IEEE SSCI 2015 Tutorial

Equation Discovery for Economic Modelling

Dimitar Kazakov
Computer Science Dept
University of York

Zhivko Georgiev
Software Engineer
Bolyar Ltd (Ericsson contract)

Cape Town, 8 Dec 2015

York
Equation Discovery

- An AI approach akin to Machine Learning
- Aiming to find equations best fitting the data
- LAGRAMGE (Todorovski; Džeroski 2001):
  - Equation search space is defined in terms of operators, used-defined functions and range of parameters
  - CFG rules used to derive all possible equations

Equation Discovery (2)

- Data: \((0, 0), (\pi, 0), (2\pi, 0)\)
- \(y = 0, y = c_i \sin(x), y = c_i x(x-\pi)(x-2\pi)\) all fit!

- Model range based on experts' knowledge
Case Study: Modelling Inflation

- Empirical modelling of inflation
- ML, equation discovery and Lagrange
- The Euro area dataset
- Learning setup: range of models explored
- Results and evaluation
- Discussion

Models of Inflation

- An issue of great interest
- A great range of models
  - Theory-driven vs empirical
  - One to hundreds of variables
- Until 2007...
  - Inflation decreasing for ~2 decades
  - Decreasing inflation and output volatility
  - Less pronounced and shorter business cycles
  - Inflation increasingly detached from other systemic variables...
  - ...and staying close to 2%
- The above branded as: The Great Moderation (hah!)
2008-present

- Quite different!
- Models...experts... did anyone see it coming?
  - See my last slide
- A few excuses, where modelling is concerned:
  - The observer changes the studied system: publishing a model changes agents' behaviour.
  - Most current models do not represent well systems with several qualitatively different modes (but we can – and have – learned such models).

Lagramge Example

- Variables x, y, z
- Output variable: z
- Grammar: \[ E \rightarrow T \mid E + T \]
  \[ T \rightarrow c \mid c \ast Var \mid c \ast Var \ast Var \]
- Search space
  \[ z = a \]
  \[ z = ax + b \]
  \[ z = ax + bxy + cy + d \]
  etc., but limited by max depth of parse tree.
The Eurozone Data

- Tracking inflation, interest and output
- Quarterly data from 1971/Q1 — 2007/Q1
- Last 2 years used for testing only

<table>
<thead>
<tr>
<th>Quarter/Year</th>
<th>(\pi)</th>
<th>(y)</th>
<th>(r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1971Q1</td>
<td>5.25</td>
<td>4.22</td>
<td>0.68</td>
</tr>
<tr>
<td>1971Q2</td>
<td>5.83</td>
<td>3.56</td>
<td>-0.16</td>
</tr>
<tr>
<td>1971Q3</td>
<td>6.09</td>
<td>3.92</td>
<td>-0.14</td>
</tr>
<tr>
<td>1971Q4</td>
<td>6.36</td>
<td>3.50</td>
<td>-0.21</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2007Q1</td>
<td>1.87</td>
<td>3.07</td>
<td>1.93</td>
</tr>
</tbody>
</table>

Range of Models Considered

- Univariate models:
  - Inflation only a function of its past values
  - Simple yet quite accurate
  - Linear vs nonlinear regression; time trend

- Models linking nominal and real side of economy with monetary policy:
  - \(\text{output} = f(\text{interest rate, past output})\)
  - \(\text{inflation} = f(\text{output, past inflation})\)
  - \(\text{interest rate} = f(\text{output, past interest rate})\)
Setting up Lagrange

- Experimented with linear and nonlinear eqns
- The most complex search space tried:

if \( c_i \) are constants and \( V_i \) – variables, the RHS of each equation \( V_{\text{out}} = f(V_1, V_2, \ldots) \) is a sum of terms from this range:

- \( c_i \)
- \( c_i \cdot V_i \)
- \( c_i \cdot V_i \cdot V_j \)
- \( c_i \cdot V_i \sin(c_j \cdot V_j + c_k) \)
- \( c_i \cdot \sin(c_j \cdot V_j + c_k) \cdot \sin(c_j \cdot V_j + c_k) \)

Sample Run

Data files: annual-89-04A.csv
Variables: t infl infl-1 output output-1 intrst intrst-1
Data length: 64

Grammar:
Axiom -> Term | Axiom + Term;
Term -> const[\[-10:0.01:10\]] | const[\[-10:0.01:10\]] \* V | const[\[-10:0.01:10\]] \* V \* sin ( LTerm );
LTerm -> const[\[-10:0.01:10\]] \* V + const[\[-10:0.01:10\]];  

Equation type: ordinary explicit (infl)
Maximal parse tree depth: 5
Search strategy: exhaustive
Stopping criterion: none
Search heuristic: sum of squared errors

Restarts of parameter estimation methods: 0
Verbose: off

27778 parse trees evaluated
Sample Run (2)

Best equations:
\[ \text{infl} = 0.221514 + 0.0248229 \times t + 0.138563 \times \text{intrst} \times \sin (0.0822064 \times t + 1.24266) + 0.0155463 \times t \times \sin (0.0822064 \times \text{intrst} + -2.08429) \{\text{MSE} = 0.00804541, \text{MDL} = 0.0253219\} \]

\[ \text{infl} = 0.221513 + 0.0248229 \times t + 0.138563 \times \text{intrst} \times \sin (0.0822064 \times \text{intrst} + -2.08429) + 0.0155463 \times t \times \sin (0.0822064 \times t + 1.24266) \{\text{MSE} = 0.00804541, \text{MDL} = 0.0253219\} \]

\[ \text{infl} = 0.0301158 \times t + 0.0176492 \times t \times \sin (0.479913 \times \text{intrst} + -1.87172) + 0.157702 \times \text{intrst} \times \sin (0.0765666 \times t + 1.32732) \{\text{MSE} = 0.00812278, \text{MDL} = 0.0242847\} \]

\[ \text{infl} = -0.0940601 + 0.208327 \times \text{infl-1} + 0.136085 \times \text{intrst} \times \sin (0.075453 \times t + 1.26898) + 0.0360579 \times t \times \sin (0.25441 \times \text{intrst} + -0.117767) \{\text{MSE} = 0.00932121, \text{MDL} = 0.0265977\} \]

Time elapsed: 578.97 [s]

Evaluation

- The evaluation of the model accuracy is performed according to the root mean squared error (RMSE), and mean absolute error (MAD).
- **In-sample evaluation**: Model’s recall measured on training data (up to 2005Q1)
- **Out-of-sample evaluation**: models’ forecast potential tested on unseen test data (‘out of sample’) – from 2005Q2 onwards.
Univariate Linear Models: Inflation

- Standard linear regression model and the one produced by Lagramge – very similar
- Results indistinguishable for 1-step-ahead (3-month) forecast
- Out-of-sample accuracy > in-sample accuracy!
- Good safety check: the tool works well!

\[ \pi_t = 0.04 + 1.49\pi_{t-1} - 0.50\pi_{t-2} + \varepsilon_t \]
\[ \pi_t = 0.06 + 1.38\pi_{t-1} - 0.40\pi_{t-2} + \varepsilon_t \]

Univariate Linear Models: Plot
Univariate Linear Models: comment

- Recursive forecast: start with the last available observation and apply the model repeatedly to look ahead $N$ steps in the future.
- Sometimes recursive models are converted into equations with time as the only independent variable (but this is not done here).

Univariate Nonlinear Models: Inflation

- Explicit time trend: allowed for, but not detected
- $\sin(\text{period}.\text{var} + \text{phase})$ provides a useful range of nonlinear functions

\[
\pi_t = -0.51 + 0.89\pi_{t-1} + 3.26\sin(-0.19\pi_{t-2} + 1.99) - 2.51\sin(-0.28\pi_{t-1} - 4.12)
\]
Univariate Nonlinear Models: Inflation

Multivariate Models

- Lagrange returns N top equations
- Expert may reject top one(s) on theoretical grounds, e.g., explicit time trend coefficient too small to matter
- Next best is then used
Multivariate Models

- Interest

\[ r_t = 6.98 + 0.05y_{t-2} + 8.88 \sin(0.09y_{t-1} - 32.21) \sin(0.05\pi_{t-1} + 14.18) + 23.14 \sin(0.01\pi_{t-2} + 0.01) \sin(0.045t - 1.78) + \epsilon_t \]  

- Output

\[ y_t = 1.60 + 0.07y_{t-2} + 8.84 \sin(0.11y_{t-2} + 1.20) \sin(0.58y_{t-1} + 0.84) + 10.66 \sin(0.56y_{t-1} - 2.02) \sin(0.11y_{t-2} + 0.88) \]  

- Inflation

\[ \pi_t = 0.11 + 0.96\pi_{t-1} + 6.65 \sin(0.60y_{t-1} - 1.02) \sin(0.02\pi_{t-1} - 0.04) - 0.62 \sin(-0.28\pi_{t-1} + 0.71) \sin(0.21t - 1.09) \]  

Multivariate Models: Output
Out-of-sample inflation close-up

Results: Comparison

<table>
<thead>
<tr>
<th>Eqn</th>
<th>In-sample</th>
<th>Out-of-sample</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>MAD</td>
</tr>
<tr>
<td>5</td>
<td>0.49</td>
<td>0.36</td>
</tr>
<tr>
<td>6</td>
<td>0.49</td>
<td>0.36</td>
</tr>
<tr>
<td>7</td>
<td>0.45</td>
<td>0.34</td>
</tr>
<tr>
<td>8</td>
<td>0.56</td>
<td>0.45</td>
</tr>
<tr>
<td>9</td>
<td>0.83</td>
<td>0.64</td>
</tr>
<tr>
<td>10</td>
<td><strong>0.38</strong></td>
<td><strong>0.30</strong></td>
</tr>
</tbody>
</table>
Discussion

- Univariate nonlinear model of inflation is best of all (out-of-sample)
- Multivariate nonlinear model best in-sample: overfit?
- The multivariate nonlinear model switches between substantially different modes when used recursively – and so do economies.

Going Hands On...
Logging on to the Server

1. Login to https://lagramge.cloudapp.net
2. ssh to lagramge.cloudapp.net
3. Use the credentials provided

Running Lagramge

Example: linear grammar, ordinary algebraic equations, depth=5
- Done via runner.py
- Configuration file needs to be setup then fed into the script
- Example
  - runner.py conf/linear.conf
- Results are shown in the web dashboard
Dashboard Screenshots

Get Directions

- Azure management portal
- Screen command line options

Click Get files to get the latest results from the server and show them.

Runner Dashboard is the LAGRAMGE experiment management and analysis dashboard created for MEng project Financial Forecasting and Economic modelling.

Inflation | Real Interest | Output Growth | Reset | Toggle | Grammar Tool

Experiments (1)

<table>
<thead>
<tr>
<th>Results</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDifferential</td>
<td>true</td>
</tr>
<tr>
<td>isValidation</td>
<td>true</td>
</tr>
<tr>
<td>models</td>
<td>Object</td>
</tr>
<tr>
<td>bestMseMld</td>
<td>22</td>
</tr>
<tr>
<td>bestRmseMld</td>
<td>22</td>
</tr>
<tr>
<td>bestMse</td>
<td>0.2028315604798781</td>
</tr>
<tr>
<td>bestMape</td>
<td>62.27068552669868</td>
</tr>
<tr>
<td>dataLength</td>
<td>21</td>
</tr>
<tr>
<td>bestMpe</td>
<td>-0.19201205474173144</td>
</tr>
<tr>
<td>bestMpeMld</td>
<td>2</td>
</tr>
</tbody>
</table>

configuration

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>bestMapeMld</td>
<td>5</td>
</tr>
<tr>
<td>bestRmse</td>
<td>0.45036824985768936</td>
</tr>
</tbody>
</table>
### Understanding the Lagrange parameters

- **Search** – exhaustive vs beam
- **Heuristic** – MSE vs MDL
- **Depth** – Depth of grammar tree expansion
- **Var** – dependent variable
- **Grammar** – Equation defining grammar
- **Time step** – time step for OD discovery
  - or
- **Time var** – Discrete time variable

<table>
<thead>
<tr>
<th>Models</th>
<th>Object</th>
<th>runMAPE</th>
<th>equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>84.73072874741185</td>
<td>upperlimit (0.86189, -5.8481 * r1 * sin (0.794633 * og + 3.49012))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-3.6240344180721715</td>
<td>runMPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.35457</td>
<td>lagrangeMSE</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>113.12600400126784</td>
<td>upperlimit (1.04604, -3.34844 * n * sin (-0.709395 * og + 5.66045)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>47.31736879598851</td>
<td>runMPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.361541</td>
<td>lagrangeMSE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.361541</td>
<td>lagrangeMDL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2757125633506425</td>
<td>runMSE</td>
</tr>
</tbody>
</table>
Changing the Grammar

- Ready-made non-linear grammar
- Any C function can be added to the grammar file, given that the C header is included
- Any C conforming function can be implemented given its dependencies are met

Sample Configuration

```json
{
  "lagramge":{
    "-g": "sample.gramm",
    "-d": 5,
    "-v": "inflation",
    "-s": "exhaustive",
    "-h": "mse",
    "-i": 0.05,
  },
  "runner":{
    "inputDataFile": "data/sample.data",
    "folds": [0.86],
  }
}
```

Prompt Line Command

`runner.py example.conf`
Forecasting the future, plotting the results

- Use Excel to extrapolate equations into the future

- Plot results: easy for the algebraic equations (Excell, gnuplot...), a bit more complicated in the case of ODE.

Bibliography


THE END