Forecasting econometric and financial time-series with Dynamic Bayesian Networks and Evolutionary Strategies

in this article, we will be forecasting **CPI** (Consumer Price Index)

LloydL

Abstract: This article concerns genetic dynamic bayesian networks. The genetic algorithm randomly samples integers which correspond to column numbers in the dynamic data.frame. These random samples are all of size 8. Each column corresponds to a time-series that hopefully has some bearing on the future behavior of the response variable, CPI. Each bayesian network so generated will be evaluated according to its predictive accuracy on the test data, the bayesian score for the network as a whole and local arc strength for the parents of the response. The genetic algorithm will mutate and cross-fertilize the best 1000 networks consisting of 8 predictive nodes and a response one gene at a time. The posterior density is multivariate normal with linear means. This article includes the data and software. I have completely illustrated all concepts. In this brief example, we will be trying to model and forecast the economy of the United States. These forecasts are for demonstration purposes only. Past behavior is no sure or certain predictor of future behavior.

Keywords: Forecasting, econometrics, financial forecasting, dynamic modeling, bayesian networks, time-series analysis, machine learning, pattern recognition, models of US economy, genetic algorithms, associative rules.

This software and data are free and comes with ABSOLUTELY NO WARRANTY. You are welcome to modify or redistribute it under the terms of the GNU General Public License (License GPL (>= 2)). For more information about these matters, see http://www.gnu.org/copyleft/gpl.html.

cpiDb is a data.frame that I created from 244 econometric time series from "fred" (US Federal Reserve Database where all time-series dates run from 1991-01-01 to the present) here is a portion of that data.frame. I imputed all missing values with bayesian methods.

Table 1: example from cpiDb

	Unemployment.Rate	CPI	Payroll.Employment	Aaa.Corp.Bond.Yld	Baa.Corp.Bond.Yld
1991-01-01	**6.4**	**5.64706**	**-0.12912**	**9.04**	10.45
1991-02-01	**6.6**	**5.31250**	**-0.63965**	**8.83**	10.07
1991-03-01	**6.8**	**4.82115**	**-0.97406**	**8.93**	10.09
1991-04-01	**6.7**	**4.80993**	**-1.20432**	**8.86**	9.94
1991-05-01	**6.9**	**5.03486**	**-1.44575**	**8.86**	9.86
1991-06-01	**6.9**	**4.69592**	**-1.38362**	**9.01**	9.96
1991-07-01	**6.8**	**4.36782**	**-1.39496**	**9.00**	9.89
1991-08-01	**6.9**	**3.79939**	**-1.18960**	**8.75**	9.65
1991-09-01	**6.9**	**3.39623**	**-1.07925**	**8.61**	9.51
1991-10-01	**7.0**	**2.84858**	**-0.92350**	**8.55**	9.49
1991-11-01	**7.0**	**3.06657**	**-0.83873**	**8.48**	9.45
1991-12-01	**7.3**	**2.98063**	**-0.76672**	**8.31**	9.26

You can have 5 databases with plain English column headers such as Payroll.Employment, instead of arcane symbols and long codes found on Fred. For example, here is the seriesID of real GDP: A191RL1Q225SBEA, where the BEA suffix means: the Bureau of Economic Analysis.)_

Synonymous terms by row:
cpiDb is a collection of time-series from FRED(C), US Federal Reserve of St. Louis.
cpiDb is the raw data for this article.
The dynamic dataframe is the lagged version of the raw data. For example, interest rate inversion often has a latency period of at least one year. I will also refer to it as xx.
time-series, 1 named column in the dynamic dataframe where each column can be identified by either its name or its column number (column numbering in R starts at one not zero)

I now concentrate on forecasting a single dependant variable, CPI for one month in advance. I can use all of the data in cpiDb (from the present back to 01-01-1991). In actuality, I must further concentrate on obtaining a small subset of this vast amount of data. The dataframe: cpiDb, has 342 rows (each row corresponding to measures taken on a particular date). I will sample 8 columns from the dataframe where each column corresponds to a time-series (refer to figure 1.). with samples drawn from the dynamic predictive data frame of the all time-series in cpiDb lagged to a depth of 4. The lags are indicated by '_L(n)'; for example, cpi_L4 means Consumer Price Index lagged by 4 periods (reported monthly) % chg from a year ago, indexed during 1982-1984 at 100. Here is a simple example of an artificial time-series x lagged 5 times, from lag 0 to lag 4:

```
##             X
## 1991-05-01  1
## 1991-06-01  2
## 1991-07-01  3
## 1991-08-01  4
## 1991-09-01  5
## 1991-10-01  6
## 1991-11-01  7
## 1991-12-01  8
## 1992-01-01  9
## 1992-02-01 10
```

```
## [1] 6 5
```

```
##            x_L0 x_L1 x_L2 x_L3 x_L4
## 1991-09-01    5    4    3    2    1
## 1991-10-01    6    5    4    3    2
## 1991-11-01    7    6    5    4    3
## 1991-12-01    8    7    6    5    4
## 1992-01-01    9    8    7    6    5
## 1992-02-01   10    9    8    7    6
```

where time runs from the past at the top to the more recent at the bottom (labeled date: 1992-02-01)

the column labeled x_L0 is the response

order of precedence for lags

month 4	month 3	month 2	month 1	month 0 (the future month)
cpi_L4	cpi_L3	cpi_L2	cpi_L1	response (cpi_L0)

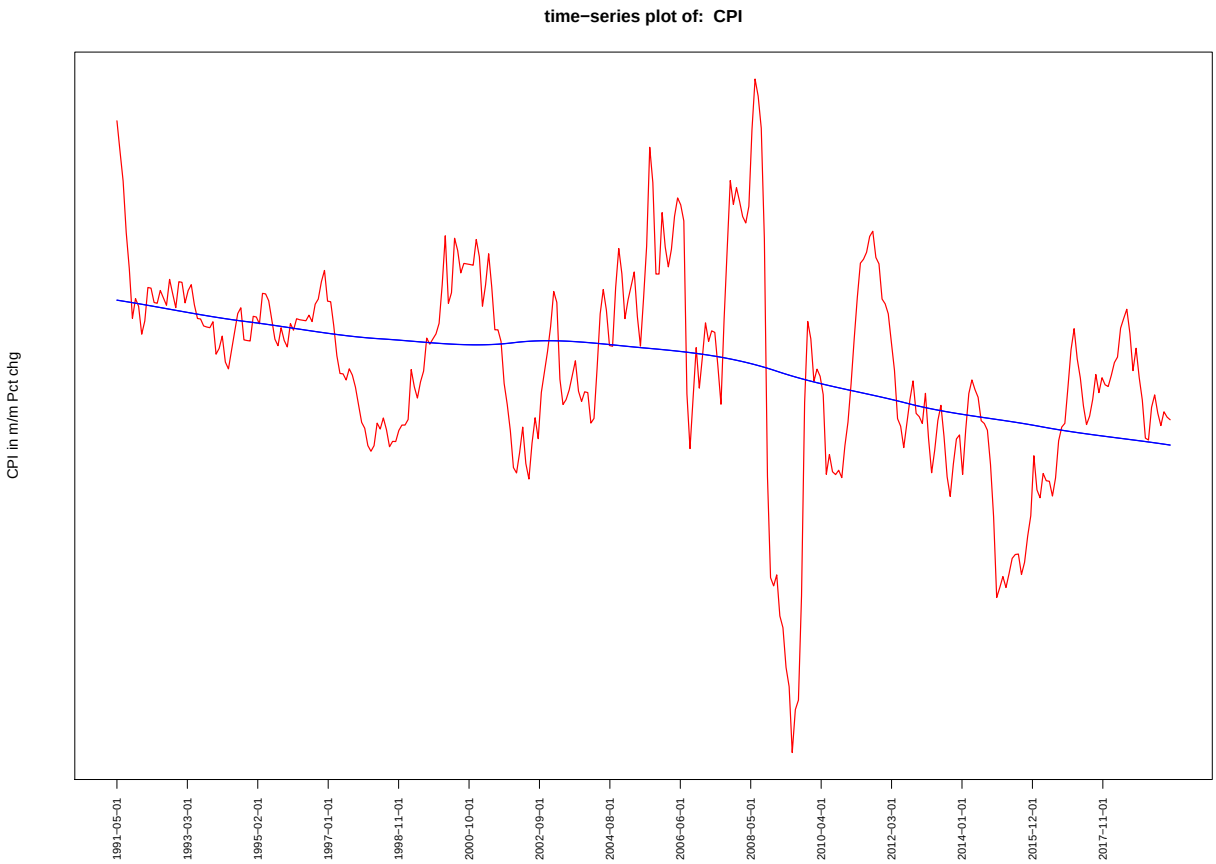
Here is a portion of xx, the predictive dynamic dataframe derived from cpiDb
Note: the genetic algorithm randomly sampled these time-series from xx in subsets of size 8.

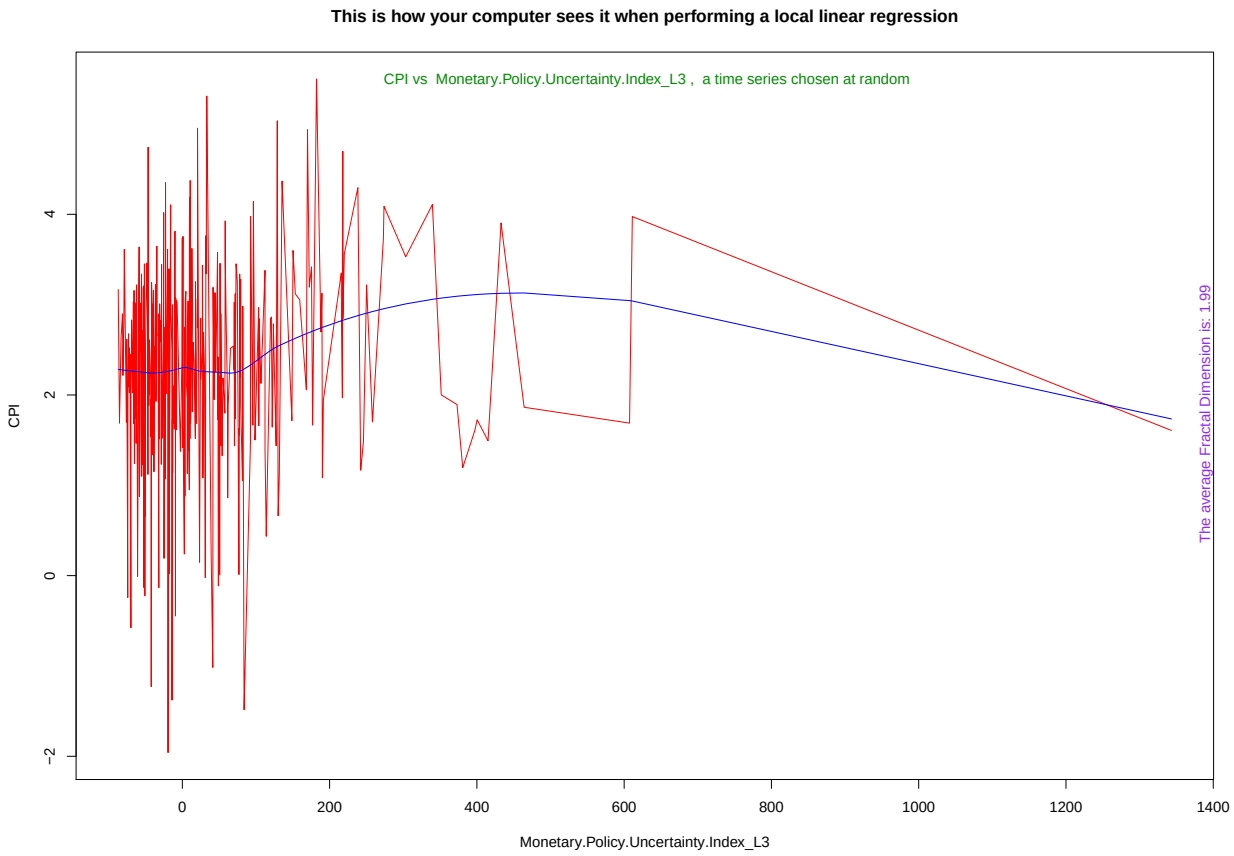
dates	Israel.Exchg.Rate_L4	PPI.Finished.Goods.Les
1991-05-01	**3.60**	
1991-06-01	**3.88**	
1991-07-01	**13.39**	
1991-08-01	**14.48**	
1991-09-01	**17.71**	
1991-10-01	**15.26**	
1991-11-01	**15.02**	
1991-12-01	**14.15**	
1992-01-01	**16.33**	

I wanted to forecast the US Economy based on time series to the US Federal reserve and I wanted to discover something about the flows of information cascading through time perhaps suggesting possible futures the economy could take with each divergent path having a probability.

I will motivate the application of dynamic bayesian networks in econometrics with an exam-

ple:


time−series plot of: CPI
CPI in m/m Pct chg
1991−05−01
1993−03−01
1995−02−01
1997−01−01
1998−11−01
2000−10−01
2002−09−01
2004−08−01
2006−06−01
2008−05−01
2010−04−01
2012−03−01
2014−01−01
2015−12−01
2017−11−01



This typical example demonstrates the futility of applying only linear regression to this data and this forecasting application. The local robust regression algorithm, loess, created the fit (in blue) of the response (cpi)(in red) as a function of time (date). A single multivariate regression is unlikely to capture the more subtle dynamic structures present in the data. * There may be potential for bayesian loess models: Bayesian Treed Gaussian Process Models; Author Robert B. Gramacy. *

In order to prepare for the genetic algorithm, I removed the last 2 rows of xx and the last 2 elements of the response and stored those as xx.test and response.test respectively.

Terminology:

gene means any named time-series in xx which can become a node in the network chromosome means 8 genes that define the predictive nodes in the dynamic bayesian network population means a set of chromosomes, in this case about 15 K after 4 successive filters. I sorted the base population of chromosomes by the average of fwd.error1 and fwd.error2. This is a valid operation because xx.test and response.test have been removed from the data prior to the run. I took the top 2000 fittest chromosomes, corresponding to networks of 8 predictive nodes each.

After re-binding xx.test to xx and response.test to response, I then applied mutation and a crossOver operators to this population. CorConn and parBN.Filter measured the fitness

of each chromosome in each successive population where only the elite survive (quantile 90%). This result is merged with the base population and sorted by totalRock (measure of connectivity). The included showStrength function shows us the 15 fittest dynamic bayesian networks arrived at so far. In at least 15 trials so far, 100,000 chromosomes appears to be adequate.

After repeatedly sampling 8 time-series from the predictive dynamic data.frame, xx and successive surviving populations, we arrive at a data frame where each row corresponds to a random sample (8 column numbers from xx) along with 4 measures of its fitness.

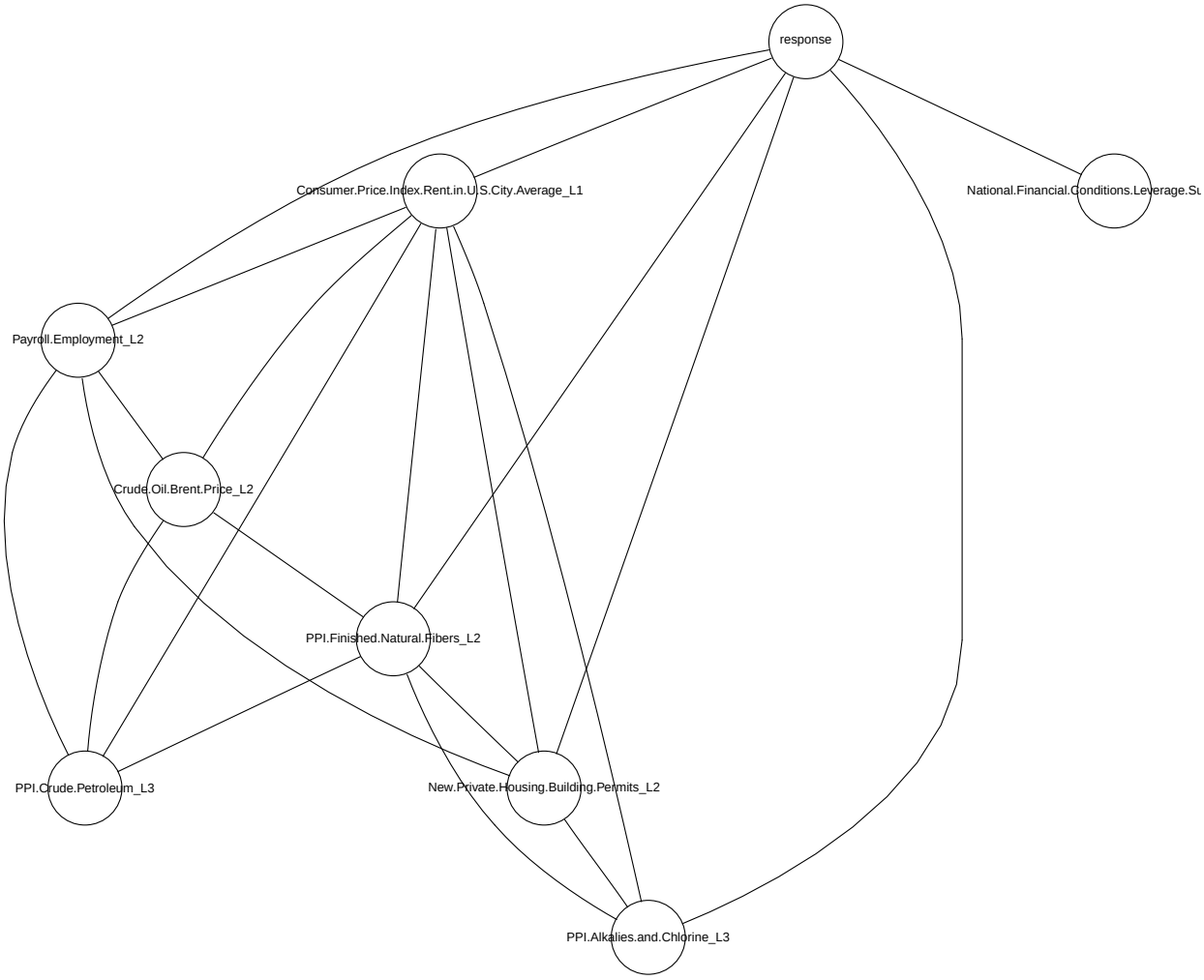
in this data.frame.bayesian.network, or df.bn, we have many primitive measures of interest:

acrynym	concise description
"ss"	randomly selected column number from xx
"totalRock"	measure of network connectivity
"response.cor"	correlation between the response and all other predictor nodes.
"fwd.error1"	out.of.sample 1 - predicted from the 8 predictor nodes from prior periods in time
"fwd.error2"	out.of.sample 2 - predicted from the 8 predictor nodes from prior periods in time
'forecast'	the forecast (see DAG directed acyclic graph)

```
##   ss1 ss2 ss3 ss4 ss5 ss6 ss7 ss8 totalRock response.cor fwd.error1 fwd.error2 foreca
## 1 164 217 225 334 411 537 543 660        27            2 0.05767080 0.00000082 1.6287
## 2  74 122 193 430 483 541 730 788        33            3 0.17723187 0.00001318 1.7872
## 3  14  98 102 179 291 372 391 669        22            3 0.25355336 0.00002079 1.9365
## 4 148 228 303 328 414 474 498 815        34            4 0.06295742 0.00003074 1.7990
## 5   3 159 204 276 439 503 559 852        31            3 0.18474390 0.00004605 1.7379
## 6 307 498 535 732 738 746 812 825        37            4 0.03354581 0.00009091 1.8041
## 7 226 232 287 340 354 413 621 626        21            3 0.06207662 0.00010291 1.9290
## 8 521 558 598 601 697 707 729 731        39            5 0.07656200 0.00010401 1.7394
```

here is the best performing random sample of size 8 from the predictive time-series matrix, xx. with the response time-seriesID: CPI (consumer price index chg % Y/Y) We select for networks with a high degree of connectivity (as measured by the linear correlation). In fact, we select for this feature from the outset.

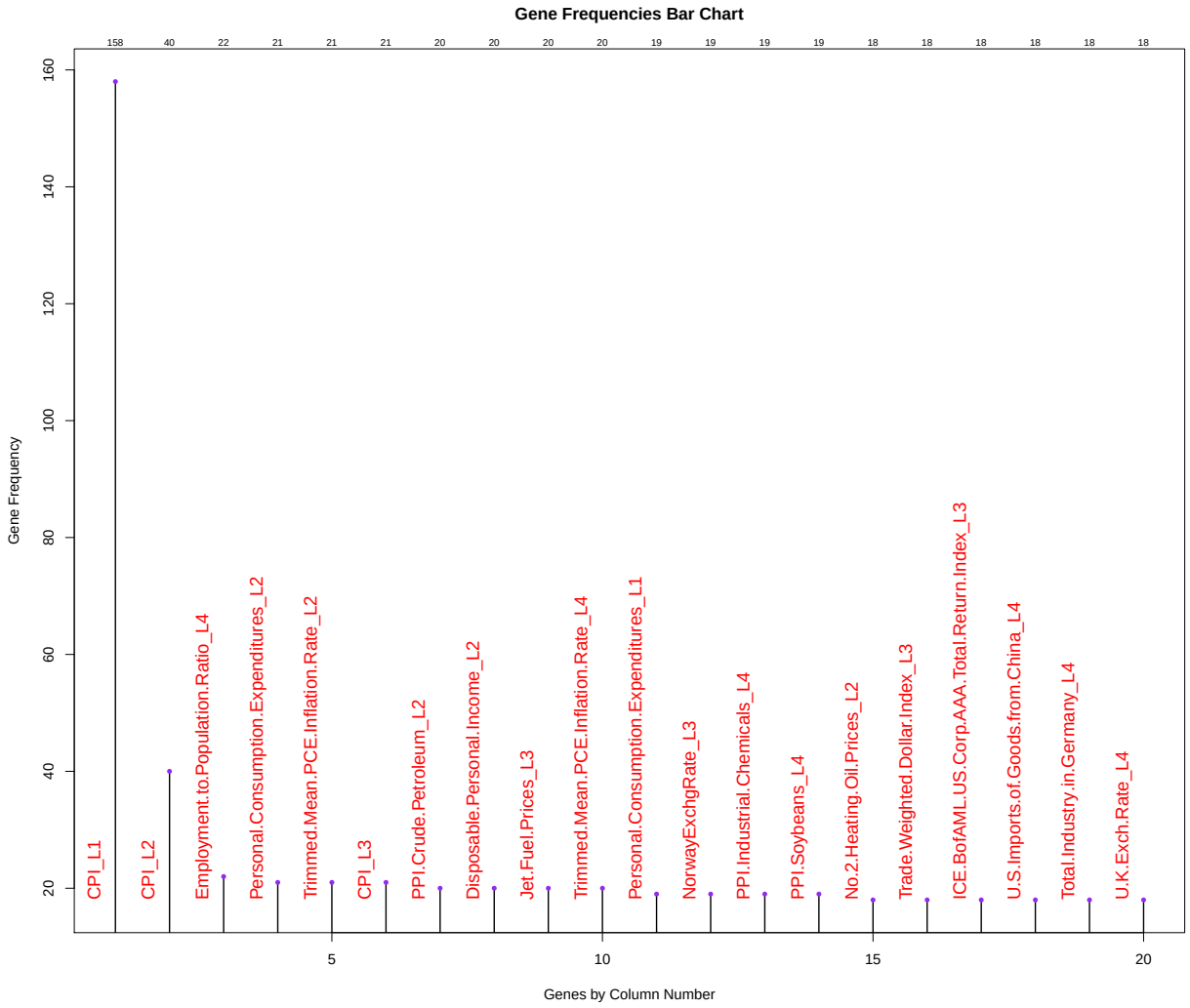
In this article we select networks with both potential for forecasting accuracy and network connectivity (linear correlation and bayes.network.score and total arcStrength for the response and its parents). Below is an example selected by the genetic algorithm. Notice the large number of lines radiating from the response and the high inter-connectivity globally.



here is the fittest random sample (of size 8) drawn from xx, the predictive, dynamic data.frame as selected by genetic algorithm.

	Consumer.Price.Index.Rent.in.U.S.City.Average_L1	
1991-06-01	4.00583	
1991-07-01	3.69833	
1991-08-01	3.60490	
1991-09-01	3.08465	
1991-10-01	3.28571	

here are the most frequently occurring leading indicators in the top 1000 models for forecasting CPI (the response)



here are the most frequently occurring genes (equivalent terms in review: net.nodes ~ colnames(xx)~ genes in a chromosome)

Here is some of the corresponding black list derived from this random sample The black list prevents reverse causation.

here is a panel of dynamic bayesian networks focused on the forecast for:

** Consumer Price Index %chg Y/Y **

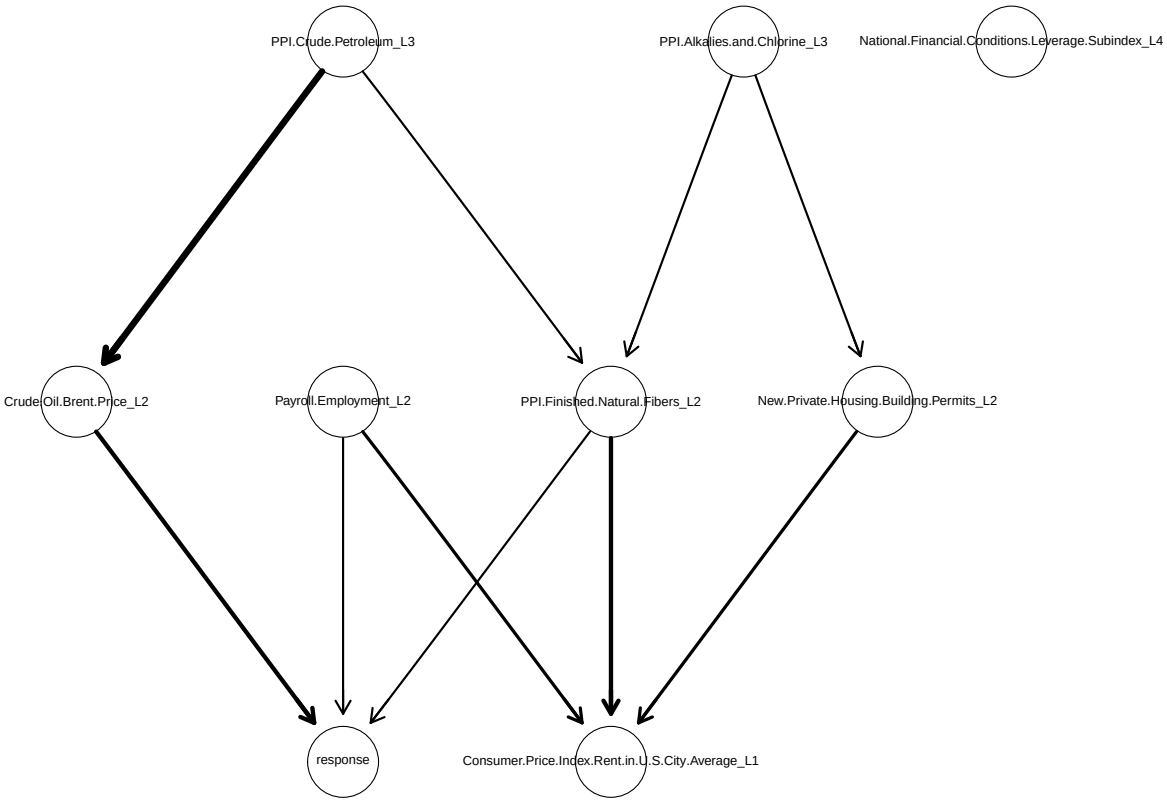
Table 4: Most Frequent Genes (nodes) in the fittest bayesian networks

frequently occurring time series name
CPI_L1
CPI_L2
Employment.to.Population.Ratio_L4
Personal.Consumption.Expenditures_L2
Trimmed.Mean.PCE.Inflation.Rate_L2
CPI_L3
PPI.Crude.Petroleum_L2
Disposable.Personal.Income_L2
Jet.Fuel.Prices_L3
Trimmed.Mean.PCE.Inflation.Rate_L4
Personal.Consumption.Expenditures_L1
NorwayExchgRate_L3
PPI.Industrial.Chemicals_L4
PPI.Soybeans_L4
No.2.Heating.Oil.Prices_L2
Trade.Weighted.Dollar.Index_L3
ICE.BofAML.US.Corp.AAA.Total.Return.Index_L3
U.S.Imports.of.Goods.from.China_L4
Total.Industry.in.Germany_L4

Table 5: black listed arcs to prevent reverse causation

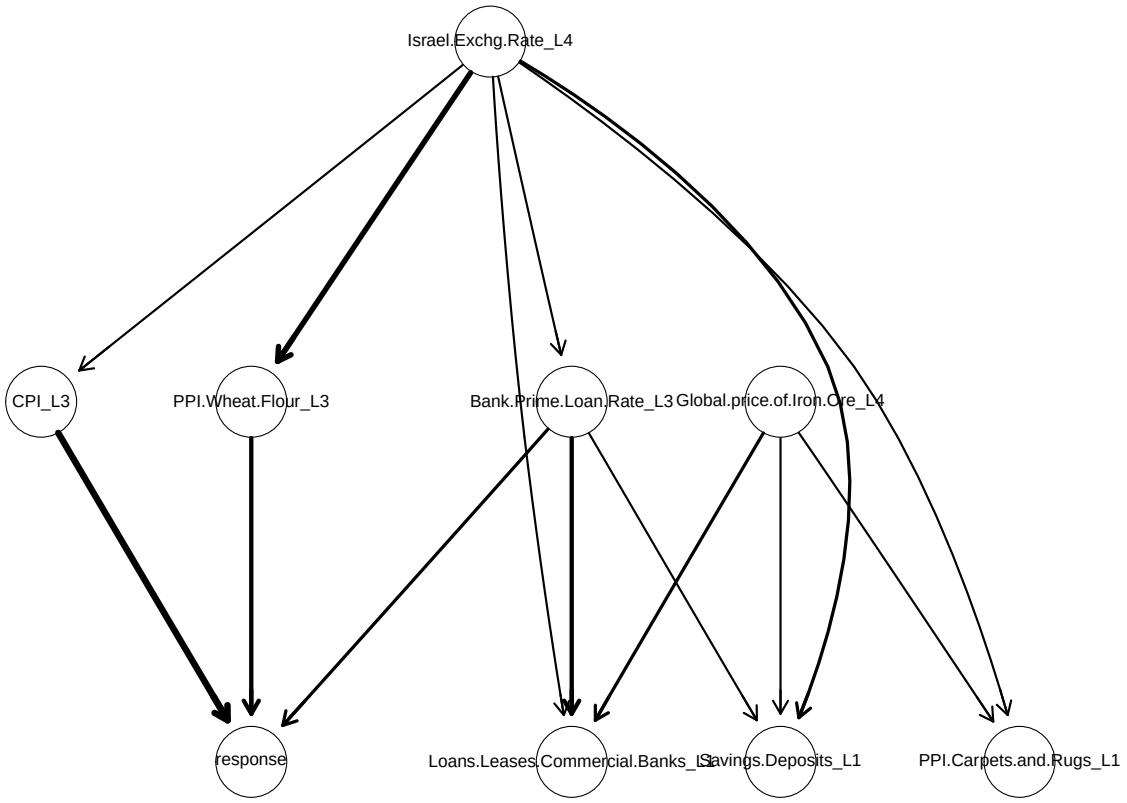
from	to
response	Consumer.Price.Index.Rent.in.U.S.City.Average_L
response	Payroll.Employment_L2
response	Crude.Oil.Brent.Price_L2
response	PPI.Finished.Natural.Fibers_L2
response	New.Private.Housing.Building.Permits_L2
response	PPI.Alkalies.and.Chlorine_L3
response	PPI.Crude.Petroleum_L3
response	National.Financial.Conditions.Leverage.Subindex_
Consumer.Price.Index.Rent.in.U.S.City.Average_L1	Payroll.Employment_L2
Consumer.Price.Index.Rent.in.U.S.City.Average_L1	Crude.Oil.Brent.Price_L2
Consumer.Price.Index.Rent.in.U.S.City.Average_L1	PPI.Finished.Natural.Fibers_L2
Consumer.Price.Index.Rent.in.U.S.City.Average_L1	New.Private.Housing.Building.Permits_L2
Consumer.Price.Index.Rent.in.U.S.City.Average_L1	PPI.Alkalies.and.Chlorine_L3
Consumer.Price.Index.Rent.in.U.S.City.Average_L1	PPI.Crude.Petroleum_L3
Consumer.Price.Index.Rent.in.U.S.City.Average_L1	National.Financial.Conditions.Leverage.Subindex_
Payroll.Employment_L2	Crude.Oil.Brent.Price_L2

Dynamic Bayesian Network for: CPI
forecast ===> 1.629   chg % in CPI (consumer price index) y/y

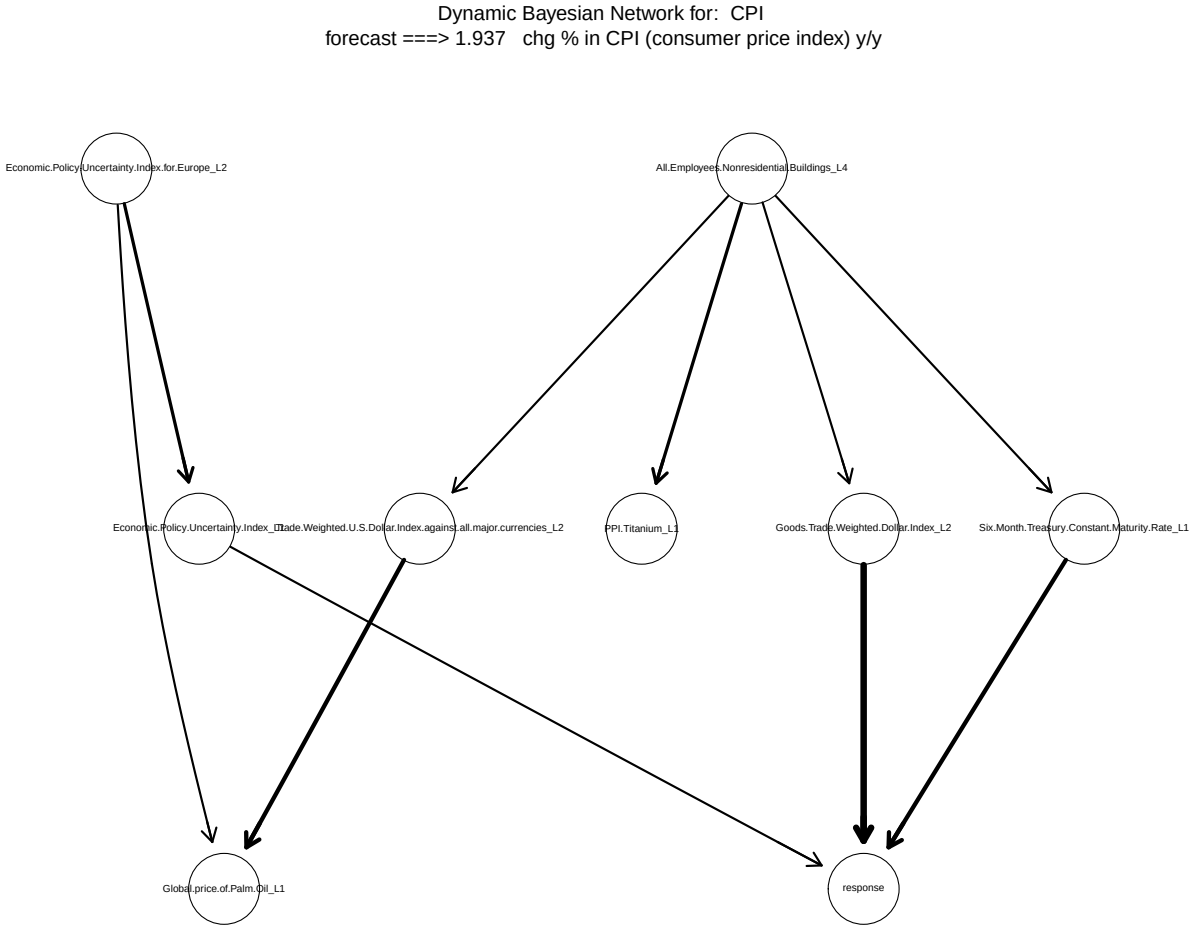


fwd errors (0.058 , 0) ; with mad error 0.49 ; bayes.net.score −11309.2 ; response total arc.strength −76.5

Dynamic Bayesian Network for: CPI
forecast ===> 1.787   chg % in CPI (consumer price index) y/y



fwd errors (0.177 , 0) ; with mad error 0.34 ; bayes.net.score −8955 ; response total arc.strength −83.3




```
## [1] "median forecast:  1.78722257399321"
```

Notes:'

the best selection criteria should be Both total arc strength for the response and lowest fwd errors on test data hidden from the genetic algorithm. In phase one, I established a base population of 100,000 chromosomes (from random samples from the column space of xx of size 8 genes). This run determined the fwd.errors. In phase two, the mutation and crossOver operators played over the base population selecting for total arcStrength, a local metric concerning the parents of the response, in terms of dependency in probability of these local variables. The genetic algorith was elitist, selecting the top 5000 to pass on to the next generation. A better alternative would be a preditor-prey evolutionary strategy which could find a sub-optimal balance between different measures of fitness, this article contains 2 such measure: total net correlation (rotalRock) and minimal foreward errors on missing test data and total Arc strength for the response and its immed. parents. ### depends upon: Package: bnlearn Cre,Aut Marco Scutari a brilliant package Package: RvizGraph