Agent-Based Computational Economics

Overview of the
Santa Fe Artificial Stock Market Model

Leigh Tesfatsion
Professor of Economics
Courtesy Professor of Mathematics &
Electrical and Computer Engineering (ECpE)
Iowa State University
Ames, IA 50011-1070

15 April 2009
Basic References (See the Econ 308 Syllabus for Links)

https://www2.econ.iastate.edu/tesfatsi/syl308.htm

Ref.[1] ** L. Tesfatsion, "Stock Market Basics"

Ref.[2] ** L. Tesfatsion, "Rational Expectations, the Efficient Market Hypothesis, and the Santa Fe Artificial Stock Market Model"

Ref.[3] * L. Tesfatsion, "Detailed Notes on the Santa Fe Artificial Stock Market Model" (NOTE: Ref.[3] contains a detailed glossary of terms. Also, the equation numbers appearing below in this slide-set are the same as in Ref.[3].)


https://www2.econ.iastate.edu/tesfatsi/BuildingTheSFASM.BLeBaron.pdf
Introduction to the Santa Fe Artificial Stock Market (SF-ASM) Model

• Originated in work at the Santa Fe Institute (SFI) in the late 1980s and early 1990s.

• **Five Developers of the SF-ASM:**
  - Blake LeBaron (economics);
  - W. Brian Arthur (economics);
  - John Holland (psychology/EE/CS, and father of GAs);
  - Richard Palmer (physics);
  - Paul Taylor (computer science).
• **Seminal Research:** One of the earliest attempts to develop and implement a *computational financial market model with heterogeneously learning traders*.

• Relatively simple model that attempts to address several important and controversial questions in financial economics.

• Many modeling issues not satisfactorily resolved by the SF-ASM model have been taken up in later research; see, for example, ref. [4].
Basic Objectives of the Authors

- Provide a test-bed for exploring the rational expectations hypothesis (REH, Ref.[2])
- Consider a traditional stock market model with traders assumed to satisfy the REH
- Replace traditional REH traders with traders who learn to forecast stock prices over time
- Study dynamics around a well-studied REH equilibrium (fundamental pricing, Ref.[1])
Basic Author Objectives ... Continued

• Examine whether the introduction of trader learning helps to explain empirical observations.

• In particular, does it help to explain well-documented financial anomalies, such as deviations of stock prices from “fundamental values”?

• Compare statistical characteristics of price and trading volume outcomes (model outcomes vs. actual empirical outcomes).
Basic Model Features (cf. Ref.[3])

- Discrete-time model: $t = 0, 1, 2, \ldots$
- Market participants consist of $N$ stock market traders plus an “auctioneer”
- **KEY ASSUMPTION:** Traders are identical except that each trader individually forms expectations over time through inductive learning.
- Each trader has same initial wealth $W_0$ in the initial time period.
Basic Model Features... Continued

- Financial assets available for purchase at beginning of each period $t = [t,t+1)$:
  - **Risk-free asset $F$** ($\infty$ supply) paying a **constant** known 1-period net return rate $r$
  - **$N$ shares of a risky stock $A$**. Each share
    - pays an **uncertain** dividend $d_{t+1}$ at the end of each period $t$ (beginning of each period $t+1$);
    - has an **uncertain** one-period net return rate $R_t$ over each holding period $t$. 
• Let $p_t$ denote the price of a share of the risky stock $A$ at time $t$

• The expected net return rate $R_t$ on this share over period $t$ (i.e. from time $t$ to time $t+1$) is defined as

$$R_t = \frac{p_{t+1}^e - p_t + d_{t+1}^e}{p_t}$$

• This definition implies that

$$p_t = \frac{d_{t+1}^e + p_{t+1}^e}{1 + R_t}$$
Basic Model Features ... Continued

• The expected net return rate $R_t$ on a share of the risky stock $A$ over period $t$ satisfies:

$$p_t = \frac{p_{e_{t+1}} + d_{e_{t+1}}}{1 + R_t}$$

• Basic rule of thumb for an investor in period $t$:

Given $r =$ net return rate on the risk-free asset, **SELL** shares of $A$ in period $t$ if $R_t < r$ because this implies $p_t$ is **GREATER** THAN the current fundamental value of these shares:

$$p_{f_t} = \frac{p_{e_{t+1}} + d_{e_{t+1}}}{1 + r}$$
Basic Model Features... Continued

• **Stock Dividend** $d_t$ paid at beginning of each period $t = [t, t+1)$ is generated by a random process unknown to the traders (see equ.(1) in Ref.[3])

• Wealth-seeking traders have identical utility of **wealth function** $U(W)$ exhibiting constant absolute risk aversion.
• In beginning of each period \( t \), each trader chooses a portfolio \((X,Y)\), where \( X = \) holdings of risky stock \( A \) and \( Y = \) holdings of risk-free asset \( F \).

• Each trader’s objective in period \( t \) is to maximize his expected utility of wealth \( E(U(W_{t+1})) \) subject to the constraint

\[
(2) \quad W_{t+1} = \text{Value in period } t+1 \text{ of the asset portfolio } (X,Y) \text{ purchased in period } t
\]
• In beginning of each period $t$, each trader has a set of $K$ if-then forecasting rules.

• Each forecasting rule forecasts the expected sum $[p_{t+1} + d_{t+1}]$ and generates an update of the rule’s “forecast variance.”

**Forecast variance** = a weighted average of a rule’s past squared forecast errors (deviations between actual and forecasted price-plus-dividend sums).
Basic Model Features ... Continued

- Form of an *if-then* forecasting rule:

Let \( VAR =: \) Updated forecast variance

\[
E[p_{t+1} + d_{t+1}] = a[p_t + d_t] + b.
\]

<table>
<thead>
<tr>
<th>IF</th>
<th>THEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market state is in condition ( C' )</td>
<td>Set values ( VAR', a', b' )</td>
</tr>
</tbody>
</table>
Basic Model Features ... Continued

- The **specificity** of a forecasting rule = number of specific conditions incorporated into its “if” condition statement C.

- A forecasting rule is **activated** if its “if” condition statement C matches the trader’s current market state information.

- The **fitness** of a forecasting rule depends *inversely* on the rule’s forecast variance (error rate) and *inversely* on its specificity (thus encouraging parsimonious info use).
Time Line of Activities in Period t

- Period-t dividend $d_t^*$ is publicly posted.
- Each trader $i=1,...,N$ determines a forecast
  \[ E[p_{t+1} + d_{t+1}] = a'[p_t + d_t] + b' \]
  as a function of the \textit{yet-to-be determined} period-t market price $p_t$.
- He then generates a \textbf{demand function} giving his expected-utility-maximizing share holdings $X_i$ as a function of $p_t$:
  \[ X_i = X_i(p_t) \]
Each trader $i = 1, \ldots, N$ submits his demand function to the Auctioneer, who determines the *period-t market clearing price* $p_t^*$:

\[
\sum X_i = \sum X_i(p_t)
\]
The Auctioneer publicly posts $p_t^*$. 

Each trader $i$ purchases $X_i(p_t^*)$. 

Each trader $i$ uses $(p_t^*,d_t^*)$ to update the fitness of the forecasting rule he used in period $t-1$ to generate a forecast $E[p_t + d_t]$. 

Each trader $i$ with probability $p_u$ then updates his entire forecasting rule set via a genetic algorithm involving recombination, elitism, and mutation operations.
GA Classifier Learning

• Each trader $i = 1, ..., N$ updates his set of forecasting rules with an exogenously given probability $p_u$ in each period $t$, making use of a Genetic Algorithm (GA).

• Thus, updating of forecasting rule sets happens in different time periods for different traders.

• $p_u$ is a very important model parameter since it determines the traders’ “speed of learning”.
GA Classifier Learning ... Continued

- Current market state $\rightarrow$ 12-bit array

- Each bit position in this 12-bit array corresponds to a distinct possible feature of the current market state:
  - Bit in kth position takes on value 1 if kth feature is true
  - Bit in kth position takes on value 0 if kth feature is false
GA Classifier Learning...Continued

- **12-bit array** used to describe market state

- **First six bit positions** ➔ *Fundamental Features*
  
  Is the current market price above or below the fundamental price level in the previous time period? *(six different possible discrepancy values)*
GA Classifier Learning ... Continued

- Next four bit positions
  - Technical Features
    Is the current market price above an $n$-period moving average of past prices? (four different values of $n$)

- Last two bit positions
  - Fixed Bit Values (no information)
### 12-Bit Array for GA Classifier Learning

<table>
<thead>
<tr>
<th>Bit</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Price * interest/dividend &gt; 1/4</td>
</tr>
<tr>
<td>2</td>
<td>Price * interest/dividend &gt; 1/2</td>
</tr>
<tr>
<td>3</td>
<td>Price * interest/dividend &gt; 3/4</td>
</tr>
<tr>
<td>4</td>
<td>Price * interest/dividend &gt; 7/8</td>
</tr>
<tr>
<td>5</td>
<td>Price * interest/dividend &gt; 1</td>
</tr>
<tr>
<td>6</td>
<td>Price * interest/dividend &gt; 9/8</td>
</tr>
<tr>
<td>7</td>
<td>Price &gt; 5-period MA</td>
</tr>
<tr>
<td>8</td>
<td>Price &gt; 10-period MA</td>
</tr>
<tr>
<td>9</td>
<td>Price &gt; 100-period MA</td>
</tr>
<tr>
<td>10</td>
<td>Price &gt; 500-period MA</td>
</tr>
<tr>
<td>11</td>
<td>On: 1</td>
</tr>
<tr>
<td>12</td>
<td>Off: 0</td>
</tr>
</tbody>
</table>

**Note on Rules 7-10:**

MA := Moving Average

= Weighted average of past observed prices

**Note on Rules 1-6:**

pr/d > 1 if and only if p > [p+d]/(1+r), i.e., if and only if the current price p for a share of the risky stock A exceeds the “fundamental” value of this share realized in the previous time period. *(Refer back to slide 10.)*
GA Classifier Learning ... Continued

Why this market state description?

* Permits testing for the possible emergence of
  
  *fundamental trading* (heavy reliance on first six bit positions)
  
  versus
  
  *technical trading* (heavy reliance on next four bit positions)
  
  versus
  
  *uninformed trading* (heavy reliance on last two bit positions).
• Each forecast rule, taking form $\text{if}[C]\text{-then[forecast this]}$, is conditioned on a 12-bit market state $C$.

• Each bit in $C$ has one of three possible values: $1$ (true), $0$ (false), or # (I don’t care).

• Specificity of $C$ =: Number of 1 and 0 bits in $C$

• $C$ is said to “match” the actual 12-bit market state if:
  (a) $C$ has a 1 or # symbol in every position for which the actual market state has a 1;
  (b) $C$ has a 0 or # symbol in every position for which the actual market state has a 0.
Experimental Design

- **Key Treatment Factor: “Speed of Learning”** Prob \( p_u \)
  Controls when each trader updates their forecasting rule set in any given time period

- **Slow-Learning Regime:** \( p_u = 1/1000 \)
  (GA learning invoked every 1000 trading periods on average for each trader)

- **Medium-Learning Regime:** \( p_u = 1/250 \)
  (GA learning invoked every 250 trading periods on average for each trader)
Experimental Findings

• **Slow-Learning Regime:** $p_u = 1/1000$
  Simulated data resemble data generated for a *rational expectations equilibrium (REE)* benchmark, for which 100% market efficiency holds by assumption.

• **Medium-Learning Regime:** $p_u = 1/250$
  *Complex outcomes* -- market does not settle down to a recognizable equilibrium. Simulated data in accordance with many empirical “anomalies” (deviations from REH) seen in actual stock markets.
Frequency of Use of “Technical Trading” Bits 7-10 in REE vs. Complex Regimes

Figure 3. Number of technical-trading bits that become set as the market evolves, (median over 25 experiments in the two regimes).
Final Remarks

- For a balanced detailed critique of the Santa Fe Artificial Stock Market (SF-ASM), see the working paper by Blake LeBaron that appears below as Ref. [5].

- In this working paper, LeBaron discusses the advantages and disadvantages of various design aspects of the SF-ASM, including the use of “classifier systems” for the representation and evolution of forecasting rules.

https://www2.econ.iastate.edu/tesfatsi/BuildingTheSFASM.BLeBaron.pdf