Financial Data Mining with Genetic Programming: a Survey and Look Forward

Nicolas NAVET – INRIA
France
nnavet@loria.fr

Shu-Heng CHEN – AIECON/NCCU
Taiwan
chchen@nccu.edu.tw


Genetic programming

- GP is the process of evolving a population of computer programs, that are candidate solutions, according to the evolutionary principles

Generate a population of random programs

Evaluate their quality (“fitness”)

Solution

Create better programs by applying genetic operators, eg
- mutation
- combination (“crossover”)
In GP, programs are represented by trees

- **Trading system**: buy if \( \text{abs}(\text{Close}(t)/0.7748) < \text{Close}(t - 218) \)

Typical genetic operator: standard crossover

- **Standard crossover**: exchange two randomly chosen sub-trees among the parents
Strong points of GP

- Solutions are produced under a symbolic form that can be analyzed by humans
- GP does not assume a predefined size and shape: it creates both the functional form and the parameters’ values

- "Ability to produce a large number of different, yet meaningful hypotheses .. that are non-intuitive and sometimes provocative" [Kei02]

G.P. in the financial domain

1. Knowledge discovery:
   - results are scarce
   - Agent based modeling: study the evolution of a population of decision rules
   - Testing the EMH in real and artificial markets

2. Financial trading:
   - Composing portfolios
   - Evolving structure of NN used for prediction
   - Predicting price evolution
   - Discovering trading rules
Discovering trading rules: the big picture

1) Creation of the trading rules using GP
2) Selection of the best resulting strategies

Further selection on unseen data - One strategy is chosen for out-of-sample

Performance evaluation

Improvements ahead of us (1/2)

1. Rigorous assessment of the GP outcomes: controlling the data-mining bias!
2. Selecting the right time series: market can be efficient
3. Reducing variability of the results from GP run to GP run
4. Re-thinking the data-division scheme for training, validation and testing periods
Improvements ahead of us (2/2)

5. Pre-processing the data ?!?  
6. Re-thinking fitness functions : GP-friendly, sensitivity and risk adjusted, ...  
7. Embedding more domain specific knowledge : GP function set is still very primitive ..

1. Rigorous assessment of the GP outcomes
### GP’s outcomes on the training interval (1/2)

- Assume an “inefficient” solution leads to a profitable trade with probability 0.5

<table>
<thead>
<tr>
<th>Success rate</th>
<th>Number of trades</th>
<th>10</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td></td>
<td>0.38</td>
<td>0.1</td>
<td>0.03</td>
</tr>
<tr>
<td>70%</td>
<td></td>
<td>0.17</td>
<td>3·10⁻³</td>
<td>4·10⁻⁵</td>
</tr>
</tbody>
</table>

- **Guideline**: penalize or discard systems with few trades

### GP’s outcomes on the training interval (2/2)

- Probability than at least one inefficient system achieves a success rate = 70% for a given number of solutions

<table>
<thead>
<tr>
<th>Number of solutions tested</th>
<th>Number of trades</th>
<th>10</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td>1</td>
<td>0.28</td>
<td>0.004</td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td>1</td>
<td>0.96</td>
<td>0.38</td>
</tr>
<tr>
<td>50000</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0.85</td>
</tr>
</tbody>
</table>

- **NB**: in a typical GP run, 50000 solutions are tested and the average number of trades is usually small...
GP’s outcomes on the testing period [ChNa07]

- Compare GP with several variants of
  - Random search algorithms
    - “Zero-Intelligence Strategies” - ZIS
  - Random trading behaviors
    - “Lottery trading” - LT

**Issue**: how to best constrain randomness?

- Statistical hypotheses testing
  - Null: GP does not outperform ZIS
  - Null: GP does not outperform LT

2. Selecting the Right Time Series

**Experiments [CIEF2007]**:
Does low entropy imply better profitability of GP-induced GP Trading Rules?

NYSE US 100 Stocks
Daily Data from 2000 to 2006
Experimental setup

- Entropy rate estimator: Kontoyannis et al 1998
- $r_t = \ln\left(\frac{p_t}{p_{t-1}}\right)$
- Discretization: $\{r_t\} \in \mathbb{R} \rightarrow \{A_t\} \in \mathbb{N}$

3,4,1,0,2,6,2,…

alphabet of size 8 - equal number of values in each bin $\Rightarrow$ max. theoretical entropy = 3

Entropy of NYSE US 100 stocks – period 2000-2006

Mean = Median = 2.75
Max = 2.79
Min = 2.68
Rand() boost = 2.96
Rand() C lib = 2.77 !

NB: a normal distribution of same mean and standard deviation is plotted for comparison.
Entropy is high but price time series are not random!

Original time series

Randomly shuffled time series

Stocks in the distribution’s tails

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>OXY</td>
<td>2.789</td>
</tr>
<tr>
<td>VLO</td>
<td>2.787</td>
</tr>
<tr>
<td>MRO</td>
<td>2.785</td>
</tr>
<tr>
<td>BAX</td>
<td>2.78</td>
</tr>
<tr>
<td>WAG</td>
<td>2.776</td>
</tr>
</tbody>
</table>

Highest entropy time series

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWX</td>
<td>2.677</td>
</tr>
<tr>
<td>EMC</td>
<td>2.694</td>
</tr>
<tr>
<td>C</td>
<td>2.712</td>
</tr>
<tr>
<td>JPM</td>
<td>2.716</td>
</tr>
<tr>
<td>GE</td>
<td>2.723</td>
</tr>
</tbody>
</table>

Lowest entropy time series
Up to a lag 100, there are 2.7 x more autocorrelations outside the 99% confidence bands for the lowest entropy stocks than for the highest entropy stocks.

BDS tests: are daily log price changes i.i.d?

<table>
<thead>
<tr>
<th>m</th>
<th>δ</th>
<th>TWX</th>
<th>EMC</th>
<th>C</th>
<th>JPM</th>
<th>GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>18.06</td>
<td>14.21</td>
<td>13.9</td>
<td>11.82</td>
<td>11.67</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>22.67</td>
<td>19.54</td>
<td>18.76</td>
<td>16.46</td>
<td>16.34</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>34.18</td>
<td>29.17</td>
<td>28.12</td>
<td>26.80</td>
<td>24.21</td>
</tr>
</tbody>
</table>

Null that log price changes are i.i.d. always rejected at 1% level but - whatever BDS parameters - rejection is much stronger for high-entropy stocks.
Results: surprisingly ..

**On high-entropy stocks**
- GP is always profitable
- LT is never better than GP (95% confidence level)
- GP outperforms LT 2 times out of 5 (95% C.L.)

**On low-entropy stocks**
- GP is never better than LT (95% C.L.)
- LT outperforms GP 2 times out of 5 (95% C.L.)

Explanations (1/2)
- GP is not good when training period is very different from out-of-sample e.g.

Typical low complexity stock (EMC) vs. Typical high complexity stock (MRO)
Explanations (2/2)

- The 2 cases where GP outperforms LT: training quite similar to out-of-sample

4. Re-thinking data division scheme
Data division scheme

- There is multiple evidence that GP performs poorly when training interval ≠ from the out-of-sample interval ...
- What is needed: characterization of the market condition – similarity measure
- Re-learning triggered when similarity or performances below a threshold

5. Re-thinking fitness functions
Rethinking fitness functions

- **Issue 1**: some fitness functions induce a "difficult" landscape for GP → **GP-friendly fitness**

- **Issue 2**: a few lucky trades alone may lead to an outstanding return → **risk-adjusted fitness**

- **Issue 3**: solutions located on peaks of the fitness landscape are not robust out-of-sample → **sensitivity-adjusted fitness**

7. Embedding more domain specific knowledge
Embedding more domain specific knowledge

- Choice of the function/terminal sets is crucial – no guidelines - 2 risks:
  - Extraneous functions
  - Required functions not available
- As yet, GP uses a very primitive language
  - Enrich primitive set with volume, indexes, bid/ask spread, ...
  - Enrich function set with cross-correlation, predictability measure, ...

References (1/2)

References (2/2)