

Analyzing Divisia Rules

Extracted from a Feedforward Neural Network

Vincent A. Schmidt
Air Force Research Laboratory
Dayton, Ohio USA

Jane M. Binner
Aston University
Birmingham, UK

Abstract

This paper introduces a mechanism for generating a series of rules that characterize the money-price relationship, defined as the relationship between the rate of growth of the money supply and inflation. Divisia component data is used to train a selection of candidate feedforward neural networks. The selected network is mined for rules, expressed in human-readable and machine-executable form. The rule and network accuracy are compared, and expert commentary is made on the readability and reliability of the extracted rule set. The ultimate goal of this research is to produce rules that meaningfully and accurately describe inflation in terms of the Divisia component dataset.

Keywords: Divisia, Inflation, Neural Network, Data Mining, Rule Generation

1 Introduction

Government policy-makers aim to provide a stable macroeconomic environment to support economic growth and rising living standards, and the maintenance of a low rate of inflation is crucial for stability. The monetary authorities therefore seek to identify indicators of macroeconomic conditions which will signal impending inflation sufficiently early to allow the necessary action to be taken. Economic theorists have traditionally held the view that a long-run relationship exists between the quantity of money and the general level of prices. Confidence in this relationship, expressed in terms of long-run rates of money growth and inflation, along with an accumulation of evidence sup-

porting a seemingly stable linear demand for broad money aggregates, led the major central banks of the world to accept monetary targeting as the means of controlling inflation. A specific measure of the rate of growth of money stock, known as a monetary aggregate, is derived from the various constituent liquid liabilities of commercial and savings banks. For monetarists the ultimate policy goal of low inflation is achieved by keeping the growth of the chosen aggregate within a target range.

Clearly, the foundations of the construction of monetary aggregates are well rooted in monetary aggregation theory and require extremely strong assumptions. (Barnett and Serletis give a detailed treatment of the theory of monetary aggregation [1].) However, the underlying philosophy of the current research is that all assumptions can be weakened and the Divisia formulation can still be improved. Recent research has focused on accounting for the riskiness of the asset in the index construction; see Barnett et al [2,3] for such efforts in the USA and Drake et al [4] and Binner and Elger [5] for approaches in the UK.

Many tools have been applied to search for relationships between monetary aggregates and inflation. One such tool is the neural network, a trainable mathematical model that tends to be robust with respect to noise and can generalize well under many circumstances. Gazely and Binner successfully used feedforward neural networks to investigate the relationships between UK Divisia M4 assets and inflation, demonstrating that reasonably simple connectionist architectures are expressive enough to prove the existence of such relationships [6]. Their model was designed to examine sensitivity of the relationships, which were specified as weights within the trained network. Additional

effort would be needed to extract and express these relationships.

Our own collaborative research started in 2002. We trained an Aggregate Feedforward Neural Network (AFFNN) using Divisia components and corresponding inflation values in order to evaluate the feasibility of analyzing Divisia data with this model [7]. The AFFNN is a general-purpose feedforward connectionist architecture designed to discover the relationships amongst all network input simultaneously and non-autoassociatively [8]. A decompositional rule extraction algorithm inspired by other researchers [9,10] was designed to operate specifically on the AFFNN. This algorithm produced a collection of MATLAB-based human-readable and machine-executable *if-then* rules expressing the discovered relationships in terms of the original data [11].

The AFFNN was initially chosen for this work because of its existing rule extraction algorithm. Ongoing research resulted in an expanded study of rule generation based on the AFFNN [12], moving past the initial proof-of-concept paper from 2003. The 2004 work revealed potential issues in computational complexity, and a short investigation of these issues was published in 2005 [13]. One goal of the current research effort is to concentrate on the quantity and quality of the rules describing Divisia relationships. Since the full potential of the AFFNN is not necessary for this dataset, complexity can be reduced by retraining the Divisia data using simpler feedforward models, and reimplementing the rule extraction process for these models.

The following sections of this paper describe the selection of an appropriate feedforward model. This includes a description of the encoding used for the monetary component asset data of the UK Divisia M4 and corresponding inflation values. Most importantly, this paper includes a discussion of the decompositional rule extraction algorithm and a limited evaluation of the rules extracted from the selected connectionist model.

2 Dataset Preparation

Historical UK Divisia M4 and corresponding inflation data was obtained¹ in order to investigate the relationship between money supply and inflation. The training data used for connectionist model selection included quarterly seasonally adjusted values from Q1 1977 through Q1 2001, a total of 97 exemplars. Inflation was constructed for each quarter as year-on-year growth rates of prices. Quarterly data over the sample period 1977Q1 to 2001Q1 was used as illustrated in Figure 1. Our preferred price series, the Consumer Price Index (CPI), was obtained from DataStream.

The data was prepared using a series of steps. First, for each category of data, the dataset was recalculated to compute the percentage of increase in value for corresponding quarters in consecutive years. This reduced the dataset to 94 exemplars. Then, an automated clustering algorithm was employed to bin similar (recomputed) values within each category of data together. The number of bins was also determined automatically by the algorithm, developed in Schmidt [8]. Finally, the bins were used to recode the dataset using a thermometer encoding scheme, a common approach for discretizing continuous data for neural network consumption.

Similar data preparation was performed for the inflation values, except the final recoding used a 1-of-N scheme instead of thermometer encoding. This was done because our recent complexity reduction studies suggested that 1-of-N encoding yields fewer rules than thermometer-encoded inflation values.

Table 1 summarizes these results. The name of each category is identified, followed by the symbol used to represent the attribute in the generated rule. The final two columns are the number of clusters and the actual cluster ranges given in (min,max) format. The first five attributes comprise the 39-element binary-valued input vectors, and inflation becomes the four-element binary-valued output. Note that these binary-valued vectors are only used for training and testing the neural models. The final rules extracted from the network will be expressed in

¹Component data is available on the Internet at <http://www.bankofengland.co.uk/mfsd/index.htm> (Bank of England Statistical Abstracts, Part 2, Section A, tables 12.1 and 12.2).

Table 1: Divisia M4 Encoding

Category (Attribute)	Symbol	Levels	Cluster Ranges			
Notes and Coins	NC	7	-0.0333 -0.0243	0.0225 0.0642	0.1604 0.1748	
			-0.0243 -0.0048	0.0642 0.1175		
			-0.0048 0.0225	0.1175 0.1604		
Non-Interest Bearing Bank Deposits	NIBD	14	-0.1273 -0.0918	0.0022 0.0169	0.0826 0.1104	
			-0.0918 -0.0451	0.0169 0.0345	0.1104 0.1543	
			-0.0451 -0.0221	0.0345 0.0497	0.1543 0.1843	
			-0.0221 -0.0100	0.0497 0.0664	0.1843 0.1923	
			-0.0100 0.0022	0.0664 0.0826		
Interest Bearing Bank Sight Deposits	IBSD	4	0.0654 0.1484		0.3069 0.4843	
			0.1484 0.3069		0.4843 0.6542	
Interest Bearing Bank Time Deposits	IBTD	7	-0.0929 -0.0480	0.0574 0.1326	0.3271 0.3573	
			-0.0480 0.0085	0.1326 0.2437		
			0.0085 0.0574	0.2437 0.3271		
Building Society Deposits	BSD	7	0.0062 0.0370	0.0820 0.1114	0.1792 0.1927	
			0.0370 0.0627	0.1114 0.1481		
			0.0627 0.0820	0.1481 0.1792		
Inflation	INFL	4	-0.0033 0.0124		0.0558 0.1116	
			0.0124 0.0558		0.1116 0.1486	

terms of the continuous attribute values (percentage increase).

3 Neural Network Selection

Our previous research has always been based on the Aggregate Feedforward Neural Network (AFFNN) model, primarily due to the availability of an existing rule extraction capability. While this model has been sufficient, the rules were numerous and complex. We believe that continuing the research based on a simpler (and typical) feedforward model will decrease the number and complexity of the generated rules.

We generated a reasonable selection of similar feedforward connectionist architectures as a basis for network selection. Each candidate architecture had the same number of inputs, the same outputs, and contained a single hidden layer. All nodes in the hidden layer used a traditional sigmoid activation function (MATLAB’s *logsig* function) and the unconstrained linear function (MATLAB’s *purelin*) in the nodes at the output layer.

The training data consisted of a randomly selected set of 75 exemplars (80%), and the testing set contained the remaining 19 exemplars (20%). This breakout was chosen to correspond

to the original efforts of Gazely and Binner [6]. The identical data was used for all instances of all networks. For each architecture shown in Table 2, 25 networks with randomly generated initial conditions were trained for 2500 epochs. (Earlier experiments showed no substantial accuracy improvement when trained past 2500 epochs.) The *best network of each class* is shown in the table (based on the accuracy of the testing data)². No instance of model training exceeded 9.25 seconds.

The table’s Architecture column expresses each network architecture in (# inputs–# hidden nodes–# outputs) form. The Training (Testing) column show the number and percentage of correct training (testing) points for all training (testing) data. Since there are four (4) binary-valued outputs in each output vector, we count each output as a separate data point: 75 training cases times 4 outputs per case yields a potential of 300 data points. (For testing, 19 cases times 4 outputs each yields 76 data points.)

The table’s rule generation column reports the estimated maximum number of rules that could be generated using the decompositional

²All model execution was performed on a Slackware 10.1 Linux-based (custom SMP 2.6.13 kernel) dual AMD Opteron 244 system with 2Gb RAM running MATLAB 5.3 (R11).

Table 2: Architecture Comparison (best of 25)

Feedforward Architecture	Training	Testing	# Bins Per Hidden Node	Estimated Maximum # Rules
39-4-4	286/300 (95%)	64/76 (84%)	5 4 2 6	240
39-5-4	298/300 (99%)	67/76 (88%)	5 4 5 2 3	600
39-6-4	300/300 (100%)	64/76 (84%)	8 2 12 2 12 9	41472
39-7-4	300/300 (100%)	64/76 (84%)	2 11 13 2 2 9 1	10296
39-8-4	300/300 (100%)	66/76 (86%)	17 4 9 5 7 11 7 2	3298680
39-9-4	300/300 (100%)	66/76 (86%)	4 7 2 10 12 3 2 11 7	3104640
39-10-4	300/300 (100%)	64/76 (84%)	5 7 2 6 8 8 9 3 13 8	75479040
39-12-4	300/300 (100%)	70/76 (92%)	1 13 15 2 15 12 13 15 10 15 2 2	8.2134e+09
39-16-4	300/300 (100%)	64/76 (84%)	4 5 8 5 2 13 5 10 10 15 8 14 12 6 9 12	1.3586e+14

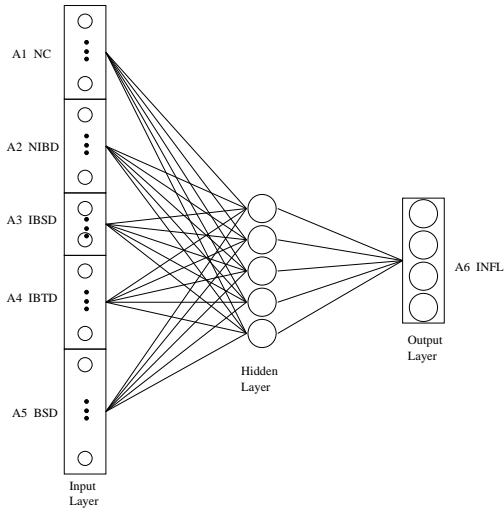


Figure 1: Architecture Selection

extraction algorithm implemented for this system. This number is based on the number of clusters of values produced by the hidden layer nodes (shown in the table as “# Bins Per Hidden Node”), and is similar to the approach described in our previous work and very briefly reviewed in the next section. The values in these columns are displayed as an aid to network selection.

We decided to select the network containing 5 nodes in the hidden layer based on a combination of the testing accuracy and the maximum number of generated rules. This network configuration generates a reasonable number of rules and has a very good training and testing accuracy. Without implying any direct relationship, the Table shows the testing accuracy and maximum number of rules for the best network of each configuration. The selected architecture is also shown in Figure 1 for reference.

Once a candidate architecture has been selected, we re-randomize the training and testing datasets, re-execute the selected model architecture 1000 times, with 2500 epochs of training each, and choose the best instance. For our selected 39-5-4 model, no instance of this architecture exceeded 6.0 seconds of training. Using criteria similar to that used for initial architecture selection, the instance with the best testing accuracy and the fewest potential rules was chosen for actual rule extraction: 296/300 (99%) training accuracy, 70/76 (92%) testing accuracy, and 128 ($2*4*2*2*4$ clusters) estimated maximum rules.

4 Rule Generation

The rule extraction technique applied for this research is a traditional decompositional approach, peering back through the trained network with an emphasis on the values dynamically generated by the hidden nodes. For each hidden node, all values are automatically clustered, and a representative (mean) value is assigned to each cluster. All combinations of these mean values are evaluated against the output node weights to determine the combinations (“candidate expressions”) producing the desired outputs. The sets of candidate expressions are simplified and re-expressed as simple rules in terms of the original network inputs. The rule extraction algorithm was rewritten from scratch for this current research phase, but is based on the original extraction algorithm originally described in Schmidt, and Schmidt and Chen [8, 14].

For the Divisia dataset, the Inflation output is binned into four ranges, with each range

representing the percentage of increase in inflation compared to the corresponding quarter of the previous year. Referring to Table 1, these ranges are:

- Node 1: (-0.0033 ... 0.0124)
(treated as $-\infty$... 0.0124)
- Node 2: (0.0124 ... 0.0558)
- Node 3: (0.0558 ... 0.1116)
- Node 4: (0.1116 ... 0.1486)
(treated as 0.1116 ... ∞)

The rule generator produces rules describing each range separately, so each rule file corresponds to a specific output node, representing a specific range of output values. (The current generation algorithm generously allows boundary conditions between two nodes to be represented in both rulesets.) These rules are expressed in terms of the original input values for readability.

Each file contains many rules, numbered for reference by human readers for convenience. If some combinations of attribute values (nc, nibd, etc.) is described by any rule in a specific file, those values would be expected to result in the "inflation increase %" represented by that node. I.e., all rules in the "node 3" output file describe conditions producing inflation increases in the range (0.0558 ... 0.1116) %.

Each line in a rule is formatted: (low_value <= attr & attr <= high_value) which is, of course, the mathematical equivalent to: low_value <= attr <= high_value. The symbols "&" and "|" are logical "AND" and "OR" operations, respectively, and Inf represents infinity. The logic of the rule must evaluate to "TRUE" for the rule to be true. If a rule does not include an attribute, then that attribute is not required for the given rule.

Figure 2 shows an example of a rule extracted from our trained network. The example clearly demonstrates the human-readable format and nature of extracted rules. This makes them ideal for validation by subject-matter experts. Perhaps more importantly, these rules can also be executed as code and applied to new data.

Table 3 summarizes the key characteristics of the extracted rules. Notice the actual number of rules is different than the estimated maximum number of extracted rules. The maximum

Table 3: Neural Model vs. Extracted Rules

	Neural Model	Extracted Rule Set
# Rules	Estimated 128	Actual 714
Test Dataset Accuracy	70/76 (92%)	70/76 (92%)

number of rules is estimated as a function of the number of training data vectors and their values, but the actual rules are generated based on all possible combinations of input values. (For this encoding of the Divisia dataset, there were 75 training vectors, but the discretization of inputs allows for 19208 distinct input vectors, resulting in a higher number of rules than estimated.) Also, the extraction algorithm relaxes some of the constraints of the original network, allowing some data to potentially fall into a different classification than selected by the corresponding neural model. This can result in slightly different accuracy specifications for the rule set versus the neural network, but the results are generally very comparable. For this encoding of the Divisia dataset, coupled with this specific neural network instance and rule extraction algorithm, the rules report essentially the same results as the trained neural network (with 98.47% accuracy).

The automated rule extractor found 96, 256, 282, and 80 rules for output nodes 1, 2, 3, and 4, respectively, for a total of 714 rules, as reported in the table.

5 Interpretation

The rules from all four generated files were examined by one of the authors, a subject-matter expert in econometrics. Despite being executable as code, the rules were found to be descriptive and easy to read. Interesting patterns were found merely by examining the rule content, and an initial analysis of these patterns is presented here.

The general trend is for interest bearing deposits to have a higher impact on inflation than the non-interest bearing ones. Hence interest bearing deposits and building society deposits are generally found to have a higher "relationship" / impact upon inflation than non-interest

```

if ( (-Inf <= nc & nc <= -0.024308) ...
  & ( (-0.045096 <= nibd & nibd <= -0.010042) ...
    | (0.016850 <= nibd & nibd <= 0.034526)) ...
  & (-Inf <= ibsd & ibsd <= 0.148376) ...
  & (0.132629 <= ibtd & ibtd <= 0.327074) ...
  & ( (-Inf <= bsd & bsd <= 0.062672) ...
    | (0.111375 <= bsd & bsd <= 0.148103)) ...
) return true;

```

Figure 2: Sample Generated Rule

bearing deposits and notes and coins. Put more simply, higher yielding assets are found to have a greater impact on inflation than lower yielding assets. This finding may have more to do with the volume of the asset than its user cost, where user cost is calculated as:

$$\Pi_{it} = P_t^* \frac{R_t - r_{it}}{1 + R_t} \quad (1)$$

where P^* is true cost of living index, approximated by a consumer price index. R is the rate of return of return on asset that yields no monetary services. r is the own-rate of individual monetary asset.

See Barnett [15] and Elger and Binner [5] for a more detailed description of user costs of monetary assets.

The generated rules look appealing from an econometrician's point of view; there is a degree of stability about the results achieved over the cases examined. These rules have potential for shedding new light on movements of inflation, given any specific monetary policy regime in operation at any time. Further experimentation is necessary to determine how the rules correspond with weights derived from user costs. This work would be of tremendous interest for proponents of Divisia money and merits further investigation in future research.

6 Conclusion

The goal in this paper was to choose a candidate architecture and examine the rules generated for the selected instance. The maximum rule quantity estimate used as a part of the architecture selection process provided useful input, but may not have been an accurate portrayal of actual rules, since many rule instances may be simplified after raw extraction. In addition, some rule sets may prove to be more

accurate than others, even when the number of rules is similar. A more comprehensive investigation would include generating rules for all candidates, comparing only the rule accuracy. Since rule extraction is still a computationally (and temporally) expensive process, this investigation must be postponed to a later date.

The selected architecture was successfully mined for interesting and descriptive rules using a newly coded decompositional approach that is much less computationally taxing than the AFFNN-based algorithm the authors previously used. Once again, the rules were encoded to be executable code, capable of processing raw data independently of the original trained neural network. Results demonstrate that the resulting rules are faithful to the original network to a high degree of accuracy.

Although it is beneficial to have executable rules, the principle goal is to use the connectionist model to identify relationships, then express the learned functions in a form useful to subject-matter experts (in the case of Divisia data, econometricians). Based on expert analysis, the generated rules are clearly valuable for describing nuances of the Divisia/Inflation relationship. These early results encourage continued research in this area.

Of course, the ultimate objective is to use this technique as a economic tool for prediction and control of inflation, leading to greater economic stability. Calibration of these results in a large scale macro model would be an interesting route to pursue to determine the full extent of the impact and implications of these rules for the U.K. economy.

We have always intended to compare the results of this research to contemporary methods once the proof-of-concept has been demonstrated. We hope that this phase of the study will finally allow us to formulate a good ap-

proach for such a comparison. Since many other techniques produce equation coefficients instead of specific rules, we are challenged to design an appropriate experiment to ensure the comparison is legitimate, fair, and conclusive.

References

- [1] William A. Barnett and Apostolos Serletis, editors. *The Theory of Monetary Aggregation*. North-Holland, Amsterdam, 2000.
- [2] William A. Barnett. Exact aggregation under risk. In William A. Barnett, Maurice Salles, Herve' Moulin, and Norman Schofield, editors, *Social Choice, Welfare and Ethics*, pages 353–374, Cambridge, 1995. Proceedings of the Eighth International Symposium in Economic Theory and Econometrics, Cambridge University Press. Reprinted in Barnett, W.A. and Serletis, A. (Eds.) (2000), *The Theory of Monetary Aggregation*, North-Holland, Amsterdam, Chapter 10, pp. 195–216.
- [3] William A. Barnett, Yi Liu, and Mark Jensen. The CAPM risk adjustment for exact aggregation over financial assets. *Macroeconomic Dynamics*, 1:485–512, 1997. Reprinted in Barnett, W.A. and Serletis, A. (Eds.) (2000), *The Theory of Monetary Aggregation*, North-Holland, Amsterdam, Chapter 12, pp. 245–73.
- [4] Leigh M. Drake, Adrian Fleissig, and Andy W. Mullineux. Are 'risky' assets substitutes for 'monetary' assets? evidence from an aim demand system. *Economic Inquiry*, 37:510–526, 1999.
- [5] Jane M. Binner and Thomas Elger. The UK household sector demand for risky money. *Berkeley Electronic Press, Topics in Macroeconomics Series*, 2004. <http://www.bepress.com/bejm/topics/vol4/iss1/art3>.
- [6] Alicia M. Gazely and Jane M. Binner. Optimal weights for Divisia aggregation using a neural network approach. In *5th Biennial Conference on Alternative Perspectives on Finance*, Dundee, 2000.
- [7] Vincent A. Schmidt and Jane M. Binner. Determining Divisia rules using the aggregate feedforward neural network. In *Proceedings of the 2003 International Conference on Artificial Intelligence*, Las Vegas, Nevada, 2003.
- [8] Vincent A. Schmidt. *An Aggregate Connectionist Approach for Discovering Association Rules*. PhD thesis, Wright State University, Dayton, OH, May 2002.
- [9] Geoffrey G. Towell and Jude W. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13:71–101, 1993.
- [10] Hongjun Lu, Rudy Setiono, and Huan Liu. NeuroRule: A connectionist approach to data mining. In *Proceedings of the 21st VLDB Conference*, Zurich, Switzerland, 1995.
- [11] Vincent A. Schmidt and C. L. Philip Chen. Extracting rules from the aggregate feedforward neural network. In *Proceedings of the 2002 International Conference on Artificial Intelligence*, Las Vegas, Nevada, 2002.
- [12] Vincent A. Schmidt and Jane M. Binner. Searching for Divisia / inflation relationships with the aggregate feedforward neural network. *Advances in Econometrics*, 19:225–241, 2004.
- [13] Vincent A. Schmidt and Jane M. Binner. Examining the internal complexity of a neural network trained with Divisia component data. In *Proceedings of the 8th Joint Conference on Information Sciences*, Salt Lake City, Utah, July 21-26 2005. 4th International Conference in Computational Intelligence in Economics and Finance.
- [14] Vincent A. Schmidt and C. L. Philip Chen. Using the aggregate feedforward neural network for rule extraction. *International Journal on Fuzzy Systems*, 4(3), 2002.
- [15] William A. Barnett. The user cost of money. *Economic Letters*, 1(2):145–149, 1978. Reprinted in W.A. Barnett and A. Serletis (Eds.) *The Theory of Monetary Aggregation*, North-Holland, Amsterdam, Chapter 1, 2000, 6–10.