), and subsituting (A.67), (A.68), and (A.69) into (A.64) we obtain the LSA algorithm.

We follow similar steps for the SGA. Particularly, as estimate of the loss gradient is given by
\[ \hat{\nabla}_{\mu_i} = -x_i' \hat{\Psi}_{SGA}^{t-1} \left( z_t - x_i' \hat{\theta}_{SGA}^{t-1} \right), \quad (A.70) \]
with \( \hat{\Psi}_{SGA}^{t} = \frac{\partial \hat{\theta}_{SGA}^{t}}{\partial \mu} \) obtained differentiating (1.4),
\[ \frac{\partial \hat{\theta}_{SGA}^{t}}{\partial \mu} = \frac{\partial \hat{\theta}_{SGA}^{t-1}}{\partial \mu} + x_t \left( y_t - x_i' \hat{\theta}_{SGA}^{t-1} \right) - \mu_t x_t x_t' \frac{\partial \hat{\theta}_{SGA}^{t-1}}{\partial \mu}, \quad (A.71) \]
which together with (A.65), reduces to the SGA algorithm.
A.8 Adaptive weight paired algorithms

Derivations similar to those used to obtain the adaptive gain calibrations can be adopted to obtain the Hybrid algorithm with adaptive mixing weights (HYA) and the Mixing algorithm with adaptive weights (MXA). The idea here is to take the learning gains as pre-determined and couple an outer mechanism to adjust the mixing weights adaptively in response estimates of the gradient of the loss function. Let's start defining the weights adaptation recursions, as given by

\[ \omega_t = \omega_{t-1} - \alpha_\omega \hat{\nabla}_t^\omega, \tag{A.72} \]

\[ w_t = w_{t-1} - \alpha_w \hat{\nabla}_t^w, \tag{A.73} \]

where \( \alpha_\omega \) and \( \alpha_w \) represent small adaptation constants, and \( \hat{\nabla}_t^\omega \) and \( \hat{\nabla}_t^w \) stand for estimates of the gradients \( \nabla_t^\omega = \frac{\partial J_t}{\partial \omega} \) and \( \nabla_t^w = \frac{\partial J_t}{\partial w} \), for the HY and the MX, respectively. These gradients can be stochastically approximated as

\[ \hat{\nabla}_t^\omega = -x_t' \hat{\Psi}_t^{HYA} (z_t - x_t' \hat{\theta}_t^{HYA}) - \hat{\Psi}_t^{HYA} x_t (y_t - x_t' \hat{\theta}_t^{HYA}) \ldots \]

\[ -H_t x_t x_t' \hat{\Psi}_t^{HYA} \frac{\partial \hat{\theta}_t^{HYA}}{\partial \omega}, \tag{A.74} \]

\[ \hat{\nabla}_t^w = -x_t' \hat{\Psi}_t^{MXA} (z_t - x_t' \hat{\theta}_t^{MXA}), \tag{A.75} \]

where \( \hat{\Psi}_t^{HYA} \) and \( \hat{\Psi}_t^{MXA} \) stand for recursive estimates of \( \frac{\partial \hat{\theta}_t^{HYA}}{\partial \omega} \) and \( \frac{\partial \hat{\theta}_t^{MXA}}{\partial w} \)

Differentiating (4.1)-(4.2) and (4.3) we obtain

\[ \frac{\partial \hat{\theta}_t^{HYA}}{\partial \omega} = \frac{\partial \hat{\theta}_t^{HYA}}{\partial \hat{\theta}_t^{HYA}} \left( \frac{\partial \hat{\theta}_t^{HYA}}{\partial \omega} + \left( g_t R_t^{-1} - \mu_t I \right) x_t (y_t - x_t' \hat{\theta}_t^{HYA}) \ldots \right) 
- H_t x_t x_t' \frac{\partial \hat{\theta}_t^{HYA}}{\partial \omega}, \tag{A.76} \]

\[ \frac{\partial \hat{\theta}_t^{MXA}}{\partial w} = \hat{\theta}_t^{LS} - \hat{\theta}_t^{SG}. \tag{A.77} \]

Substituting (A.74) and (A.76) into (A.72), and (A.75) and (A.77) into (A.73), we obtain the HYA and the MXA algorithms.
Algorithm 7 (HYA). Under the estimation context of (1.1), the HYA algorithm assumes the form of

\[
\hat{\theta}^{HYA}_t = \hat{\theta}^{HYA}_{t-1} + H_t x_t \left( y_t - x_t' \hat{\theta}^{HYA}_{t-1} \right), \tag{A.78}
\]

\[
H_t = \omega_t \gamma_t R^{-1}_t + (1 - \omega_t) \mu_t I, \tag{A.79}
\]

\[
\omega_t = \omega_{t-1} + \alpha_\omega x_t' \hat{\Psi}^{HYA}_{t-1} \left( z_t - x_t' \hat{\theta}^{HYA}_{t-1} \right), \tag{A.80}
\]

\[
\hat{\Psi}^{HYA}_t = \left( I - H_t x_t x_t' \right) \hat{\Psi}^{HYA}_{t-1} + \left( \gamma_t R^{-1}_t - \mu_t I \right) x_t \left( y_t - x_t' \hat{\theta}^{HYA}_{t-1} \right), \tag{A.81}
\]

where \(\alpha_\omega\) is an adaptation constant, \(\hat{\Psi}^{HYA}_t\) stands for an estimate of \(\partial \hat{\theta}^{HYA}_t / \partial \omega\), and the remaining components follow from the definitions of the Hybrid algorithm.

Algorithm 8 (MXA). Under the estimation context of (1.1), the MXA algorithm assumes the form of

\[
\hat{\theta}^{MXA}_t = w_t \hat{\theta}^{LS}_t + (1 - w_t) \hat{\theta}^{SG}_t, \tag{A.82}
\]

\[
w_t = w_{t-1} + \alpha_w \left( x_t' \hat{\theta}^{LS}_t - x_t' \hat{\theta}^{SG}_t \right) \left( z_t - x_t' \hat{\theta}^{MXA}_{t-1} \right), \tag{A.83}
\]

where \(\alpha_w\) is an adaptation constant, and the remaining components follow from the definitions of the previous algorithms.

It is interesting to note that (A.83) represents an intuitive update rule to the mixing weights: if the latest forecasting error observed using the Mixing algorithm was positive (negative), meaning that the forecast was below (above) the actual observation, the outer mechanism increases the weight of the algorithm that currently forecasts higher (lowest).
Appendix B

Details on data and statistical tests

B.1 Details on data

Short time series history: some vintages lack of earlier observations due to delays into BEA revisions (see Philadelphia’s Fed documentations). This was the case of the vintages of 1992q1-1992q4 (missing data from 1947-1958), 1996q1-1997q1 (missing data from 1947-1959q2), and 1999q4-2000q1 (missing data from 1947-1958). We circumvent this problem (to turn the dataset vintages-balanced) by reproducing observations from the last available vintage while rescaling in accordance to the ratio between the first observation available in the missing observation vintage and the value observed for the same period in the vintage being used as source for the missing observations.

Missing observation for 1995q4 in vintage 1996q1: as a result of the US federal government shutdown in late 1995, the observation for 1995q4 was missing in the 1996q1 vintage. Fortunately, this is the only point in this dataset that this happens. We fulfill this gap by using the observation available in the March 1996 monthly vintage for the same series. Incidentally, the SPF 1996q1 median backcast for 1995q4 is identical to the value later observed in March 1996, thence, our simplifying procedure is not favoring any method.

Caveat on SPF’s forecasts for Real GDP: forecasts for real GDP were not asked in
the surveys prior to 1981q3. To extend this series of forecast back to 1968q4, real GDP prior to 1981q3 is computed by using the formula (nominal GDP / GDP prices) * 100.

B.2 Review of statistical tests for equal predictive ability

We want to determine whether two series of forecasts are statistically different from each other. Let $f_{1,t,h}$ and $f_{2,t,h}$ stand for these forecasts, where $h$ (going from 0 to 4 in our case) denotes the horizon at which these forecasts were made, and $y_t$ stand for the series of targets of these forecasts. Under the notation of section 1.4.1, the losses associated to each of these forecasts are given by $\mathcal{L}\left(f_{1,t,h}, y_t\right)$ and $\mathcal{L}\left(f_{2,t,h}, y_t\right)$. Letting $d_{t,h} = \mathcal{L}\left(f_{1,t,h}, y_t\right) - \mathcal{L}\left(f_{2,t,h}, y_t\right)$ denote the series of loss differentials between the two forecasts at horizon $h$, the Diebold and Mariano (1995) test evaluates whether their average loss differences,

$$d_{h} = \frac{1}{T} \sum_{i=1}^{T} d_{i,h},$$

is significantly different from zero. Under the null hypothesis of equal predictive ability the DM statistic,

$$DM_{h} = \frac{d_{h}}{\sqrt{\hat{\sigma}^2_{d} / T}},$$

has a $t$-distribution with $T - 1$ degrees of freedom, where $\hat{\sigma}^2_{d}$ is an estimate of the long-run variance of $d_{i,h}$. For the estimation of $\hat{\sigma}^2_{d}$ we adopt the heteroskedasticity and autocorrelation consistent (HAC) estimator proposed by Newey and West (1987).

The Giacomini and White (2006) test, in contrast, evaluates the null hypothesis of equal conditional predictive ability. The main caveat on this test relates to the specification of a test function, $q_{i,h}$ containing $q$ instruments, which attempts to control for the informational conditioning required by the
null hypothesis. To test the conditional moment restriction $E[q_{t,h}d_{t,h}] = 0$, a Wald-type test statistic is proposed having the form of

$$GW_h = T \left( T^{-1} \sum_{i=1}^T q_{i,h}d_{i,h} \right)' \hat{\Omega}_h^{-1} \left( T^{-1} \sum_{i=1}^T q_{i,h}d_{i,h} \right),$$

(B.3)

where $\hat{\Omega}_h$ is a $q \times q$ consistent estimate of the covariance matrix of $q_{t,h}d_{t,h}$. Under the null hypothesis of equal conditional predictive ability $GW_h$ has a $\chi^2_q$ distribution.

Apart from the first horizon, $\hat{\Omega}_h$ is again estimated using the HAC estimator of Newey and West (1987), with $h$ determining the truncated kernel bandwidth. For the case of the first horizon, Giacomini and White (2006) simplify the computation of (B.3) to be given by $TR^2$, where $R^2$ is the uncentered squared multiple correlation coefficient obtained by regressing a constant unity on $q_{t,h}d_{t,h}$. Finally, regarding the specification of $q_{t,h}$, in the lack of better alternatives, the recommendation is for the use of $h$-lagged loss differentials. Thus, in our calculations we set $q_{t,h} = d_{t-h,h}$. 

175
Appendix C

Supplementary results

In this appendix we present results supplementary to those of chapter 2 (Tables C.1-C.4, and Figures C.1-C.3), chapter 3 (Table C.5), and chapter 4 (Tables C.6-C.8, and Figure C.4). For succinctness we do not discuss these results as their interpretation follows from the main text.
Table C.1: Calibration of parameters for simulation with growth-like data.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Calibration values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) For artificial series:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma^2_e$</td>
<td>Variance of $e_t$ in (2.5).</td>
<td>13.00</td>
</tr>
<tr>
<td>$\sigma^2_w$</td>
<td>Variance of $w_{t+1}$ in (2.6).</td>
<td>$7 \times 10^{-5}$</td>
</tr>
<tr>
<td>$\bar{\theta}$</td>
<td>Steady-state value of $\theta_t$.</td>
<td>0.40</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Persistence of deviations from $\bar{\theta}$.</td>
<td>0.999</td>
</tr>
<tr>
<td>(b) For algorithms:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\bar{\gamma}_1$</td>
<td>LS “low” constant learning gains.</td>
<td>0.02</td>
</tr>
<tr>
<td>$\bar{\gamma}_2$</td>
<td>LS “high” constant learning gains.</td>
<td>0.10</td>
</tr>
<tr>
<td>$\bar{\beta}_1$</td>
<td>SG “low” constant learning gain.</td>
<td>0.0017</td>
</tr>
<tr>
<td>$\bar{\beta}_2$</td>
<td>SG “high” constant learning gain.</td>
<td>0.0334</td>
</tr>
<tr>
<td>(c) For initialization methods:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td>Size of initial sample of training data.</td>
<td>75</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Convergence tolerance for smoothed initials.</td>
<td>0.01</td>
</tr>
<tr>
<td>$\bar{S}$</td>
<td>Maximum number of smoothing repetitions.</td>
<td>100</td>
</tr>
</tbody>
</table>

See footnotes to Table 2.2.

Figure C.1: Example of growth-like artificially generated series.

(a) Autoregressive coefficient time series.  
(b) Observations time series.

See footnotes to Figure 2.2.
Figure C.2: MSD learning curves for growth-like artificial data.

(a) Least Squares.

(b) Stochastic Gradient.

See footnotes to Figure 2.3.
Table C.2: Average MSDs after initializations on growth-like data.

(a) Least Squares.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>76-100</td>
<td>101-150</td>
</tr>
<tr>
<td>$\gamma_1 = 0.02$</td>
<td>Diffuse-track</td>
<td>0.0132</td>
<td>0.0113</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[10.5]</td>
<td>[5.0]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0127</td>
<td>0.0099</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[9.2]</td>
<td>[1.3]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0123</td>
<td>0.0100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[7.8]</td>
<td>[1.5]</td>
</tr>
<tr>
<td>$\gamma_2 = 0.10$</td>
<td>Diffuse-track</td>
<td>0.0387</td>
<td>0.0390</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.2]</td>
<td>[0.0]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0145</td>
<td>0.0328</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-13.4]</td>
<td>[-3.4]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0385</td>
<td>0.0389</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.2]</td>
<td>[0.0]</td>
</tr>
</tbody>
</table>

See footnotes to Table 2.3.

(b) Stochastic Gradient.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>76-100</td>
<td>101-150</td>
</tr>
<tr>
<td>$\mu_1 = 0.001$</td>
<td>Diffuse-track</td>
<td>0.1105</td>
<td>0.0920</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[269.8]</td>
<td>[220.0]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0791</td>
<td>0.0666</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[191.4]</td>
<td>[156.7]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0162</td>
<td>0.0157</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[17.3]</td>
<td>[15.7]</td>
</tr>
<tr>
<td>$\mu_2 = 0.0205$</td>
<td>Diffuse-track</td>
<td>0.0214</td>
<td>0.0215</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-0.7]</td>
<td>[-0.7]</td>
</tr>
<tr>
<td></td>
<td>Diffuse-ordinary</td>
<td>0.0482</td>
<td>0.0249</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[28.5]</td>
<td>[3.0]</td>
</tr>
<tr>
<td></td>
<td>Smoothing</td>
<td>0.0192</td>
<td>0.0194</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-1.1]</td>
<td>[-0.9]</td>
</tr>
</tbody>
</table>
Table C.3: Average MSEs after initializations on growth-like data.

(a) Least Squares.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>76-100</td>
<td>101-150</td>
<td>151-200</td>
</tr>
<tr>
<td>( \gamma_1 = 0.02 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[-0.1]</td>
<td>[0.1]</td>
<td>[-0.1]</td>
</tr>
<tr>
<td></td>
<td>[-0.1]</td>
<td>[0.0]</td>
<td>[-0.1]</td>
</tr>
<tr>
<td></td>
<td>[-0.1]</td>
<td>[0.0]</td>
<td>[-0.1]</td>
</tr>
<tr>
<td>( \gamma_2 = 0.10 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[-0.2]</td>
<td>[-0.0]</td>
<td>[-0.1]</td>
</tr>
<tr>
<td></td>
<td>[-1.2]</td>
<td>[-0.2]</td>
<td>[-0.1]</td>
</tr>
<tr>
<td></td>
<td>[-0.2]</td>
<td>[-0.0]</td>
<td>[-0.1]</td>
</tr>
</tbody>
</table>

(b) Stochastic Gradient.

<table>
<thead>
<tr>
<th>Gains</th>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>Steady state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>76-100</td>
<td>101-150</td>
<td>151-200</td>
</tr>
<tr>
<td>( \beta_1 = 0.001 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[3.0]</td>
<td>[2.6]</td>
<td>[1.6]</td>
</tr>
<tr>
<td></td>
<td>[2.0]</td>
<td>[1.8]</td>
<td>[1.0]</td>
</tr>
<tr>
<td></td>
<td>[-0.0]</td>
<td>[0.2]</td>
<td>[-0.0]</td>
</tr>
<tr>
<td>( \beta_2 = 0.0205 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[-0.3]</td>
<td>[-0.1]</td>
<td>[-0.2]</td>
</tr>
<tr>
<td></td>
<td>[0.3]</td>
<td>[-0.0]</td>
<td>[-0.2]</td>
</tr>
<tr>
<td></td>
<td>[-0.3]</td>
<td>[-0.0]</td>
<td>[-0.2]</td>
</tr>
</tbody>
</table>

See footnotes to Table 2.4.
Figure C.3: Estimates on growth equation coefficients.

(a) Least Squares.

(b) Stochastic Gradient.

See footnotes to Figure 2.4.
The series of adaptive weights were obtained using adaptation constants given by $\alpha_w = \alpha_{\omega} = 0.1$ on both inflation and growth variables.
Table C.4: Statistics on growth forecasts.

<table>
<thead>
<tr>
<th>Initializations</th>
<th>Samples after initializations</th>
<th>76-100</th>
<th>101-150</th>
<th>151-200</th>
<th>201-250</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) Least Squares estimates with $\bar{\gamma} = 0.025$:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diffuse-track</td>
<td></td>
<td>1.73</td>
<td>2.78</td>
<td>0.45</td>
<td>2.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(15.63)</td>
<td>(21.20)</td>
<td>(3.73)</td>
<td>(7.50)</td>
</tr>
<tr>
<td>Diffuse-ordinary</td>
<td></td>
<td>1.47</td>
<td>2.27</td>
<td>0.44</td>
<td>2.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(14.72)</td>
<td>(21.46)</td>
<td>(3.73)</td>
<td>(7.49)</td>
</tr>
<tr>
<td>Smoothing</td>
<td></td>
<td>1.82</td>
<td>2.82</td>
<td>0.46</td>
<td>2.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(15.48)</td>
<td>(21.11)</td>
<td>(3.73)</td>
<td>(7.49)</td>
</tr>
<tr>
<td>(b) Stochastic Gradient estimates with $\bar{\mu} = 0.0046$:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diffuse-track</td>
<td></td>
<td>1.50</td>
<td>2.60</td>
<td>0.95</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(15.67)</td>
<td>(22.60)</td>
<td>(3.93)</td>
<td>(7.86)</td>
</tr>
<tr>
<td>Diffuse-ordinary</td>
<td></td>
<td>0.94</td>
<td>2.53</td>
<td>0.95</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(14.73)</td>
<td>(22.88)</td>
<td>(3.92)</td>
<td>(7.86)</td>
</tr>
<tr>
<td>Smoothing</td>
<td></td>
<td>1.50</td>
<td>2.60</td>
<td>0.95</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(15.64)</td>
<td>(22.62)</td>
<td>(3.93)</td>
<td>(7.86)</td>
</tr>
</tbody>
</table>

See footnotes to Table 2.5.
Table C.5: Illustrative excerpt of output of comparative evaluation.

<table>
<thead>
<tr>
<th></th>
<th>DM</th>
<th>GW</th>
<th>DM</th>
<th>GW</th>
<th>DM</th>
<th>GW</th>
<th>DM</th>
<th>GW</th>
<th>DM</th>
<th>GW</th>
<th>DM</th>
<th>GW</th>
<th>DM</th>
<th>GW</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSiC</td>
<td>-0.51</td>
<td>2.46</td>
<td>-1.65</td>
<td>0.48</td>
<td>-0.63</td>
<td>1.42</td>
<td>-0.69</td>
<td>1.74</td>
<td>0.08</td>
<td>0.39</td>
<td>-1.27</td>
<td>1.75</td>
<td>-0.78</td>
<td>0.00</td>
</tr>
</tbody>
</table>
|          | (0.61)| (0.12)| (0.10)| (0.49)| (0.53)| (0.23)| (0.49)| (0.19)| (0.94)| (0.53)| (0.21)| (0.19)| (0.44)| (0.98)| ...
| LSiC     | 1.17 | 1.34 | -1.39| 0.01 | -0.64| 1.20 | -0.72| 1.53 | 0.33 | 0.27 | -1.13| 1.94 | -0.76| 0.03 |
|          | (0.24)| (0.25)| (0.17)| (0.94)| (0.52)| (0.27)| (0.47)| (0.22)| (0.75)| (0.60)| (0.26)| (0.16)| (0.45)| (0.87)| ...
| LSrC     | -0.00| 0.54 | -1.04| 0.97 | -0.16| 1.19 | -0.09| 1.19 | 1.17| 1.14 | -0.90| 0.76 | -0.46| 0.02 |
|          | (1.00)| (0.46)| (0.30)| (0.32)| (0.87)| (0.27)| (0.93)| (0.27)| (0.24)| (0.29)| (0.37)| (0.38)| (0.65)| (0.90)| ...
| LSpP     | -1.72| 2.23 | -1.56| 2.27 | -1.29| 1.22 | 0.38 | 0.28 | 0.77| 2.97 | -0.58| 3.40 | -0.40| 0.26 |
|          | (0.09)| (0.14)| (0.12)| (0.13)| (0.20)| (0.27)| (0.70)| (0.60)| (0.44)| (0.08)| (0.56)| (0.07)| (0.69)| (0.61)| ...
| LSpP     | —    | —    | -1.17| 1.34 | 0.00 | 0.54 | 1.72 | 2.23 | 0.85| 3.04 | -0.67| 3.08 | -0.48| 0.46 |
|          | (—)| (—)| (0.24)| (0.25)| (1.00)| (0.46)| (0.09)| (0.14)| (0.40)| (0.08)| (0.51)| (0.08)| (0.63)| (0.50)| ...
| LSpP     | -1.28| 0.90 | -1.59| 0.49 | 0.65 | 1.09 | -1.28| 0.90 | -1.11| 1.33 | -0.81| 1.25 | -0.81| 1.25 |
|          | (0.20)| (0.34)| (0.75)| (0.11)| (0.48)| (0.51)| (0.30)| (0.20)| (0.34)| (0.27)| (0.25)| (0.42)| (0.26)| (0.42)| (0.26)| ...
| LSaC     | 1.77 | 1.20 | 1.80 | 1.32 | 1.84 | 0.96 | 1.79 | 1.10 | 1.77| 1.20 | 1.83| 0.73 | 0.44 | 4.12 |
|          | (0.08)| (0.27)| (0.07)| (0.25)| (0.07)| (0.33)| (0.08)| (0.29)| (0.08)| (0.27)| (0.07)| (0.39)| (0.66)| (0.04)| (0.66)| ...
| SGfC     | -0.17| 1.77| -0.41| 2.66| -0.16| 1.90| 0.75| 0.17| -0.17| 1.77| 0.40| 2.55| -1.36| 1.03 |
|          | (0.87)| (0.18)| (0.68)| (0.10)| (0.87)| (0.17)| (0.45)| (0.68)| (0.87)| (0.18)| (0.69)| (0.11)| (0.18)| (0.31)| ...
| SGfC     | -0.44| 1.62| -0.29| 2.68| -0.04| 1.72| 0.86| 0.06| -0.04| 1.62| 0.53| 2.42| -1.31| 1.02 |
|          | (0.97)| (0.20)| (0.77)| (0.10)| (0.97)| (0.19)| (0.39)| (0.80)| (0.97)| (0.20)| (0.59)| (0.12)| (0.19)| (0.31)| (0.01)| ...

This excerpt refers to the accuracy comparisons of forecasts for $h = 0$ using the VAR (1) model. Our hit rates analysis is based on a set of 40 of such tables: 5 horizons x 4 VARs x 2 exercises (accuracy and resemblance). Results for inflation are presented above the main diagonal, whereas those for growth are below the main diagonal. Each specification is abbreviated into four characters as follows: the first two indicate the algorithm (LS or SG); the third character indicates the sample/method used to select the gain (“f” for full-sample, “i” for in-sample, “r” for recursive, or “a” for adaptive); the last character indicates the gain calibration assumption (“C” for choice, or “P” for primitive). DM and GW stand for the Diebold and Mariano (1995) and the Giacomini and White (2006) tests, respectively. Values in parentheses refer to the p-values of the test statistics above them. From its definition, the signal of the DM statistic indicates which specification presented a smaller averaged loss: positive (negative) values indicate the specification of the column (row) header outperformed that of the row (column).
Table C.6: Data and forecasts statistics for paired algorithms.

(a) Inflation.

<table>
<thead>
<tr>
<th>Series/Algorithm</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
<th>Var</th>
<th>AR(1)</th>
<th>CorA</th>
<th>CorS</th>
<th>MSFE</th>
<th>MSFCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuals</td>
<td>2.67</td>
<td>-0.33</td>
<td>9.39</td>
<td>2.68</td>
<td>0.63</td>
<td>1.00</td>
<td>0.81</td>
<td>0.00</td>
<td>1.03</td>
</tr>
<tr>
<td>Surveys</td>
<td>2.93</td>
<td>0.62</td>
<td>9.49</td>
<td>2.46</td>
<td>0.94</td>
<td>0.81</td>
<td>1.00</td>
<td>1.03</td>
<td>0.00</td>
</tr>
<tr>
<td>Hybrid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Fixed /choice</td>
<td>3.01</td>
<td>-0.04</td>
<td>10.60</td>
<td>2.84</td>
<td>0.73</td>
<td>0.91</td>
<td>1.60</td>
<td>0.49</td>
<td></td>
</tr>
<tr>
<td>-Time-var. /choice</td>
<td>2.97</td>
<td>-0.83</td>
<td>10.35</td>
<td>3.01</td>
<td>0.71</td>
<td>0.91</td>
<td>1.73</td>
<td>0.53</td>
<td></td>
</tr>
<tr>
<td>-Fixed /primit.</td>
<td>2.98</td>
<td>0.04</td>
<td>10.96</td>
<td>2.68</td>
<td>0.70</td>
<td>0.91</td>
<td>1.70</td>
<td>0.49</td>
<td></td>
</tr>
<tr>
<td>-Time-var./primit.</td>
<td>3.04</td>
<td>0.03</td>
<td>10.69</td>
<td>2.61</td>
<td>0.72</td>
<td>0.91</td>
<td>1.62</td>
<td>0.49</td>
<td></td>
</tr>
<tr>
<td>Mixing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Fixed /choice</td>
<td>2.96</td>
<td>-0.10</td>
<td>10.53</td>
<td>2.87</td>
<td>0.72</td>
<td>0.90</td>
<td>1.61</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>-Time-var. /choice</td>
<td>2.94</td>
<td>-0.64</td>
<td>10.35</td>
<td>3.04</td>
<td>0.71</td>
<td>0.91</td>
<td>1.70</td>
<td>0.53</td>
<td></td>
</tr>
<tr>
<td>-Fixed /primit.</td>
<td>2.94</td>
<td>-0.27</td>
<td>10.54</td>
<td>2.73</td>
<td>0.70</td>
<td>0.90</td>
<td>1.70</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>-Time-var./primit.</td>
<td>3.01</td>
<td>0.01</td>
<td>10.55</td>
<td>2.60</td>
<td>0.72</td>
<td>0.90</td>
<td>1.57</td>
<td>0.53</td>
<td></td>
</tr>
</tbody>
</table>

(b) Growth.

<table>
<thead>
<tr>
<th>Series/Algorithm</th>
<th>Mean</th>
<th>Min</th>
<th>Max</th>
<th>Var</th>
<th>AR(1)</th>
<th>CorA</th>
<th>CorS</th>
<th>MSFE</th>
<th>MSFCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuals</td>
<td>2.54</td>
<td>6.14</td>
<td>8.67</td>
<td>6.17</td>
<td>0.50</td>
<td>1.00</td>
<td>0.78</td>
<td>0.00</td>
<td>2.56</td>
</tr>
<tr>
<td>Surveys</td>
<td>2.24</td>
<td>5.19</td>
<td>7.01</td>
<td>3.14</td>
<td>0.72</td>
<td>0.78</td>
<td>1.00</td>
<td>2.56</td>
<td>0.00</td>
</tr>
<tr>
<td>Hybrid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Fixed /choice</td>
<td>3.12</td>
<td>-2.58</td>
<td>6.29</td>
<td>1.27</td>
<td>0.59</td>
<td>0.42</td>
<td>0.61</td>
<td>5.40</td>
<td>2.72</td>
</tr>
<tr>
<td>-Time-var. /choice</td>
<td>2.77</td>
<td>-0.95</td>
<td>6.17</td>
<td>1.29</td>
<td>0.36</td>
<td>0.59</td>
<td>5.46</td>
<td>2.31</td>
<td></td>
</tr>
<tr>
<td>-Fixed /primit.</td>
<td>2.50</td>
<td>-1.01</td>
<td>4.85</td>
<td>1.01</td>
<td>0.46</td>
<td>0.62</td>
<td>4.83</td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td>-Time-var./primit.</td>
<td>2.57</td>
<td>-1.47</td>
<td>5.41</td>
<td>1.20</td>
<td>0.41</td>
<td>0.64</td>
<td>5.13</td>
<td>1.95</td>
<td></td>
</tr>
<tr>
<td>Mixing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Fixed /choice</td>
<td>2.74</td>
<td>-3.26</td>
<td>5.95</td>
<td>1.35</td>
<td>0.60</td>
<td>0.42</td>
<td>0.62</td>
<td>5.10</td>
<td>2.17</td>
</tr>
<tr>
<td>-Time-var. /choice</td>
<td>2.79</td>
<td>-1.02</td>
<td>6.38</td>
<td>1.25</td>
<td>0.56</td>
<td>0.36</td>
<td>0.60</td>
<td>5.47</td>
<td>2.30</td>
</tr>
<tr>
<td>-Fixed /primit.</td>
<td>2.50</td>
<td>-2.07</td>
<td>5.02</td>
<td>1.10</td>
<td>0.54</td>
<td>0.46</td>
<td>0.64</td>
<td>4.82</td>
<td>1.92</td>
</tr>
<tr>
<td>-Time-var./primit.</td>
<td>2.62</td>
<td>-1.56</td>
<td>5.56</td>
<td>1.19</td>
<td>0.54</td>
<td>0.40</td>
<td>0.64</td>
<td>5.15</td>
<td>1.98</td>
</tr>
</tbody>
</table>

See footnotes to Table 1.1. The algorithms’ forecasts refer to those from the VAR(1).
Table C.7: Hit rates on paired algorithms gain/weight selection methods.

(a) Forecast accuracy.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Full-sample wins</th>
<th>Recursive wins</th>
<th>Adaptive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>GW-20%</td>
<td>Hit rate</td>
</tr>
<tr>
<td>Inflation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/choice</td>
<td>30%</td>
<td>5%</td>
<td>70%</td>
</tr>
<tr>
<td>- HY/primitive</td>
<td>55%</td>
<td>20%</td>
<td>45%</td>
</tr>
<tr>
<td>- MX/choice</td>
<td>60%</td>
<td>15%</td>
<td>40%</td>
</tr>
<tr>
<td>- MX/primitive</td>
<td>70%</td>
<td>25%</td>
<td>25%</td>
</tr>
<tr>
<td>Growth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/choice</td>
<td>100%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>- HY/primitive</td>
<td>75%</td>
<td>15%</td>
<td>25%</td>
</tr>
<tr>
<td>- MX/choice</td>
<td>100%</td>
<td>20%</td>
<td>0%</td>
</tr>
<tr>
<td>- MX/primitive</td>
<td>75%</td>
<td>15%</td>
<td>25%</td>
</tr>
</tbody>
</table>

(b) Forecast resemblance.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Full-sample wins</th>
<th>Recursive wins</th>
<th>Adaptive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>GW-20%</td>
<td>Hit rate</td>
</tr>
<tr>
<td>Inflation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/choice</td>
<td>55%</td>
<td>25%</td>
<td>45%</td>
</tr>
<tr>
<td>- HY/primitive</td>
<td>60%</td>
<td>0%</td>
<td>40%</td>
</tr>
<tr>
<td>- MX/choice</td>
<td>30%</td>
<td>20%</td>
<td>70%</td>
</tr>
<tr>
<td>- MX/primitive</td>
<td>20%</td>
<td>0%</td>
<td>70%</td>
</tr>
<tr>
<td>Growth</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/choice</td>
<td>70%</td>
<td>55%</td>
<td>30%</td>
</tr>
<tr>
<td>- HY/primitive</td>
<td>75%</td>
<td>75%</td>
<td>25%</td>
</tr>
<tr>
<td>- MX/choice</td>
<td>55%</td>
<td>25%</td>
<td>45%</td>
</tr>
<tr>
<td>- MX/primitive</td>
<td>75%</td>
<td>50%</td>
<td>25%</td>
</tr>
</tbody>
</table>

See footnotes to Table 3.3.
Table C.8: Hit rates on paired algorithms gain/weight determination.

(a) Forecast accuracy.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Gain/weight as a choice wins</th>
<th>Gain/weight as a primitive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>Inflation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/Full-sample</td>
<td>30%</td>
<td>0%</td>
</tr>
<tr>
<td>- HY/Recursive</td>
<td>55%</td>
<td>5%</td>
</tr>
<tr>
<td>- HY/Adaptive</td>
<td>25%</td>
<td>15%</td>
</tr>
<tr>
<td>- MX/Full-sample</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>- MX/Recursive</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>- MX/Adaptive</td>
<td>30%</td>
<td>0%</td>
</tr>
<tr>
<td>Growth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/Full-sample</td>
<td>75%</td>
<td>10%</td>
</tr>
<tr>
<td>- HY/Recursive</td>
<td>35%</td>
<td>0%</td>
</tr>
<tr>
<td>- HY/Adaptive</td>
<td>65%</td>
<td>0%</td>
</tr>
<tr>
<td>- MX/Full-sample</td>
<td>65%</td>
<td>15%</td>
</tr>
<tr>
<td>- MX/Recursive</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>- MX/Adaptive</td>
<td>55%</td>
<td>15%</td>
</tr>
</tbody>
</table>

See footnotes to Table 3.4.

(b) Forecast resemblance.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Gain as a choice wins</th>
<th>Gain as a primitive wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hit rate</td>
<td>DM-20%</td>
</tr>
<tr>
<td>Inflation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/Full-sample</td>
<td>20%</td>
<td>5%</td>
</tr>
<tr>
<td>- HY/Recursive</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>- HY/Adaptive</td>
<td>30%</td>
<td>10%</td>
</tr>
<tr>
<td>- MX/Full-sample</td>
<td>25%</td>
<td>5%</td>
</tr>
<tr>
<td>- MX/Recursive</td>
<td>15%</td>
<td>5%</td>
</tr>
<tr>
<td>- MX/Adaptive</td>
<td>20%</td>
<td>0%</td>
</tr>
<tr>
<td>Growth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- HY/Full-sample</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>- HY/Recursive</td>
<td>35%</td>
<td>5%</td>
</tr>
<tr>
<td>- HY/Adaptive</td>
<td>60%</td>
<td>0%</td>
</tr>
<tr>
<td>- MX/Full-sample</td>
<td>15%</td>
<td>0%</td>
</tr>
<tr>
<td>- MX/Recursive</td>
<td>40%</td>
<td>10%</td>
</tr>
<tr>
<td>- MX/Adaptive</td>
<td>50%</td>
<td>10%</td>
</tr>
</tbody>
</table>

See footnotes to Table 3.4.
Bibliography


