Hierarchical Forecasting with Functional Trees

M.Lauretto*†, F.Nakano*, C.A.B.Pereira* and J.M.Stern*

*University of Sao Paulo, Brazil
†marcelo.lauretto@gmail.com

Abstract. This paper presents the use of polynomial networks, synthesized as optimal functional trees by genetic algorithms, in a hierarchical forecasting model.

Keywords: Genetic algorithms, Functional trees, Forecasting, Logistics, Polynomial networks.
PACS: 07.50.Mh, 02.50.-r

INTRODUCTION

This article describes the authors' consulting project for a leading Brazilian magazines editor, nicknamed ABC, and its associated distributor company, nicknamed DE. One of the major logistic challenges of this business is the classic newsstand, newsvendor or newsboy problem, asking for optimal inventory levels. The operations research models for this problem asume fixed prices and random demand, see Hadley and Whitin (1963) and Denardo (1982). The inventory levels are then optimized in order to minimize the costs of being either overstocked or understocked.

The cost of overstock is captured by a well known Brazilian proverb stating that "a day-old newspaper is only good for wrapping fish". Unfortunately, old magazines do not even have that use. In most cases, only the cover page of an old issue is striped and sent back some way along the distribution channel for control purposes, while its body is reprocessed at the nearest paper factory. The immediate cost of understock is lost sales. The long term costs of understock include costumer frustration, possibly leading to permanent phidelity or loyalty transfer to another magazine, low visibility, loss of mind and market share, etc.

The distributor company, DE, has to solve this problem at several hierarchical levels trough the distribution channels, stocking and possibly restocking one or more times from large and small regional depots to individual newsstands. In this article the word newsstand is used as a generic name, encompassing point of sales ranging from street kiosks to supermarket or bookstore shelves.

However, the first step of the newsstand problem has to be dealt with at ABC, planning print (printing or press) runs, that is, the number of copies printed at each batch. Usually a magazine issue stays at the newsstand from one week to one month, and there is no time to reprint an issue.

Most magazines are printed in sections, typically of 16 pages, that are then put together with a cover and bounded. Some of the sections, containing articles and advertisement planned and written far ahead, can be printed in advance, while the cover and others sections, with articles refereeing to current events, are printed in a tight sched-

CP1073, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering—28th International Workshop*
edited by M. de Souza Lauretto, C. A. de Bragança Pereira, and J. M. Stern

ule. This process allows for substantial savings in the production costs, but is a complex operation that requires careful planning.

The optimization aspects of the problem are going to be reported elsewhere; at this article we focus on demand forecasting. The demand is generated by subscribers, newsstands, and a small reserve for the back issue service. The number of subscribers is a relatively stable time series, posing little challenge for accurate forecasting. In contrast, the newsstands demand is very sensitive to current events, specific aspects of an issue, and current marketing efforts.

The forecasting tool developed at this consulting projects uses a two level hierarchical approach. The first level uses a VARMA (vector auto-regressive moving average) model, see Brockwell and Davis (1991). The VARMA model is based on econometric variables like purchase price for subscription and at the newsstand, minimum, average or typical wage or income of the target populations, seasonal effects, number of days in the newsstand, delay between the release date of an issue and typical payday(s), etc. This first level gives good predictions for average sales, but can be improved to more accurately predict local fluctuations.

The second level of the hierarchical model is based on polynomial networks, described in section 2. The second level takes into consideration the qualitative aspects specific to each particular issue like the (quality of the) cover story, cover celebrity, cover photo, editorial content, point of sale advertising, national/regional marketing, promotional gifts, etc.

FUNCTIONAL TREES

Functional tree methods are used for finding the correct specification of a complex function. This complex function must be composed recursively from a finite set, $OP = \{op_1, op_2, \ldots op_p\}$, of primitive functions or operators, and from a set, $A = \{a_1, a_2, \ldots\}$, of atoms. The k-th operator, op_k, takes a specific number, $r(k)$, of arguments, also known as the arity of op_k. We use three representations for (the value returned by) the operator op_k computed on the arguments $x_1, x_2, \ldots x_{r(k)}$:

$$op_k(x_1, \ldots x_{r(k)}) \;, \qquad \begin{array}{ccc} & op_k & \\ / & & \backslash \\ x_1 & \ldots & x_{r(k)} \end{array} \;, \qquad \left(op_k\, x_1\, \ldots\, x_{r(k)}\right) \;.$$

The first is the usual form of representing a function in mathematics; the second is the tree representation, which displays the operator and their arguments as a tree; and the third is the prefix, preorder or LISP style representation, which is a compact form of the tree representation.

As a first problem, let us consider the specification of a Boolean Network, that is, the specification of a Boolean function of q variables, $f(x_1, \ldots x_q)$, to mach a target table, $g(x_1, \ldots x_q)$, see Angeline (1996) and Banzhaf el al. (1998). The primitive set of operators and atoms for this problem are:

$$OP = \{\sim, \wedge, \vee, \rightarrow, \odot, \otimes\} \text{ and } A = \left\{x_1, \ldots x_q, 0, 1\right\} \;.$$

Notice that while the first operator (not) is unary, the last five (and, or, imply, nand, xor) are binary. Also, this set, *OP*, of Boolean operators is clearly redundant. Notice, for example, that

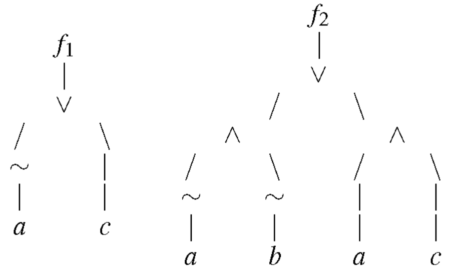
$$x_1 \rightarrow x_2 = \sim (x_1 \wedge \sim x_2)\ ,\quad \sim x_1 = x_1 \odot x_1 \text{ and } x_1 \wedge x_1 = \sim (x_1 \odot x_2)\ .$$

This redundancy may, nevertheless, facilitate the search for the best configuration in the problem's functional space.

x	y	$\sim x$	$x \wedge y$	$x \vee y$	$x \rightarrow y$	$x \odot y$	$x \otimes y$
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	0	0	1
1	1	0	1	1	1	0	0

Figure 1 shows a target table, $g(a,b,c)$. As it is usual when the target function is an experimentally observed variable, the target function is *not* completely specified. Unspecified values in the target table are indicated by the don't-care symbol $*$. The two solutions, f_1 and f_2, match the table in all specified cases. Solution f_1, however, is simpler and for that may be preferred, see section 4 for further comments.

a	b	c	g	f_1	f_2
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	$*$	1	0
0	1	1	$*$	1	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	1	1



$$f_1 = (\sim A) \vee C\ ,\quad f_2 = (\sim A \wedge \sim B) \vee (A \wedge C)\ .$$

$$f_1 = (\vee\ (\sim A)\ C)\ ,\quad f_2 = (\vee\ (\wedge\ (\sim A)\ (\sim B))\ (A \wedge C))\ .$$

Figure 1a: Two Boolean functional trees for the target $g(a,b,c)$.

Starting from a given random tree, one can start a stochastic search in the problem's (topological) space. In Genetic Programming (GP) terminology, the individual's functional specification is called its *genotype*. the individual's expressed behavior, or computed solutions, is called its *phenotype*. Changing a genotype to a neighboring one is called a *mutation*. The quality of a phenotype, its performance, merit or adaptation, is measured by a *fitness* function. GP does not look at the evolution of a single individual, bur rather at the evolution of a population. A time parameter, t, indexes the successive generations of the evolving population. In GP, individuals typically have short lives, surviving only a few generations before dying. Meanwhile, populations may evolve for a very long time.

In GP an individual may, during its ephemeral life, share information, that is, swap (copies) of its (partial) genome, with other individuals. This genomic sharing process is called *sex*. In GP an individual, called a *parent*, may also participate in the creation of a new individual, called its *child*, in a process called *reproduction*. In the reproduction process, an individual gives (partial) copies of its genotype to its offspring. Reproduction involving only one parent is called asexual, otherwise it is called a sexual reproduction.

Sexual reproduction can be performed by crossover, with parents giving (partial) copies of their genome to the children. Figure 1b shows a pair of parents and children generated by a single crossover, for the Boolean problem considered in the last example. The tree representation indicates the crossover points by broken edges (=). Notice that in this example the child corresponds to a solution presented earlier. For further details see Stern (2008) and also Banzahf et al. (1998), Goldberg (1989) and Reeves (1993).

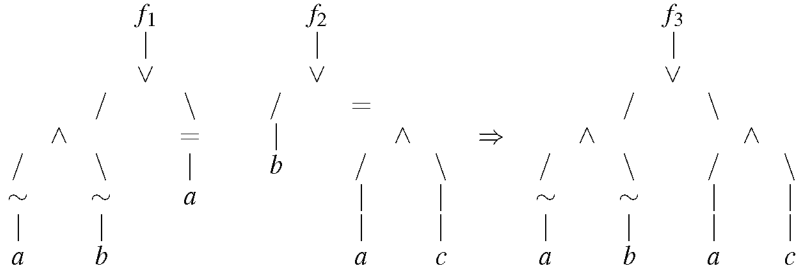


Figure 1b: Crossover between Boolean functional trees.

As a second problem, let us consider Polynomial Network models. These functional trees use as primitive operators Linear, Quadratic or Cubic polynomials in one, two or three variables. Two auxiliary operators are defined as follows: A Normalizer converts the input variable in a new one of mean 0 and variance 1. A Denormalizer converts the network output into a new variable with same mean and standard deviation of the original output variable being modelled.

Figure 2 displays a typical network used for sales forecast used in the consulting project described in sections 1 and 3. Variable x_5 is the magazine's sales forecast obtained by a VARMA time series model. Variables x_1 to x_4 are qualitative variables, in the scale: Bad, Weak, Average, God, Excellent, used to assess the appeal or attractiveness of an individual issues of the magazine, namely: (1) cover impact; (2) editorial content; (3) promotional items; and (4) point of sale marketing. Normalizers at the input edges and a denormalizer at the output edge of the polynomial network are not shown in the figure.

Of course, the optimization of a Polynomial Network is far more complex than the optimization of a Boolean Network: Even having a specified the network topology (identification problem), also the parameters $w_0, w_1, \ldots$ of the polynomial function have to be optimized (estimation problem). Parameter optimization can be based on recursive sub-tree regression; gradient, Partan or conjugate-gradient learning rules, etc. Topology optimization is based on genetic programming and simulated annealing algorithms. For several examples and algorithmic details, see Farlow (1984), Madala and Ivakhnenko (1994), Nikolaev and Iba (2001, 2003, 2006), and Stern (2008).

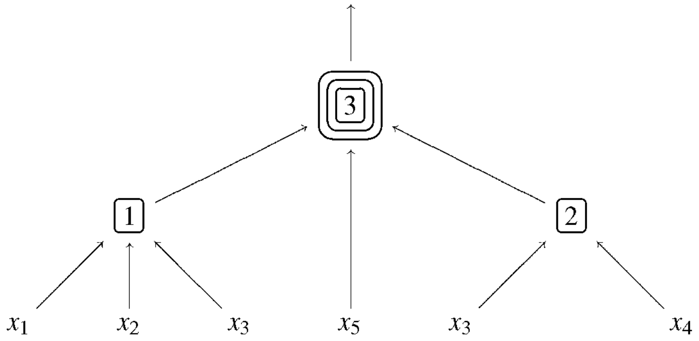


Figure 2: Polynomial Network.
Rings on a node: 1- Linear; 2- (incomplete) Quadratic; 3- (incomplete) Cubic.

EXAMPLE OF FORECASTING

In this section we present a case study based on time series for a magazine published by ABC. Due to confidentiality and disclosure agreements, the time series given in this example has been de-trend and is presented in a relative percentage scale. Forecasts are always made three month ahead with the current past data. From the total of 39 months comprising the time series, the first 28 were used as training data, and the remaining 12 months as test data.

The first level of the hierarchical model consists of a VARMA model, implemented in order to capture trends, seasonal effects and market elasticities, build using automated variable selection procedures, see, Brockwell and Davis (1991). The available explanatory variables include the dates of distribution and recall of each issue, its price for subscription and at the newsstand, minimum, average or typical wage or income of the target populations, typical payday schedules, etc.

Figure 3 presents the actual sales time series and the sales forecasts provided by the VARMA econometric model (top), as well as the models improved by qualitative data, using linear regression (center) and the polynomial network in Figure 2 (bottom). Table 1 shows the average error rates for VARMA, linear regression and polynomial network models. Notice that error rates provided by polynomial networks are smaller than in VARMA and linear regression models. The optimal polynomial network selection is guided by a regularization parameter, ρ, that controls the optimal rate between the penalties for network complexity vs. training error. The default value of the regularization parameter is $\rho = 1.0$. As expected, if ρ is too large, the network becomes too simple, resulting in an underfitted model. At the other hand, if ρ is too small, the network becomes too complex, resulting in a model that is adjusted to the peculiarities of the training data, and has a poor performance at the test data, that is, an overfited model having a low predictive power.

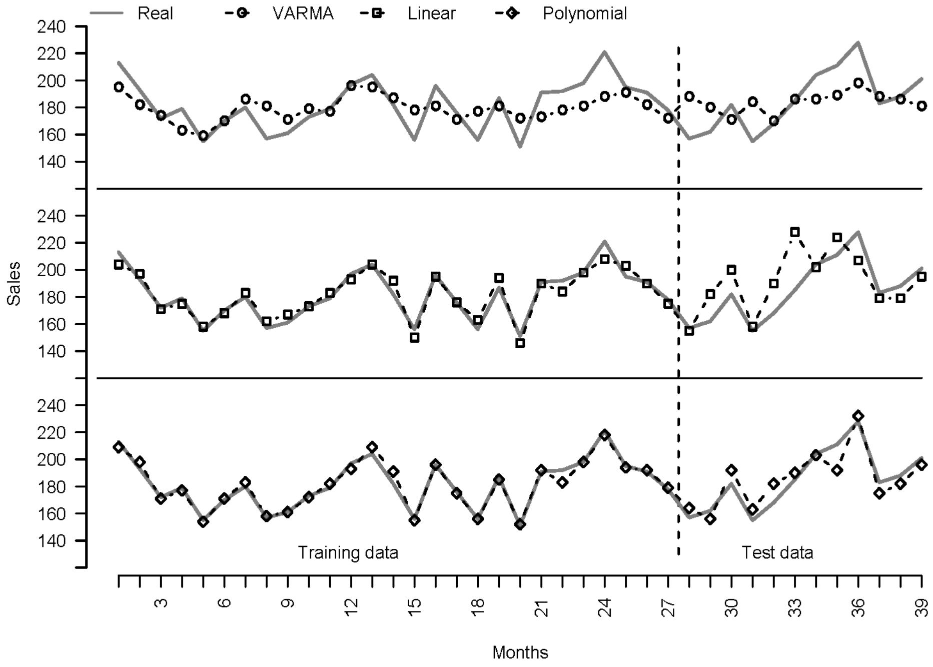


Figure 3: Monthly sales and corresponding forecasts with VARMA models (top), linear regression (center) and polynomial network (bottom).

Model	Training dataset	test dataset
VARMA alone	6.3%	8.6%
+ Linear regression	4.1%	7.4%
+ Polynomial network, $\rho = 0.5$	1.1%	5.1%
+ Polynomial network, $\rho = 1.0$	2.5%	4.2%
+ Polynomial network, $\rho = 2.0$	3.4%	4.6%

Table 1: Error averages provided by VARMA, Linear regression and Polynomial networks.

ENTERPRISE INTEGRATION

As in so many Operations Research projects, solving the mathematical and algorithmic aspects of the optimization and statistical models, and its computational implementation in a user friendly decision support tool, is just part of the entire consulting project. Figure 4 displays some of the Graphic User Interfaces (GUIs). Training the corporate decision makers to use the tools and carefully explaining the concepts involved is also an essential part of the project. All those are prerequisites to the vital goal of integrating the new OR tools into the every day life of the enterprise. Otherwise, the full benefits of the project are never achieved or, even worst, the new fancy tools are soon condemned to oblivion.

ABC is organized in business units according to major target populations like, for example: children, including comic books; male teens; female teens; woman's, including arts, house and garden, gossip, etc.; man's, including cars, computers, sports, swim suite, etc; business and economy; and general news.

ABC's business units were often evaluated by their total sales, market share, and other performance indices that do not take into account production and distribution costs. Meanwhile, DE and the printing plants were often evaluated by their operating costs, regardless of the global company performance. Needles to say, such evaluation metrics generated conflict and misunderstanding inside the company.

The primary objective of the statistical and optimization tools developed in this consulting project was to improve the quantitative fine tuning of the operation, and there the project was very successful. However, the project could also make significant contributions to a secondary objective, namely, to improve the cooperation, integration, rational dialogue and mutual understanding concerning the different roles played by the several agents in such a complex operation. We hope that, in the future, it will also contribute for the development of more encompassive performance metrics, capable of harmonizing and integrating locally conflicting goals into global objective functions.

REFERENCES

W.Banzahf, P.Nordin, R.E.Keller, F.D.Francone (1998). *Genetic Algorithms.*

P.J.Brockwell, R.A.Davis (1991). *Time Series: Theory and Methods* (2nd ed.) NY: Springer.

E.Denardo (1982). *Dynamic Programming.* Engelwood Cliffs: Prentice Hall.

S.J. Farlow (1984) *Self-Organizing Methods in Modeling: GMDH-type Algorithms.* Marcel Dekker, Basel.

D.E.Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley.

G.Hadley, H.M.Whitin (1963). *Analysis of Inventory Systems.* Engelwood Cliffs: Prentice Hall.

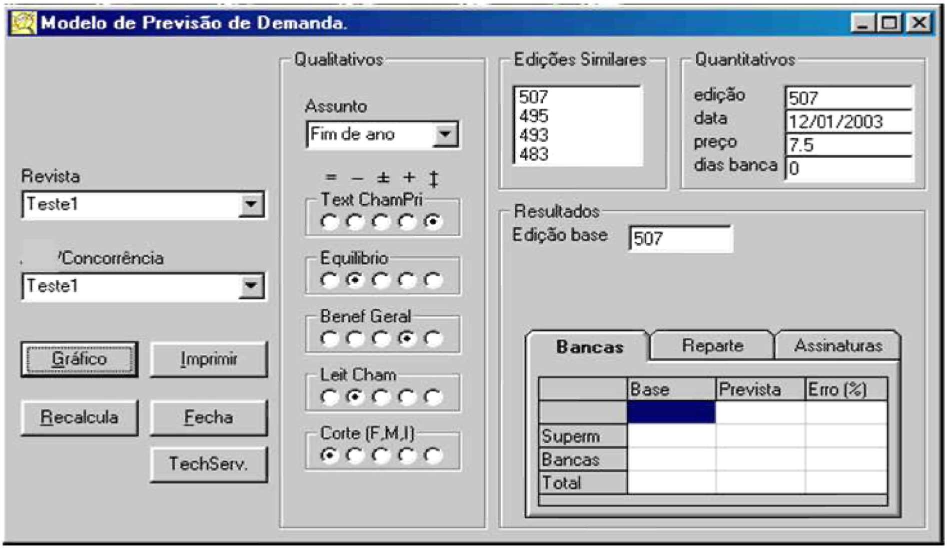
H.R.Madala, A.G.Ivakhnenko (1994). *Inductive Learning Algorithms for Complex Systems Modeling.* CRC.

N.Y.Nikolaev, H.Iba (2001). Regularization Approach to Inductive Genetic Programming. *IEEE Transactions on Evolutionary Computation*, 5, 4, 359-375. Recombinative Guidance.

N.Y.Nikolaev, H.Iba (2003). Learning Polynomial Fedforward Neural Networks by Genetic Programming and Backpropagation. *IEEE Transactions on Neural Networks,* 14, 2, 337-350.

N.Y.Nikolaev, H.Iba (2006). *Adaptive Learning of Polynomial Networks.* Genetic and Evolutionary Computation. NY: Springer.

C.R.Reeves (1993). *Modern Heuristics for Combinatorial Problems.* Blackwell Scientific.



Modelo de Previsão de Demanda.

Revista: Teste1

Concorrência: Teste1

Gráfico | Imprimir | Recalcula | Fecha | TechServ.

Qualitativos

Assunto: Fim de ano

= – ± + ‡

Text ChamPri: ○ ○ ○ ○ ●

Equilibrio: ○ ● ○ ○ ○

Benef Geral: ○ ○ ○ ● ○

Leit Cham: ○ ● ○ ○ ○

Corte (F,M,I): ● ○ ○ ○ ○

Edições Similares

507
495
493
483

Quantitativos

edição	507
data	12/01/2003
preço	7.5
dias banca	0

Resultados

Edição base: 507

Bancas | Reparte | Assinaturas

	Base	Prevista	Erro (%)
Superm			
Bancas			
Total			

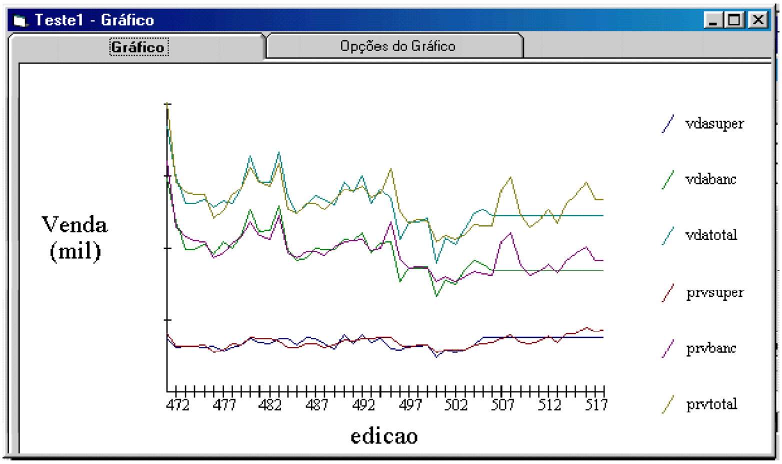


Figure 4: Magazines Forecast System Screenshots;
Top: Simulation interface; Bottom: Forecasts graphs.