Journal of
Applied
Business
Information
Systems

http://www.jabis.ro

Models of estimation, forecasting and computing

Ion Ivan[1], Eugen Dumitrascu[2], Nicolae-Iulian Enescu[3], Gheorghe Marian[4]

[1]Academy of Economic Studies, Economic Informatics Department, Bucharest, Romania
[2]University of Craiova, Faculty of Automation, Computers and Electronics, Craiova, Romania
[3]University of Craiova, Faculty of Automation, Computers and Electronics, Craiova, Romania
[4]University of Craiova, Faculty of Automation, Computers and Electronics, Craiova, Romania

Abstract: *Economic and mathematical models are defined together with their construction methods. It is specified the characteristics of estimation models, forecasting models and respectively computing models and the characteristics of the ways for effective uses in informatics applied domain. It is established ways of analyzing the results of almost close from the estimate to the computing and even the post-computing. It will be done the comparable analysis of the models and will be drawn conclusions about the levels of errors.*

Keywords: *models, estimates, forecasts, computing, post-computing, analysis, errors.*

1. Technology for building models

It is considered a process P to be studied. The factors that influence the structure and evolution of the process P are identified, their number is n, and they are noted F_1, F_2, ..., F_n. The variables X_1, X_2, ..., X_n are associated with factors F_1, F_2, ..., F_n and will be determined their domains of variation, namely D_1, D_2, ..., D_n, the measured level i of the variable X_j is $x_{ji} \in D_i$.

The procedures for measuring the levels of variables are built , so that measurement errors to be insignificant, if they appear they are accepted, whereas to apply the measurements they are eliminated and through elimination from data series are not produced major disruptions to distort the essence of the analysis process P.

When studying the process P the resultative variables Y_1, Y_2, ..., Y_m are identified, they depends on the variation of levels for variables X_1, X_2, ..., X_n named independent variable and the measuring procedures are defined for them.

All measurement procedures for both the resultative or dependent variables Y_1, Y_2, ..., Y_m and the independent variables X_1, X_2, ..., X_n must ensure reproducible results.

If the k measurements are made by two groups independently, will get two tables, each having k lines and m+n columns. The lines correspond to the k measurements and the columns correspond to the m resultative variables Y_1, Y_2, ..., Y_m and the n independent variables X_1, X_2, ..., X_n.

If from a statistically point of view there is no significant differences between the data from both tables, it means was ensured the reproducibility through measurement procedures , i.e.:

- the averages $\overline{X}_i^{(1)}$ and $\overline{X}_i^{(2)}$ of independent variables from the tables (1) and (2) not differ significantly, i=1,2,…,n;
- the averages $\overline{Y}_j^{(1)}$ şi $\overline{Y}_j^{(2)}$ of rezultative variables from the tables (1) and (2) not differ significantly, i=1,2,…,m;
- the dependencies of variables and variation coefficients have levels by showing that there are no major errors in the two tables.

There are many analytical expressions that are used to highlight links between the resultative variables and independent variables.

With the advent of computer techniques were developed software products that:

- stores the tables with data collected for a large number of independent and dependent variables, and also for a much larger number of times for which measurements were made; many of these products go up to consider the unlimited lengths of time series, knowing that time series usually have a few tens of terms, and when allows to store hundreds of thousands of terms, it is considered that already works with a series of unlimited length;
- implements models which correspond to very different analytical expressions, the level of complexity is strictly dependent on the imagination of the analyst and builds a structure model;
- includes very high levels of generality mainly by the dimensions of problems to solve; if it is considered a linear model:

$$y = \sum_{i=0}^{H} a_i \cdot x_i \text{, with } x_0 = 1$$

the generality comes from a certain number H of independent variables;

- generates analytical expressions structures including combinations of independent variables:

  $y_1^{(k)} = a^{(k)} x_k + a_0$ with k=1, 2,..., n

  $y_2^{(k,l)} = a^{(k)} x_k + b^{(l)} x_l + c_0^{(k,l)}$ with $k \neq l$ and k,l =1, 2,..., n

  the combinations of structures that were established with more than 2n+1, strictly dependent on the complexity of linear generator of models; things become more difficult to analyze whether the generators of models include nonlinearities and also variables with delayed argument;
- builds structures of models that allow making many equations to mark dependencies between variables Y_1, Y_2, ..., Y_m and independent variables, obtaining simultaneous equations:

  Y_1= $f_1(X_1, X_2, \ldots, X_n)$
  Y_2= $f_2(X_1, X_2, \ldots, X_n)$
  ....
  Y_m= $f_m(X_1, X_2, \ldots, X_n)$

- implements the various criteria of choice from a set of generated models, that model that meets selected criteria; things get more elaborate if is implemented multicriteria choice;
- refines the models, in the idea that reduces the complexity of analytical expressions, without inducing the level of information's loss from initial model; there are many refined methods presented in [1]:

  a) refinement through variable elimination

  The independent variables X_1, X_2, .., X_n and the dependent variable Y are considered. For the given dataset, the complete model is built, containing all the variables:

  $$y = \sum_{i=1}^{n} a_i \cdot x_i$$

  Coefficients are estimated and the model is assessed using as performance criterion SS, the sum of squared differences between estimated values for the dependent variable and the real values. Also, the coefficient of determination is fit to be used as performance criterion.

  Variants of models are built by removing one by one an independent variable. For each model variant built, coefficients are estimated. The SS indicator is computed and the model list is ordered by it. The first model is chosen its SS indicator value being the smallest among all. This is the result of the 1-refinement process.

The same way, combinations of two independent variables are removed. The number of models obtained is combinations of n taken as 2. Coefficients are estimated and the SS indicator computed for each model. The model with the smallest SS indicator is chosen as the result of the 2-refinement process.
The process continues until the linear model has the form:
$y = a_i*x_i$, i=1, 2, .., n
There is a number of 2^n-2 model variants.

b) refinement through complexity decrease

Reducing nonlinearities refers to the situation in which power terms are replaced by variables, function calls are replaced by variables, function composition is replaced by directly aggregated functions.
The model:
$y=ax^2+bz^2+cu^2+e$ is replaced by $y = Ax+BZ+CU+E$
The model
y=a tg x +b ln u+ e is replaced by *y=Ax+Bu+E*
The model
y = a sin(log x)+b esin u+g by *y = A sin x*log x+B ex*sinx +G*
Reducing nonlinearities simplifies the model and creates the context for easy to observe processings and allow term removal in following refinement steps.

c) refinement through genetic algorithms

For genetic algorithms, which implement the model of population evolution, one important application is symbolic regression. Symbolic regression evolved itself with the introduction of genetic programming and later with the gene expression programming. Starting from a dataset in which it is specified the dependent variable and the independent variables, an initial population of chromosomes is built and then it is subject to a replication process including specific rules.
These algorithms have a very specific way of representing analytical expressions of models. The chromosome is a linear structure of fixed length made up of genes. The role of the chromosome is to code an analytical expression. A gene structure is obtained from the syntax tree of the expression it represents. The nodes of the tree are numbered starting from the root and then level by level and from left to right obtaining a linear structure. Each node contains either an operator, a constant or a variable. Like the nonlinear generator do, the domain for the generated expressions depends on the set of accepted operators and the set of operands built up from variables and coefficients. To create the final expression from the chromosome, the subexpressions derived from genes are aggregated using a simple aggregation function like summation of multiplication.

To build a technology for developing the models means:

- to analyze the process P;
- to establish for what kind of variables, the data are collected;
- to create databases;
- to use a software product;
- to define the selection criteria;
- to obtain general analytical expressions;
- to obtain the criteria of coefficients from analytical selected expressions;
- to use the models and to interpret the results.

Most of the times is chosen model that has the sum of squares of differences between the computed level and real level of dependent variables, from the set of computed sums for generated models.

2. Models of estimation

It is considered a community C made up of elements c_1, c_2, ..., c_r, described by the characteristics of s_1, s_2, ..., s_p. For each characteristic is assigned a variable w_1, w_2, ..., w_p.

It is studied the relationship between variables associated characteristics and result the existence of linear dependence of the form:

$$w_k = \sum_{i=1}^{k-1} a_i \cdot w_i + \sum_{j=k+1}^{p} a_i \cdot w_j + a_0$$

or of a nonlinear dependencies which its variety is extraordinarily high and only the remercabile qualities of the analysts determine a proper model structure compared with the proposed goal.

If using data from the table with measurements for all *r* elements of comunity and for all *p* characteristics, the models are built and the coefficients is estimated obtaining **models of estimation** used in the construction of hypotheses for elements of community, c_{r+1}, c_{r+2}, ..., c_{r+o}, which are to be designed and built, but which must know in advance the cost, the resource requirements, the execution time, the levels of quality.

The estimated levels creates an sufficiently rigorous image but not exactly in the resultative characteristics of a component which will be done in the future, but about its levels for independent characteristics are made the assumptions.

The given levels of estimation rates are meant to guide the decisions such through the creation of variants of combinations for the levels of independent variables, they will be chosen to be convenient in relation to economic criteria.

Estimation models provide information for guidance. The way how to build a component c_{r+i}, i=1, 2, ...o , will finally determine effective levels of the characteristics s_1, s_2, ..., s_p. If in the process of realizing were kept the initial requirements, between estimated and real levels should be no significant differences. But, if the initial assumptions have been abandoned between estimated levels and real levels exist extremely large differences.

The need to elaborate models of estimation is given by:

- need for additional information on levels of aggregate indicators that shape a component of community C, which is in project phase, but it is necessary to be constructed and included in the community;
- accepting that not all information is known about what will happen in the future with the construction process of component c_{r+i}, i=1, 2, ...o and yet one must know, even approximately, costs, timelines, risk levels, quality levels to see the opportunity to develop that variant which corresponds to existing managerial and financial potential at a given time;
- accepting that based on partial information, the estimated models provide levels for resultative variables that will be surely different from the estimated levels and real levels tend to zero; in fact, if there is a measurement order between the estimated and real levels, the quality of estimated model is not questioned.

Nobody will require from a model of estimation to offer levels which overlap 100% with the real levels of characteristics. One is to established the estimated levels of a project for the element c_{r+i} from community C and another to take that element after it has been made and to be measured its characteristics.

It considers the set of software products already implemented PS_1, PS_2, ..., PS_{NS}, consisting of NS components.

These software products exists and for them are registered:

- Halstead complexity levels CP= $n_1 \log n_1 + n_2 \log n_2$, where n_1 represents the number of operands and n_2 represents the number of operators; there are open source products that receive to the input the source text of software products PS_i, i=1, 2, ..., NS, specifying the programming language they were written and these open source products give to the output the complexity levels of NS software products;

- The total registration costs for realize all software products, knowing that salaries were paid, depreciation of equipment were paid for, consumables have been paid, rents, etc.; all costs incurred during the development cycle are into the accounting applications of the organization that developed the software products, so that it registers with absolute precision the total cost CTI of the software products PS_i, i=1, 2, ..., NS. The Table 1 is resulted:

Table 1. Records of the costs and complexity of software products

Software product	Complexity	Total cost
PS_1	CP_1	CT_1
PS_2	CP_2	CT_2
PS_3	CP_3	CT_3
...	...	...
PS_i	CP_i	CT_i
...	...	...
PS_{NS}	CP_{NS}	CT_{NS}

Data analysis is made to see the type and nature of dependence between variables CP_i and CT_i, i=1, 2, ..., NS.

Assuming that the cost of a software product increases exponentially with the complexity, it will build the model:

$$CT = a \cdot e^{b \cdot CP}$$

Using data recorded in the table 1 and a program for estimating the coefficients by the method of least squares to obtain the estimated levels of the coefficients *a* and *b*. So, it is obtained the model of estimation of total cost based on hypothetical level of complexity for any software product that aims to be achieved in the future. Thus, if one wants to build a new software product PS_{NS+1} That is estimated to have a level of complexity similar to another existing software product, c_x, the estimated complexity of the product PS_{r+1} will be $CP_{r+1} = 1{,}2 \cdot CP_x$, which means that the product PS_{r+1} will have an estimated total cost:

$$CT_{r+1} = a \cdot e^{b \cdot CP_{r+1}}$$

If in reality, the total cost of application will be quite different, it is normal to have these differences. One must find the causes which led to the assumptions considered to be less efficient.

3. Models of forecasting

It is considered a process P that evolves over time, with differences at a time t_1, t_2, t_3, ..., t_n, at other times t_{n+1}, t_{n+2}, ..., t_{n+s}, for independent variables X_1, X_2, ..., X_m and the dependent variables Y_1, Y_2, ..., Y_k. The Table 2 is constructed.

Table 2. Variation in time for the variables that characterize the process P

Moments	$X_1, X_2, ..., X_j,...,X_m$	$Y_1, Y_2, ..., Y_h,...,Y_k$
t_1	$a_{1,j}$	$b_{1,h}$
t_2	$a_{2,j}$	$b_{2,h}$
...	...	...
t_i	$a_{i,j}$	$b_{i,j}$
...	...	...
t_n	$a_{n,j}$	$b_{n,h}$
t_{n+1}	$a_{n+1,j}$	$b_{n+1,h}$
...	...	...
t_{n+s}	$a_{n+s,j}$	$b_{n+s,h}$

The part contains data on forecasting models aimed moments $t_1, t_2, ..., t_n$.

It outlines the models and estimates are made using these data.

Using data from the moments $t_{n+1}, t_{n+2}, ..., t_{n+s}$, the dependent variables levels are predicted. Calculate the sums of squares of the differences and choose the most suitable forecasting model.

For software production, the productivity levels are recorded over time as shown in Table 3.

Table 3. Labour productivity in software production

Moment t	Productivity sum lines w
1	200
2	250
3	280
...	...
15	400
16	410
17	450
18	480

The forecasting model is built $w = a \cdot t + b$ and is done estimations using data from the moments t ∈ {1, 2, 3, ..., 14, 15}

Obtain $w = 40{,}2 \cdot t - 35$

It used data from the moments t∈ {16, 17, 18} to calculate the square to see the differences:

$$S = (w_1 - 410)^2 + (w_2 - 450)^2 + (w_3 - 480)^2$$

$$w_1 = 40{,}2 \cdot 16 - 35 = 608{,}2$$

$$w_2 = 40{,}2 \cdot 17 - 35 = 648{,}4$$

$$w_3 = 40{,}2 \cdot 18 - 35 = 688{,}6$$

$$S = (602{,}8 - 410)^2 + (648{,}4 - 450)^2 + (688{,}6 - 480)^2$$

$$S = 37171{,}84 + 39362{,}56 + 43513{,}96 = 120048{,}36$$

The sum of squares has a very high value compared to the productivity of each moment.

The forecast model keeps the performance if and only if it is an indication of assumptions regarding the dynamics of the independent variables.

In the applied informatics is forecast:

- the average complexity of software products;
- the costs of programs;
- the levels of performance for quality characteristics such as reliability, maintainability;
- number of developers;
- the amount of software makers at national level;

- the profit level;
- the level of training effort generated by the implementation of products at level of entire company.

It is important to have models, datasets and especially estimates. Equally important are the assumptions used in developing variants of the forecast data sets.

Forecasts are needed to underlie decisions relating to:

- the way how the experts are trained;
- allocation of funds in the development of new types of software;
- resizing software development phases by increasing the number of experts working in certain phases considered poor, phases that unduly prolonging the terms.

4. Models of computing

These models refer to existing new programs, just launched in exploitation .

They are known all charges were made at each stage.

The model of computing for cost CC is:

$$CC = \sum_{i=1}^{NE} CE_i$$

where:

NE – the number of stages of developing cycle

CE_i – total expenditure on stage i

The computing model aims:

- the documents that include costs incurred into company by all people involved in developing software for salaries, consumables, for use of equipment and tools to assist the development process;
- key distribution overheads at the division level, organization level in order to obtain a more complete picture of the costs incurred by creating the framework conditions.

Computing models are meant to establish exactly what characterizes the software product just released to utilisation .

After a number of runs NR of a program is recorded NS number of program runs successfully, which allows calculation of the actual effective reliability FE:

$$FE = \frac{NS}{NR}$$

If the program is run NR times to record the number of transactions and transaction times or duration, as in Table 4.

Table 4. Recording of transactions and the duration

Running	Tranzaction no. NT	Transaction time DT
1	NT_1	DT_1
2	NT_2	DT_2
3	NT_3	DT_3
...	...	...
NR	NT_{NR}	DT_{NR}

Is calculated:

- average number of transactions;
- average duration of a transaction
- processing stability, using the coefficient of variation.

Computing models are designed to create an effective picture of the concrete software in recently exploitation.

To calculate the real duration of software product software DR, using the formula:

$$DR = \sum_{i=1}^{L} DS_i + \sum_{i=1}^{H} \max_{1 \le j \le M_H} \{DP_{ij}\}$$

where:

L – number of activities that are executed sequentially
H – number of chains of activities that run in parallel
DS_i – duration of activity i that is executed sequentially
DP_{ij} – duration of activity j belongs to the chain i

For a software product SOF whose graph is given in Figure 1, the duration of realizing DR is determined using the relationship as follows:

$$DR = DS_1 + DS_2 + \max\{DP_{11}, DP_{12}, DP_{13}, DP_{14}\} + DS_3 + DS_4 + DS_5 + \\ + \max\{DP_{21}, DP_{22}, DP_{23}, DP_{24}\} + DS_6 + \max\{DP_{31}, DP_{32}\}$$

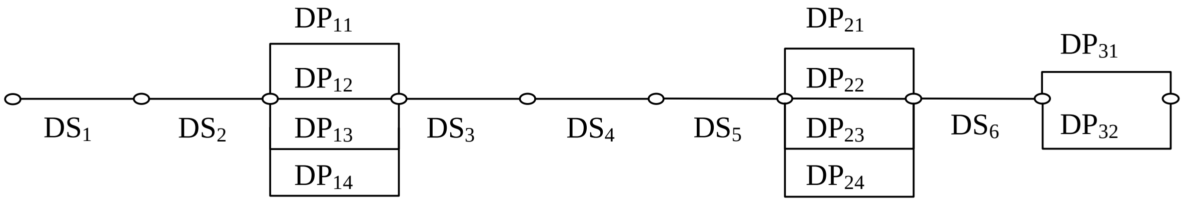

Figure 1. The graph of the performance of SOF program

Also is calculated the amount of effective processing of software product, running on the control structures implemented.

For program:

```
S = 0;
for(i=0; i<n; i++)
          S += x[i];
xm = S/n;
```

The amount of processing VP = 2 + 2n because the instructions S = 0 and xm = S/n runs only once while the instructions for() and S += x[i] runs for n times.

The model of computing for program length LP, is given by:

LP = LD + LC + LE + LB + LK

where:

LD – number of instructions defining the operands
LC – number of comment lines
LE – number of executable instructions
LB – number of delimiters for blocks
LK – number of instructions for defining the included libraries and to define conditional compiling options

The model of computing the memory areas concerns the restrictions that must appear in connection with existing operators and operands.

If the program has a tree like structure or graph structure is calculated maximum length Lmax occupied by memory modules that are simultaneously being executed.

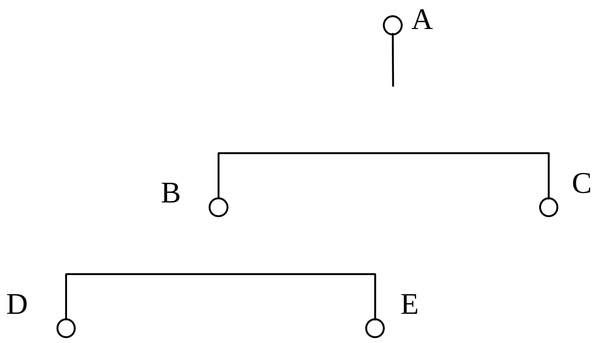

Figure 2. Tree structure

For the program from Figure 2, the maximum length is:

Lmax = max{A + B + D, A + B + E, A + C}

The software product that running has a source text for which a McCabe complexity C_C is calculated by the formula:

C_C = NA – NN + 2

where:

NA – number of graph arcs associated with program

NN – number of nodes of the graph associated with program

For the SPR sequence of the program:

```
S=0;
Sp=0;
for(i=0; i<n; i++){
        S += x[i];
        Sp += x[i] * x[i];
}
if (Sp > 3 * S * S)
        e = 1;
else
        e = 2;
k = 3;
```

is associated the graph given in Figure 3.

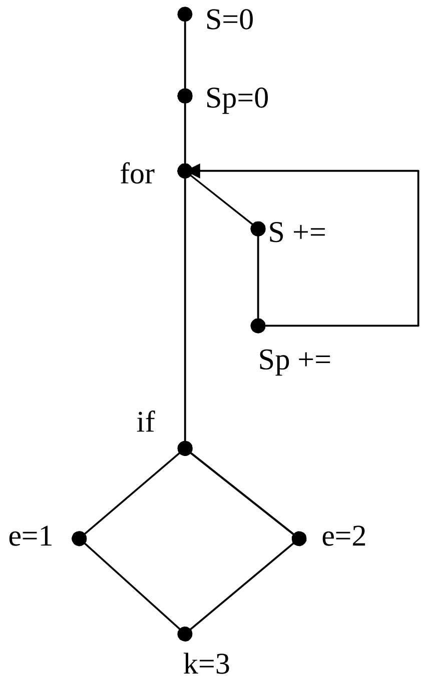

Figure 3. The graph associated to the program SPR

Number of nodes is NN = 9, number of arcs in NA = 10, and the complexity C_C=10-9+2=3 shows a medium level of complexity, whereas for simple programs, C_C = 1.

5. Conclusions

Modeling is an activity that combines science and art. For a phenomenon, a process or a product is built countless models, all strictly dependent on the assumptions used in construction.

There are relationships between models of estimation ECA, models of forecasting PCC, models of computing CCC.

These differences are not weaknesses of these models, but there are differences in approach levels that are less information when the software product is making plans, there is much more information for the already existing product, which is in current use and decreases when making forecasts.

It only must to amplify the effort to increase the volume of information and especially to improve the working hypotheses.

6. References

[1] Ion Ivan, Adrian Visoiu - **A Comparative Analysis of Software Refinement Techniques**, Proceedings of International Multi-Conference on Engineering and Technological Innovation, June 29th - July 2nd, 2008, Orlando, Florida, USA, Organized by International Institute of Informatics and Systemics, ISBN: (13) 978-1-934272-43-5, pp 235-239

[2] Adrian Vişoiu - **Performance Criteria for Software Metrics Model Refinement**, Journal of Applied Quantitative Methods, Volume 2, Issue 1, March 30, 2007

[3] Ion Ivan, Adrian Vişoiu - **IT Project metrics**, Journal of Applied Quantitative Methods, Volume 2, Issue 3 - September 30, 2007

[4] Ion Ivan, Gheorghe Nosca, Marius Popa - **Managementul calitatii aplicatiilor informatice**. Editura ASE, Bucuresti, 2006.

[5] Ion Ivan, Cătălin Boja - **Managementul calităţii proiectelor** *TIC,* Editura ASE, Bucureşti, 2005, ISBN 973-594-558-4

[6] Ion Ivan, Adrian Vişoiu - **Baza de modele economice**, Editura ASE, Bucureşti, 2005, ISBN 973-594-571-1
[7] Cătălin Boja, Ion Ivan - **Metode statistice în analiza software**, Editura ASE, Bucureşti, 2004
[8] Pankaj JALOTE: **Software Project Management in Practice**, Addison Wesley, 2002