Abstract

The discovery of evolutionary laws of financial market is always built on the basis of financial data. Any financial market must be controlled by some basic laws, including macroscopic level, submicroscopic level and microscopic level laws. How to discover its necessity-laws from financial data is the most important task of financial market analysis and prediction. Based on the evolutionary computation, this paper proposes a multi-level and multi-scale evolutionary modeling system which models the macro-behavior of the stock market by ordinary differential equations while models the micro-behavior of the stock market by natural fractals. This system can be used to model and predict the financial data(some time series), such as the stock market data of Dow-Jones index and IBM stock price, and always get good results.

1 INTRODUCTION

The financial data or data get from some real-world systems such as stock market usually are very complex, but any complex system is bound to be controlled by some basic laws, including macroscopic level laws, submicroscopic level laws and microscopic level laws. Consider the following financial data as the time series:

\[ x(t_i), x(t_1), \ldots, x(t_m) \]  

(1)

where \( t_i = t_0 + i \theta \), \( \theta \) is the time stepsize. As for the time series, besides the traditional method of time series analysis, evolutionary algorithm is usually used to cope with these data.

Suppose that the financial data are controlled by macroscopic, sub-macroscopic and microscopic rules. We take the multi-level, multi-scale models for analyzing and predicting the financial data. In this paper, we use the ordinary differential equation (ODE) model to describe the macroscopic behavior of the stock market, while we use the natural fractal models (a kind of natural discrete wavelets) to describe its microscopic behavior. In this way, we build a multi-level and multi-scale evolutionary modeling system for financial data. This system provides a strong tool for the analysis and prediction of complex time series. The observed data of the Dow-Jones index and IBM stock price are used as the test data for this system.

The rest of the paper is arranged as follows: in section 2, we introduce the macroscopic ODE model; in section 3, we introduce the microscopic natural fractal model; numerical experiments are given in section 4; and in the end, section 5 is some conclusions.

2 MACROSCOPIC ODE MODEL

The complex time series are usually characteristics of multi-level and multi-scale. Assume that it has two levels: macro and micro. In order to describe it in macroscopic level, the researchers take many kinds of methods to handle the time series.

2.1 DECOMPOSITION OF ORIGINAL DATA

In order to find out macroscopic laws from complex data, the first step is to decompose the original data \( x(t_i), i=0,1,2,\ldots,m \) as in (1) into two parts: the smooth part and the coarse part (non-smooth part). We assume that the evolutionary process of the smooth part is controlled by macroscopic factors, and the evolutionary process of the coarse part is controlled by microscopic factors. The smooth part will be modeled by ordinary differential equations (ODE), while the coarse part be modeled by natural fractals (a kind of multi-scale discrete wavelets).

For the time series (1), we decompose it into two parts:

\[ x(t_i) = \bar{x}(t_i) + \tilde{x}(t_i) \quad i = 0,1,\ldots,m \]  

(2)

where the smooth part \( \bar{x}(t_i) \) is defined as:
\[ x(t_j) = \sum_{i=1}^{1} \sum_{j=0}^{l} x(t_{ij}) \quad i < l \]
\[ x(t_j) = \frac{1}{l+1} \sum_{j=0}^{m} x(t_{ij}) \quad l \leq i \leq m \]  
\[ (3) \]

and the coarse part:
\[ \tilde{x}(t_j) = x(t_j) - x(t_j), \quad i = 0, 1, \cdots, m \]  
\[ (4) \]

Notice: (3) and (4) is related to the value of the smooth parameter \( l \) which is often proportional to \( m \), when \( l \) is bigger, the time series \( \{ \tilde{x}(t_j) \} \) is more smooth.

### 2.2 MACROSCOPIC HIGHER-ORDER ODE MODEL

In this subsection, we will mainly introduce how to model and predict the smooth data \( \{ x(t_j) \}_{j=0}^{m} \). Since the smooth data describe the macro behavior of dynamic system and determine the macroscopic tendency of the system, it is the essential part of observed data. Because it is the smooth part of observed data, we assume that \( \tilde{x}(t) \) is sufficiently smooth, that is, assume that \( \tilde{x}(t) \in C_{[t_0, T]}, \quad \forall t \in [0, T]. \)

The modeling problem of the dynamic system \( \tilde{x}(t) \) is to find an initial value problem of the \( n \)-th order ordinary differential equation:
\[ x^{(n)}(t) = f(t, x(t), x'(t), \cdots, x^{(n-1)}(t)) \]
\[ (5) \]

such that the mean square error between the values of its solution \( \tilde{x}(t) \) at \( t_i, \quad i = 0, 1, \cdots, m \) and the series \( \{ \tilde{x}(t_j) \} \) as small as possible.

Denote
\[ \| x - \tilde{x} \| = \frac{1}{m+1} \left( \sum_{i=0}^{m} (x(t_i) - \tilde{x}(t_i))^2 \right) \]
\[ (6) \]

That is to say, to find \( f \) in function space \( F \), such that
\[ \min_{f \in F} \| x - \tilde{x} \| \]  
\[ (7) \]

This problem is solved by evolutionary modeling algorithm described in [7]. The main idea of the algorithm is to embed a genetic algorithm in genetic programming used to discover and optimize the structure of a model, while GA is used to optimize its parameters.

The evolutionary modeling algorithm for higher-order ordinary differential equation can be simply described as follows:

**PROCEDURE 1**

**begin**

Initialize population \( P(0) = \{ p_1(0), p_2(0), \cdots, p_N(0) \} \); (produce \( N \) parse trees randomly)
\[ t := 0; \]

**end**

Output solution of \( \tilde{P}_{best} : \tilde{x}^\circ(t_j), i = 0, 1, \cdots, m + q \). For the details of the process of modeling, please refer to [7].

### 3 MICROSCOPIC FRACATL MODEL

For the coarse part \( \{ \tilde{x}(t_j) \}_{j=0}^{m} \) of the time series (1):
\[ \tilde{x}(t_j) = x(t_j) - x(t_j), \quad i = 0, 1, \cdots, m \]

we are going to build a multi-scale micro natural fractal model.

#### 3.1 CONSTRUCTION OF NATURAL WAVELETS

Denote
\[ \tilde{x} = \sum_{i=0}^{m+1} \tilde{x}(t_j) / (m+1) \]
\[ (8) \]

In order to search an \( l \)-scale basic natural wavelet of series (4), we divide the series \( \{ \tilde{x}(t_j) \}_{j=0}^{m} \) into \( l \) groups (from row to column to form the following matrix (see Table 1), each column as a group, including \( l \) groups), where \( x_i \) denotes \( \tilde{x}(t_j) \).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>S+1</th>
<th>...</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x_0 )</td>
<td>( x_1 )</td>
<td>...</td>
<td>( x_{S+1} )</td>
<td>...</td>
<td>( x_l )</td>
</tr>
<tr>
<td>2</td>
<td>( x_1 )</td>
<td>( x_{S+1} )</td>
<td>...</td>
<td>( x_{2S+1} )</td>
<td>...</td>
<td>( x_{2l} )</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\ddots</td>
<td>\vdots</td>
<td>\ddots</td>
<td>\vdots</td>
</tr>
<tr>
<td>( k )</td>
<td>( x_{S+1} )</td>
<td>( x_{(S+1)+1} )</td>
<td>...</td>
<td>( x_{2S+1} )</td>
<td>...</td>
<td>( x_{S+(l-1)+S} )</td>
</tr>
<tr>
<td></td>
<td>average</td>
<td>( \tilde{x}_1 )</td>
<td>( \tilde{x}_2 )</td>
<td>...</td>
<td>( \tilde{x}_{S+1} )</td>
<td>( \tilde{x}_l )</td>
</tr>
</tbody>
</table>

\[ \tilde{x}_i = \left( \sum_{j=1}^{k} x_{(j-1)l+S} / k \right) \]  
\[ \tilde{x}_i \]  
\[ (9) \]

where
The process can be described as follows: a multi-scale natural fractal model with scale part in order to build the mathematical model for the coarse part of time series \( x(t) \) of order \( m \), we introduce the variance ratio:

\[
E_i = \frac{\sum_{i=1}^{m} k_i^* (x_i^e - \bar{x})^2}{(l-1)} - \frac{\sum_{i=1}^{m} (x_{i,j-1,j+1} - \bar{x})^2}{(m-l+1)}
\]

In order to test whether \( x(t) \) is a basic natural wavelet of time series (4), we introduce the variance ratio:

\[
E_i = \frac{\sum_{i=1}^{m} k_i^* (x_i^e - \bar{x})^2}{(l-1)} - \frac{\sum_{i=1}^{m} (x_{i,j-1,j+1} - \bar{x})^2}{(m-l+1)}
\]

Assume that \( E_i \) has an F-distribution with \((l-1, m-l+1)\) degrees of freedom. For different confidence level of \( a \) and \((l-1, m-l+1)\) degrees of freedom, we can get \( F_a(l-1, m-l+1) \) from a F-distribution table.

If \( E_i = F_a(l-1, m-l+1) \), then the time series (4) exists \( l \)-scale basic natural wavelet \( x(t) \) in confidence level \( a \). If \( E_l < F_a(l-1, m-l+1) \), then the series (4) does not exist \( l \)-scale basic natural wavelet in confidence level \( a \).

### 3.2 MICROSCOPIC NATURAL fractal MODEL

In order to build the mathematical model for the coarse part \( \bar{x}(t) \) of the time series (4), we construct a multi-scale natural fractal model with scale \( l=2, 3, \ldots, L \). The process can be described as follows:

**PROCEDURE 2**

**begin**

1. **initialize** \( \bar{x} := \bar{x} \) where \( \bar{x} = \{ \bar{x}(t_i) \}_{i=0}^{m} \)

2. **for** \( l=2, L, \) **do**

   a. **using** \( \{ \bar{x}(t_i) \}_{i=0}^{m} \) **calculate** \( x' = \{ x'_1, x'_2, \ldots, x'_l \} \);

   b. **if** \( E_l = F_a(l-1, m-l+1) \) **then**

      - \( x^* := x^* + x' \) where \( x' = \{ x'_j \}_{j=0}^{l} \), \( j \equiv i \) (mod \( l \))

   **end for**

3. **end for**

\[
\varepsilon := \frac{\sum_{i=0}^{m} x(i) - x^*(i)}{m+1};
\]

**end for**

**Remark 1:** The output \( \{ \bar{x}^* (t_i) \}_{i=0}^{m} \) is the fitting part of \( \{ \bar{x}(t_i) \}_{i=0}^{m} \) and \( \{ \bar{x}^* (t_i) \}_{i=0}^{m+q} \) is the prediction part of \( \bar{x}(t) \). \( \varepsilon \) is the average fitting error, and it has been eliminated as the correction (random error).

**Remark 2:** The first part of the procedure is to test successively whether the time series exists basic natural wavelet with scale \( l \) which is less than \( L \), usually \( L=(m+1)/3 \), for some special problems, where \( m \) is relatively small, \( L \) can be magnified to \( L=(m+1)/2 \) if it exists, then prolong it periodically to whole interval \( [t_0, t_{m+q}] \) and add to the time series \( \{ x^* (i) \}_{i=0}^{m+q} \).

**Remark 3:** The second part of the procedure is to evaluate the random error of \( \{ \bar{x}(t_i) \} \), and then correct it when fitting and prediction.

### 3.3 THE MULTI-LEVEL AND MULTI-SCALE EVOLUTIONARY MODELING SYSTEM

Using the macroscopic modeling \**PROCEDURE 1** of nth-order ordinary differential equation (5), and the microscopic modeling \**PROCEDURE 2** of natural fractal, we can build a multi-level and multi-scale evolutionary modeling system for fitting and prediction of complicated time series. Firstly, call \**PROCEDURE 1** to build the ODE (5), and use Runge-Kutta method to solve it to get the fitting and prediction values of the smooth part \( x^*(t) \), \( i = 0, 1, \ldots, m + q \), then call \**PROCEDURE 2** to get the fitting and prediction values of the coarse part: \( \bar{x}'(t) \), \( i = 0, 1, \ldots, m + q \). Adding up these data, we can get the needed fitting and prediction values:

\[
\bar{x}'(t_i) + x^*(t_i) = x(t_i), i = 0, 1, \ldots, m + q
\]

This procedure can be described as follows:

**PROCEDURE 3**

**begin**

1. **Decompose data** \( x[0, m] \) into \( \bar{x}[0, m] \) and \( x[0, m] \);
2. **Call \**PROCEDURE 1** to get** \( \bar{x}'[0, m + q] \);
3. **Call \**PROCEDURE 2** to get** \( \bar{x}'[0, m + q] \);

   a. **for** \( i = 0, m \) **do**

      - \( x^*(t_i) := x^*(i) + \bar{x}^*(i) \);

      - \( e(i) := x(i) - x^*(t_i) \);

   **end for**

   b. **for** \( i = m+1, m+q \) **do**

      - \( x^*(t_i) := x^*(i) + \bar{x}^*(i) \);

   **end for**

**output** \( x^*(t_i), i = 0, 1, \ldots, m + q \);
output $e(i), i = 0, 1, ..., m$; 

end

Remark 1: The first step of the procedure is to decompose the original time series into two parts: the smooth part and the coarse part (non-smooth) part.

Remark 2: The second step of the procedure is to call PROCEDURE 1 to deal with the smooth data $\{\mathbf{X}(i)\}$ and get an ODE model of and the values of its solution: $\mathbf{X}(t_0), \mathbf{X}(t_1), \cdots, \mathbf{X}(t_{m+q})$, where the first $m+1$ values are the fitting values of $\mathbf{X}(t)$, and the later $q$ values are the prediction values of $\mathbf{X}(t)$.

Remark 3: The third step of the procedure is to call PROCEDURE 2 to deal with the coarse data $\mathbf{X}(t)$ and get a multi-scale natural fractal model and its solution: $\mathbf{X}(t_0), \mathbf{X}(t_1), \cdots, \mathbf{X}(t_{m+q})$, where the first $m+1$ values are the fitting values of $\mathbf{X}(t)$, and the later $q$ values are the prediction values of $\mathbf{X}(t)$ at $t_{m+1}, t_{m+2}, ..., t_{m+q}$.

Remark 4: The fourth step of the procedure is to combine the fitting values of the smooth part with those of the coarse part of $x(t)$ to get the time series $(x^*(t), \mathbf{X})$ and the fitting error $(\mathbf{X}(t))$. The fifth step of the procedure is to combine the prediction values of the smooth part with those of the coarse part of $x(t)$ to get the prediction values of $x(t)$ at the time $t_{m+1}, t_{m+2}, ..., t_{m+q}$, where the prediction length $q$ can be decided by the users.

4 NUMERICAL EXPERIMENTS

In this section, we mainly study the applications of multi-level and multi-scale evolutionary modeling system to the financial data.

Firstly we use the smooth data of BUMP problem as the test data of the smooth model.

Using the method described in section 2, we discovered higher-order ODEs to model the time series $f_2, f_3, f_4, ..., f_{50}$. Denote $f = f(t)$, where $t = t_0 + t$, $t_0 = 2$, and $t = 0.01$.

Table 2. Solution table of BUMP problem

<table>
<thead>
<tr>
<th>$n$</th>
<th>$f_n$</th>
<th>$n$</th>
<th>$f_n$</th>
<th>$n$</th>
<th>$f_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18</td>
<td>2</td>
<td>1.59497975</td>
<td>19</td>
<td>0.79800887</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>5</td>
<td>0.51578550</td>
<td>30</td>
<td>0.80361910</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>6</td>
<td>0.62282103</td>
<td>31</td>
<td>0.80464587</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>7</td>
<td>0.63444869</td>
<td>32</td>
<td>0.80833226</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>8</td>
<td>0.69386488</td>
<td>33</td>
<td>0.81182640</td>
</tr>
<tr>
<td>7</td>
<td>24</td>
<td>9</td>
<td>0.70495107</td>
<td>34</td>
<td>0.81399253</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>10</td>
<td>0.72762616</td>
<td>35</td>
<td>0.81496495</td>
</tr>
<tr>
<td>9</td>
<td>26</td>
<td>11</td>
<td>0.74126604</td>
<td>36</td>
<td>0.81648731</td>
</tr>
<tr>
<td>10</td>
<td>27</td>
<td>12</td>
<td>0.7473103</td>
<td>37</td>
<td>0.81918437</td>
</tr>
<tr>
<td>11</td>
<td>28</td>
<td>13</td>
<td>0.76105561</td>
<td>38</td>
<td>0.82188436</td>
</tr>
<tr>
<td>12</td>
<td>29</td>
<td>14</td>
<td>0.76256413</td>
<td>39</td>
<td>0.82210164</td>
</tr>
<tr>
<td>13</td>
<td>30</td>
<td>15</td>
<td>0.77333853</td>
<td>40</td>
<td>0.82442369</td>
</tr>
<tr>
<td>14</td>
<td>31</td>
<td>16</td>
<td>0.77726156</td>
<td>41</td>
<td>0.82783593</td>
</tr>
<tr>
<td>15</td>
<td>32</td>
<td>17</td>
<td>0.78444946</td>
<td>42</td>
<td>0.83047389</td>
</tr>
<tr>
<td>16</td>
<td>33</td>
<td>18</td>
<td>0.78787044</td>
<td>43</td>
<td>0.83226201</td>
</tr>
<tr>
<td>17</td>
<td>34</td>
<td>19</td>
<td>0.79150564</td>
<td>44</td>
<td>0.83323002</td>
</tr>
</tbody>
</table>

Liu got the best solutions of the BUMP problem for $n = 2, 3, ..., 50$ as showed in Table 2, where $f_n = \text{Max } f_n(x)$. The best solutions are depicted in Fig. 1.

We want to discover higher-order ODEs to model the time series $f_2, f_3, f_4, ..., f_{50}$. Denote $f = f(t)$, where $t = t_0 + i$, $t_0 = 2$, and $t = 0.01$.

Fig. 1: Best results of $f_2$ to $f_{50}$

Using the method described in section 2, we discovered the following model by computer automatically:

\[
\frac{d^2 f(t)}{dt^2} = -15.658156 \frac{df(t)}{dt}(t + df(t)/dt) \\
\frac{df(t)}{dt} = 0.1508058
\]
Where the modeling error is 0.00095677, this means the model fits the solutions of BUMP problems very well. We use it to predict the solution of the challenge problem by using Runge-Kutta method with \( \Delta t = 0.01 \) in 1000000 steps, the results \( f(100), f(200), \ldots, f(1000000) \) are shown in Table 3. The results of \( f_{100}, f_{200}, \ldots, f_{1000000} \) of the BUMP problem[9] got by Liu on a massively parallel computer are compared in Table 3.

Table 3: the Comparison of \( f_n \) and \( f(n) \)

<table>
<thead>
<tr>
<th>n</th>
<th>( f_n )</th>
<th>( f(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.8448539</td>
<td>0.8445141</td>
</tr>
<tr>
<td>200</td>
<td>0.8468442</td>
<td>0.84503153</td>
</tr>
<tr>
<td>300</td>
<td>0.8486441</td>
<td>0.84503450</td>
</tr>
<tr>
<td>400</td>
<td>0.8511074</td>
<td>0.84503451</td>
</tr>
<tr>
<td>500</td>
<td>0.8504975</td>
<td>0.84503451</td>
</tr>
<tr>
<td>1500</td>
<td>0.8449622</td>
<td>0.84503451</td>
</tr>
<tr>
<td>10000</td>
<td>0.8456407</td>
<td>0.84503451</td>
</tr>
<tr>
<td>20000</td>
<td>0.8455883</td>
<td>0.84503451</td>
</tr>
<tr>
<td>100000</td>
<td>0.8448940</td>
<td>0.84503451</td>
</tr>
<tr>
<td>1000000</td>
<td>0.8445861</td>
<td>0.84503451</td>
</tr>
</tbody>
</table>

These results show that the smooth model got by the new modeling system gives a good long-range prediction.

4.2 MODELING OF THE DATA OF DOW-JONES INDEX

The observed data shown in Fig.2 are taken from [10] giving the daily Dow-Jones index over 132 days in 2000. We take the observed data of the first 126 days as historical data (training data) to build models which are used to predict the Dow-Jones index of the last 6 days.

Parameter settings of the modeling experiments are \( m=126, l=4 \) for smoothing, \( m=126, q=6, t_0=0, \Delta t=0.01 \) (one day), \( N=100, n=2 \) (the second-order ODE) for macroscopic ODE model, and \( m=359, q=10, L=120, a=0.1 \) for microscopic natural fractal model. We get a second-order ODE model as follows:

\[
\frac{d^2x}{dt^2} = -\frac{1507.55537}{\cos x} - \cos^2 x
\]

The results are shown in Fig.2.

4.3 MODELING OF IBM STOCK PRICE DATA

The observed data shown in Fig.3 are taken from [10] giving the daily stock price of IBM Company from May 17,1961 to November 2,1962. We take the observed data of the first 359 days as the training data to build models which are used to predict the stock price of the last 10 days.

Parameter settings of the modeling experiments are \( m=54, l=10 \) for smoothing, \( m=359, q=10, t_0=0, \Delta t=0.01 \) (one day), \( N=100, n=2 \) (the second-order ODE) for macroscopic ODE model, and \( m=359, q=10, L=120, a=0.1 \) for microscopic natural fractal model. We get a second-order ODE model as follows:

\[
\frac{d^2 x}{dt^2} = -17228.009766 + \frac{3672.875732}{\sin(x) \sin(\cos t * 1115.356812)}
\]

The results are shown in Fig.3.

5 CONCLUSION

Compared with most available modeling methods, the multi-level and multi-scale evolutionary modeling system has the following advantages:

Firstly, the entire process is automatic and requires little information in the way of the real-world system or
Secondly, it allows one to model the macro-behavior of the system by ordinary differential equations and to model the micro-behavior of the system by multi-scale natural fractals simultaneously.

Finally, the models discovered by computers from the complicated financial data can fit the original data quite well, and the structures of the ODE models are unimaginably to humans.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Nos. 70071042, 60073043, 60133010).

References


