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

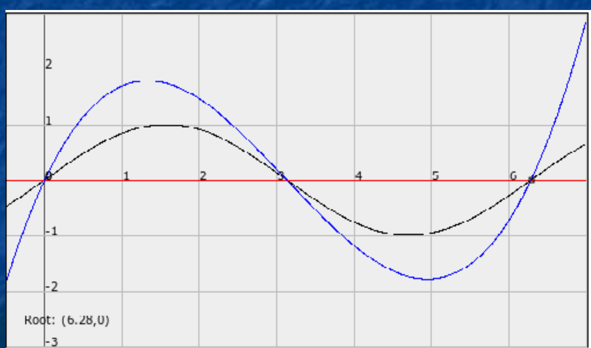
York

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), (*п*, 0), (2*п*, 0)
- *y = 0*, *y =*c_i *sin(x)*, *y =* c_i *x(x-п)(x-2п)* all fit!



- Model range based on experts' knowledge

Case Study: Modelling Inflation

- Empirical modelling of inflation
- ML, equation discovery and Lagramge
- 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: $E \rightarrow T \mid E + T$

  $T \rightarrow c \mid c * Var \mid c * Var * Var$
- Search space $z = a$

  $z = ax + b$

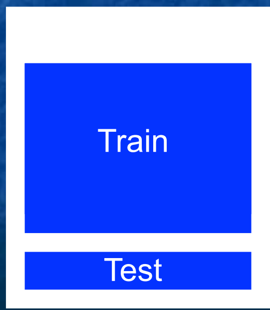
  $z = ax + bxy + cy + d$

etc., but limited by max depth of parse tree.

The Eurozone Data

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

Quarter/Year	π	y	r
1971Q1	5.25	4.22	0.68
1971Q2	5.83	3.56	-0.16
1971Q3	6.09	3.92	-0.14
1971Q4	6.36	3.50	-0.21
⋮	⋮	⋮	⋮
2007Q1	1.87	3.07	1.93



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:
  - *output = f(interest rate, past output)*
  - *inflation = f(output, past inflation)*
  - *interest rate = f(output, past interest rate)*

Setting up Lagramge

- 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 $V_{out} = f(V_1, V_2, \ldots)$ is a sum of terms from this range:

- c_i .
- $c_i . V_i$.
- $c_i . V_i . V_j$.
- $c_i . V_i \sin(c_j . V_j + c_k)$.
- $c_i . \sin(c_j . V_j + c_k) . \sin(c_j . V_j + c_k)$.

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:
infl = 0.221514 + 0.0248229 * t + 0.138563 * intrst * sin (0.0822064 * t + 1.24266) + 0.0155463 * t * sin (0.507664 * intrst + -2.08429) {MSE = 0.00804541, MDL = 0.0253219}

infl = 0.221513 + 0.0248229 * t + 0.0155463 * t * sin (0.507664 * intrst + -2.08429) + 0.138563 * intrst * sin (0.0822064 * t + 1.24266) {MSE = 0.00804541, MDL = 0.0253219}

infl = 0.0301158 * t + 0.0176492 * t * sin (0.479913 * intrst + -1.87172) + 0.157702 * intrst * sin (0.0765666 * t + 1.32732) {MSE = 0.00812278, MDL = 0.0242847}
.
.
.
infl = -0.0940601 + 0.208327 * infl-1 + 0.136085 * intrst * sin (0.075453 * t + 1.26898) + 0.0360579 * t * sin (0.25441 * intrst + -0.117767) {MSE = 0.00932121, MDL = 0.0265977}

Time elapsed: 578.97 [s]

Evaluation

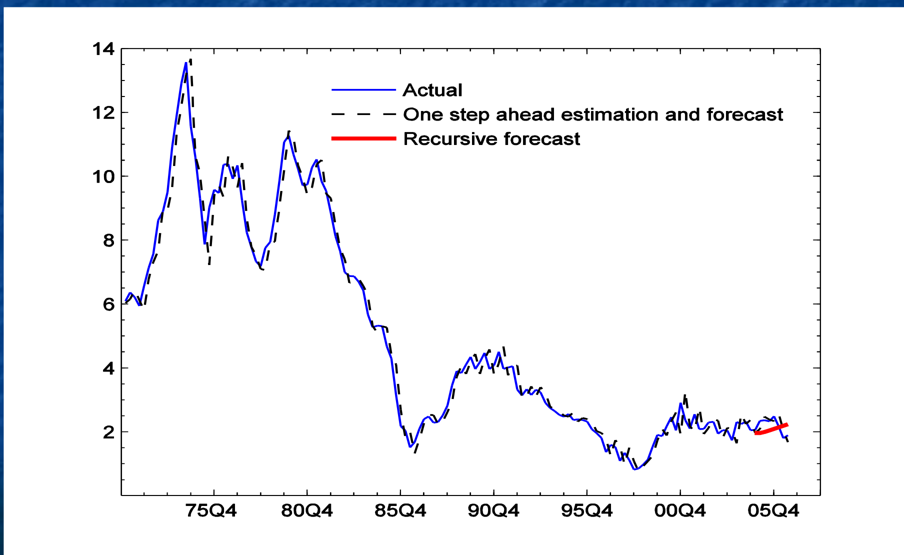
- The evaluation of the model accuracy is performed according to the root mean squared error (RMSE), and mean absolute error (MAD).
- In-sample evaluation: Model's recall measured on training data (up to 2005Q1)
- Out-of-sample evaluation: models' forecast potential tested on unseen test data ('*out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

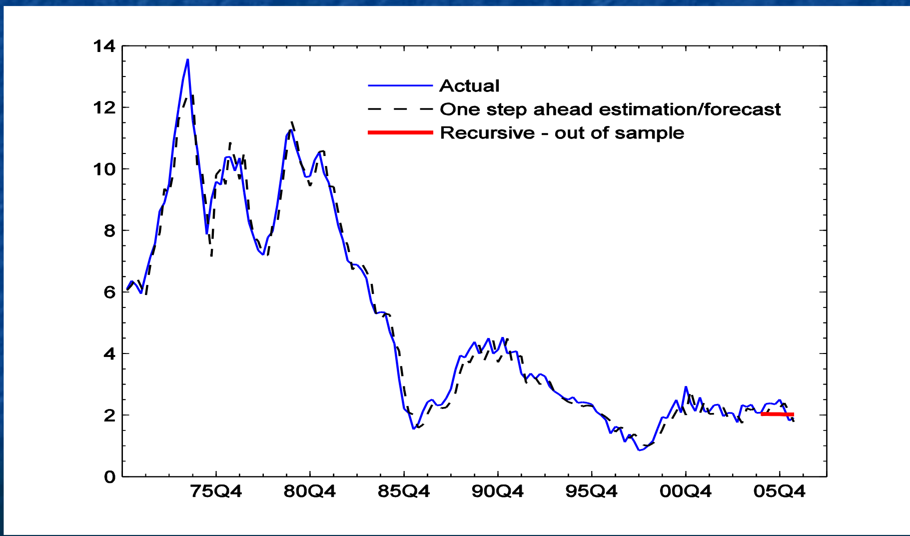
- Recursive forecast: start with the last available observation and apply the model repeatedly to look ahead N steps in the future.
- Sometimes recursive models are converted into equations with time as the only independent variable (but this is not done here).

Univariate Nonlinear Models: Inflation

- Explicit time trend: allowed for, but not detected
- Sin(period.var + phase) provides a useful range of nonlinear fns

$$\begin{aligned}\pi_t \quad = \quad & -0.51 + 0.89\pi_{t-1} + \\ & 3.26 sin(-0.19\pi_{t-2} + 1.99) - \\ & 2.51 sin(-0.28\pi_{t-1} - 4.12)\end{aligned}$$

Univariate Nonlinear Models: Inflation



Multivariate Models

- Lagramge 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

$$r_t = 6.98 + 0.05 y_{t-2} + 8.88 \sin(0.09 r_{t-1} - 32.21) \sin(0.05 \pi_{t-1} + 14.18) + 23.14 \sin(0.01 \pi_{t-2} + 0.01) \sin(0.045 t - 1.78) + \varepsilon_t \quad (8)$$

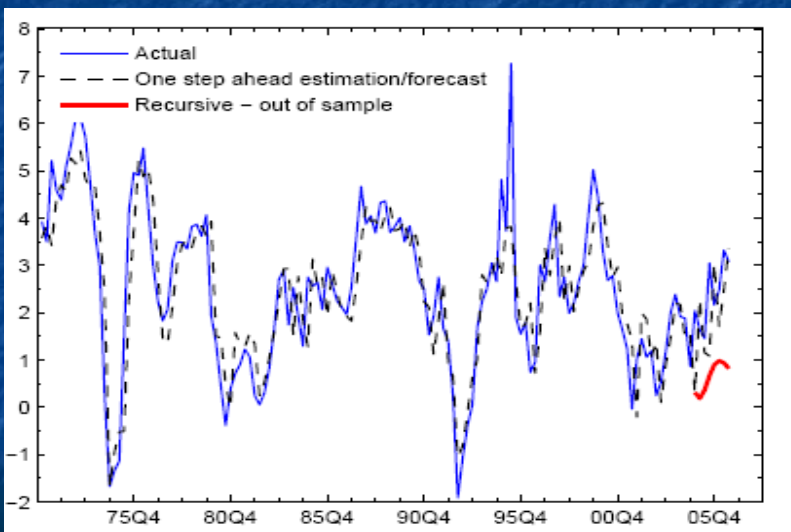
- Output

$$y_t = 1.60 + 0.07 y_{t-2} + 8.84 \sin(0.11 r_{t-2} + 1.20) \sin(0.58 y_{t-1} + 0.84) + 10.66 \sin(0.56 y_{t-1} - 2.02) \sin(0.11 y_{t-2} + 0.88) \quad (9)$$

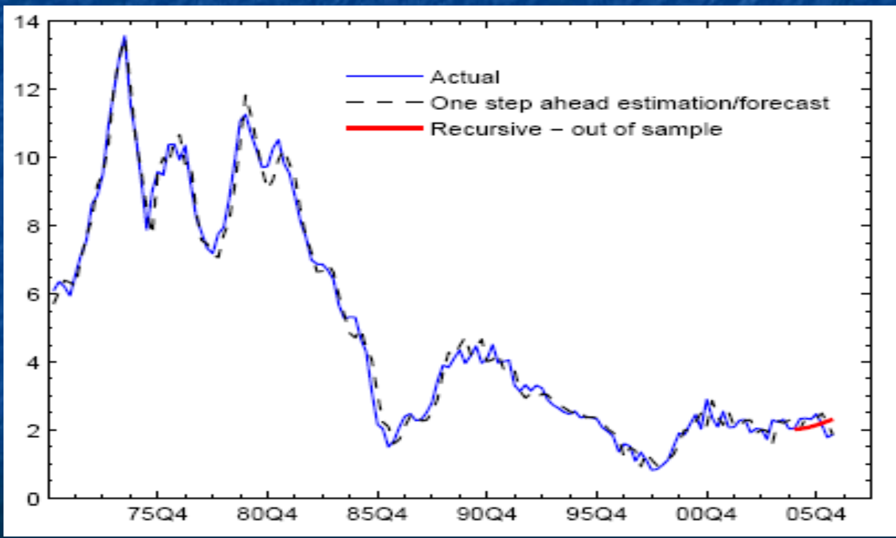
- Inflation

$$\pi_t = 0.11 + 0.96 \pi_{t-1} + 6.65 \sin(0.60 y_{t-1} - 1.02) \sin(0.02 \pi_{t-1} - 0.04) - 0.62 \sin(-0.28 \pi_{t-1} + 0.71) \sin(0.21 t - 1.09) \quad (10)$$

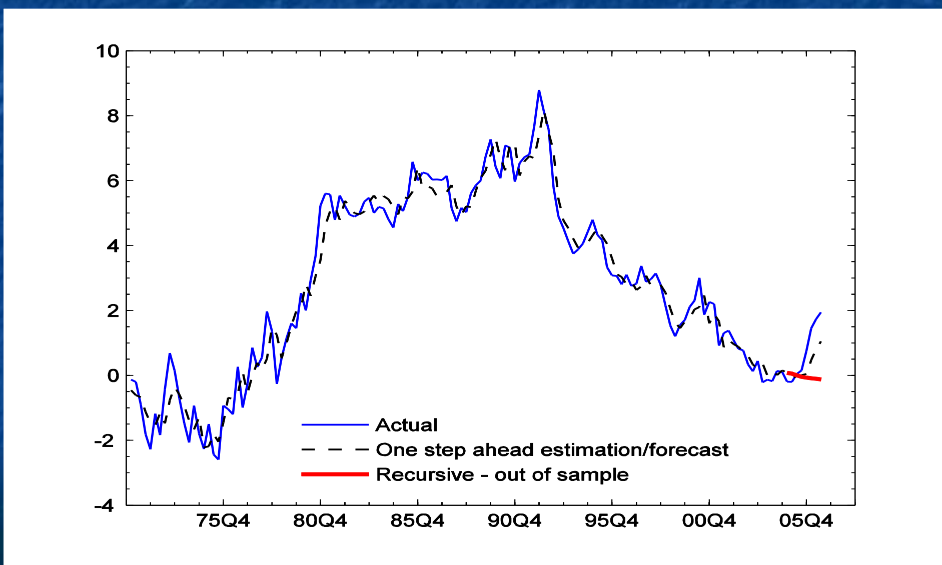
Multivariate Models: Output



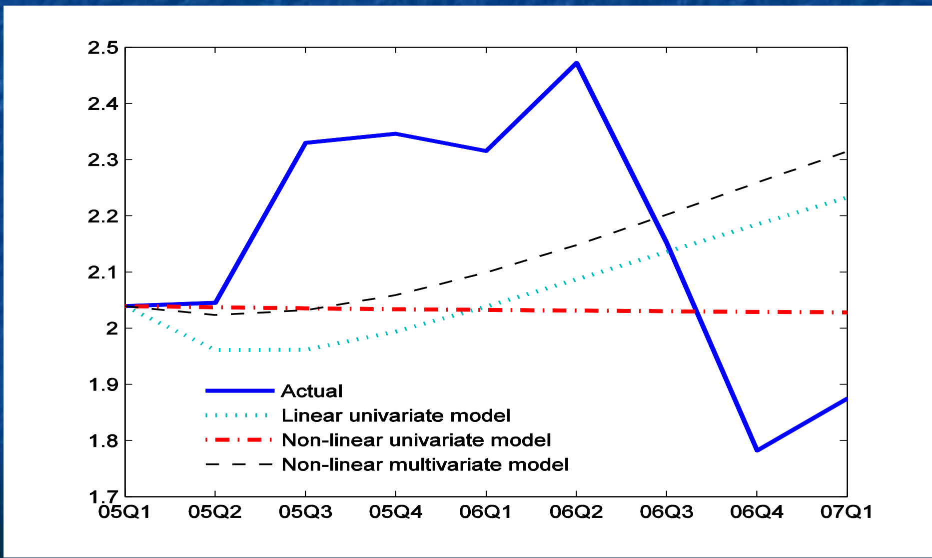
Multivariate Models: Inflation



Multivariate Models: Real Interest



Out-of-sample inflation close-up



Results: Comparison

	Eqn	In-sample		Out-of-sample			
				One step ahead		Recursive	
		RMSE	*MAD*	*RMSE*	*MAD*	*RMSE*	*MAD*
Linear	5	0.49	0.36	0.22	0.19	0.46	0.41
	6	0.49	0.36	0.22	0.19	0.46	0.41
Nonlinear	7	0.45	0.34	**0.20**	**0.15**	**0.26**	**0.23**
Nonlinear multivariate	8	0.56	0.45	0.64	0.52	1.15	0.87
	9	0.83	0.64	1.13	0.95	1.82	1.75
	10	**0.38**	**0.30**	0.23	0.16	0.31	0.26

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

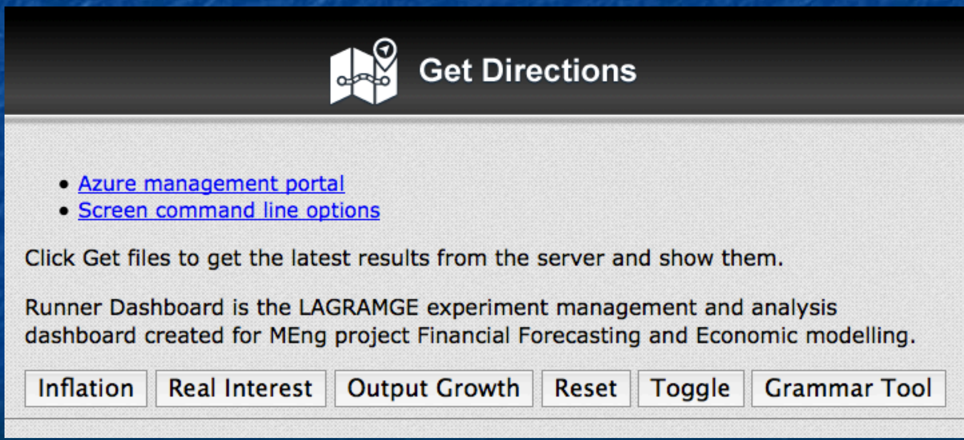
1. Login to https://lagramge.cloudapp.net
2. ssh to lagramge.cloudapp.net
3. Use the credentials provided

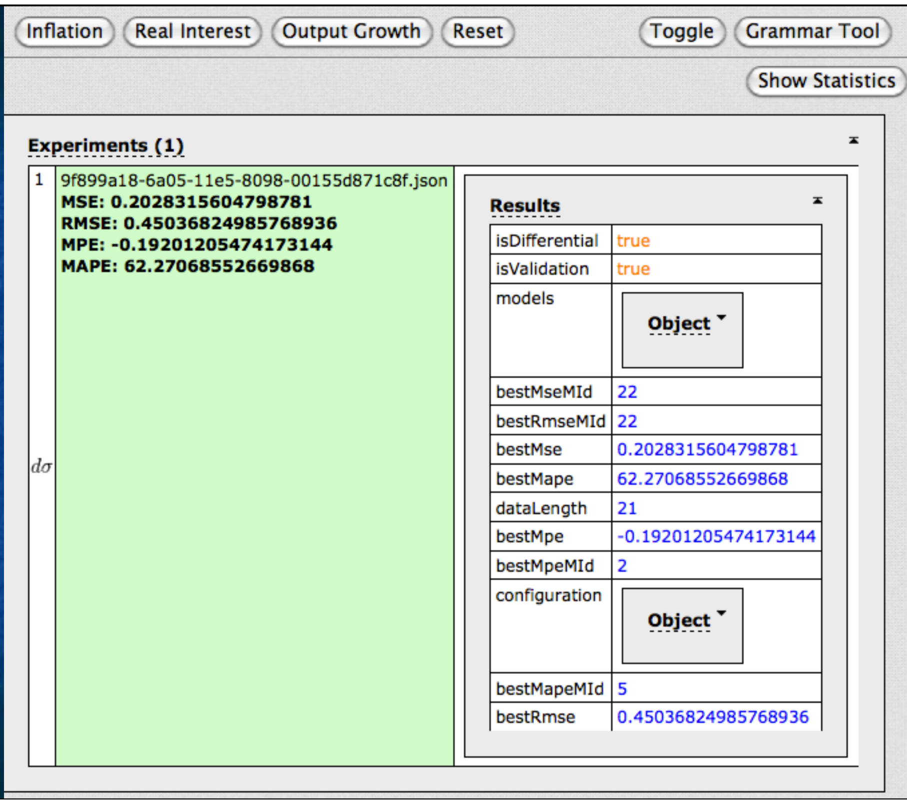
Running Lagramge

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





Results

isDifferential	true
isValidation	true
models	

Object

0

Object

runMAPE	84.73072874741185
equation	upperlimit (0.86189 , -5.8481 * ri * sin (0.794633 * og + -3.49012)) + upperlimit (2.27867 , 2.41542 * og * sin (0.13914 * infl + -0.871535)) + 0.443452 * ri + 0.734752 * infl * sin (0.942451 * infl + -7.64843)
runMPE	-3.6240344180721715
lagramgeMSE	0.35457
lagramgeMDL	0.35457
runMSE	0.2717743020561174
runRMSE	0.5213197694852147

1

Object

runMAPE	113.12600400126784
equation	upperlimit (1.04604 , -3.34844 * ri * sin (-0.709395 * og + -5.66045)) + 1.11054 * ri + 5.82816 * sin (0.339618 * infl + 10) + 1.38092 * infl * sin (-6.83742 * og + -7.47096)
runMPE	47.31736987958891
lagramgeMSE	0.361541
lagramgeMDL	0.361541
runMSE	0.27571256833506425

Understanding the Lagramge 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

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

Learning ODE

Sample Configuration

```
{  "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 exprapolate equations into the future
- Plot results: easy for the algebraic equations (Excell, gnuplot…), a bit more complicated in the case of ODE.

Bibliography

- Ljupco Todorovski and Saso Dzeroski. *Declarative bias in equation discovery*. ICML'97. San Mateo, CA.
- Dimitar Kazakov and Tsvetomira Tsenova. Jan 2009. *Equation Discovery for Macroeconomic Modelling*. International Conference on Agents and Artificial Intelligence (ICAART), Porto, Portugal. http://www-users.cs.york.ac.uk/~kazakov/papers/crc-Kazakov-Tsenova.pdf
- Zhivko Georgiev and Dimitar Kazakov. Dec 2015. *Learning Ordinary Differential Equations for Macroeconomic Modelling*. IEEE CIFEr'15/SSCI 2015, Cape Town, South Africa.

THE END