International Journal of Software Engineering
and Knowledge Engineering
Vol. 16, No. 1 (2006) 127–140


World Scientific
www.worldscientific.com

# A COMPARISON OF EFFORT ESTIMATION METHODS FOR 4GL PROGRAMS: EXPERIENCES WITH STATISTICS AND DATA MINING

JOSE C. RIQUELME[*,‡], MACARIO POLO[†,§], JESUS S. AGUILAR–RUIZ[*,¶]
MARIO PIATTINI[†,‖], FRANCISCO J. FERRER–TROYANO[*,**] and FRANCISCO RUIZ[†,††]

[*] *Computer Science Department, University of Sevilla,*
*Escuela Superior de Ingeniería Informática,*
*Av. Reina Mercedes S/N, Seville, 41012, Spain*
*http://www.lsi.us.es*
[†] *Computer Science Department, University of Castilla La Mancha,*
*Escuela Superior de Informática,*
*Ronda de Calatrava 5, Ciudad Real, 13071, Spain*
*http://www.inf-cr.uclm.es*
[‡] *riquelme@lsi.us.es*
[§] *mpolo@inf-cr.uclm.es*
[¶] *aguilar@lsi.us.es*
[‖] *mpiattin@inf-cr.uclm.es*
[**] *ferrer@lsi.us.es*
[††] *fruiz@inf-cr.uclm.es*

Received 1 October 2004
Accepted 17 May 2005

This paper presents an empirical study analysing the relationship between a set of metrics for Fourth–Generation Languages (4GL) programs and their maintainability. An analysis has been made using historical data of several industrial projects and three different approaches: the first one relates metrics and maintainability based on techniques of descriptive statistics, and the other two are based on Data Mining techniques. A discussion on the results obtained with the three techniques is also presented, as well as a set of equations and rules for predicting the maintenance effort in this kind of programs. Finally, we have done experiments about the prediction accuracy of these methods by using new unseen data, which were not used to build the knowledge model. The results were satisfactory as the application of each technique separately provides useful perspective for the manager in order to get a complementary insight from data.

*Keywords*: 4GL; data mining; metrics; maintenance prediction.

## 1. Introduction

Most information systems in organizations use some type of database for saving data although in the last few years, new models and paradigms for databases have emerged (for example, object–oriented or object–relational). Recent studies point

out that the relational model is still the most commonly used in organizations [10]. To allow users to manipulate persistent data saved in the database, programmers provide them with programs that embed SQL instructions for accessing the database. These programs may have been developed in a 3GL (such as Cobol or Visual Basic), directly in a 4GL, or have migrated from a 3rd to a 4th Generation Language, which are much more productive environments [9].

4GL are non-procedural languages whose programs specify what the program must do without giving the details of "how to do it". They appeared as a powerful mechanism to easily exploit the characteristics of the Database Management Systems. Examples of 4GL are Oracle Forms, Powerbuilder or CA–OpenIngres.

This paper analyzes the relationship between a set of metrics for 4GL programs and the maintenance time for this sort of programs. The metrics presented have been formally and empirically validated:

- The formal validation can be found in [11] and was made taking into account the formal frameworks proposed by [5] and [19].
- A first empirical study for validating the metrics was presented in [12]. This study used classical Descriptive Statistics to find linear correlations between the metric values and the time devoted to maintenance.

In both studies, meaningful correlations between both variables were found. From the empirical study, a set of useful linear equations was also proposed, in order to predict the maintenance time of 4GL programs.

In this paper we put our focus on a re-analysis of the obtained results, using two Data Mining techniques: M5', developed in [15], and HIDER, an evolutionary algorithm for generating decision rules [2]. Decision rules cannot only be used to predict, but also to provide additional insight from data. Our goal is two-fold: firstly, to present a comparative analysis about statistic techniques and data mining methods; secondly, to build different approaches to predict the maintenance time of 4GL programs from the metrics considered in this work.

This paper is organized as follows: Sec. 2 describes the proposed and collected metrics, together with a brief discussion on the special characteristics of 4GL programs. In Sec. 3 we present a table containing the historical data collected and explaining some characteristics of the collection method. A summary of the analysis of this data with Descriptive Statistics is presented in Sec. 4 as well as the linear equations obtained; Sec. 5 is devoted to the analysis of the same data with Data Mining, as well as to present the decision rules. In Sec. 6, results are discussed and, finally, we present our conclusions in Sec. 7.

## 2. Brief Description of the Proposed Metrics

The definition of metrics for 4GL presents the problem of the great heterogeneity of possible types of statements that may compose these types of programs. In this manner, at least seven sub-languages can be identified:

- Procedural-control sub-language
- Visual-control sub-language
- Exception-handling sub-language
- Database-definition sub-language
- Database-manipulation sub-language
- Security-control sub-language
- Transaction-control sub-language

Metrics presented in this work are designed for the database-manipulation sub-language. Other authors have proposed metrics for measuring different attributes of 4GL programs, although most of them have focused on the estimation of the development effort and on its correlation with the size of programs [3, 8, 17].

The database–manipulation sub-language is mainly composed of SQL instructions. Our metrics take into account the total number of Select, Insert, Delete and Update instructions, the total number of Nestings, as well as the total number of tables and the total number of Where clauses in the program. So, the metrics defined and formally validated (see [11]) are:

- NS = total number of Select instructions in the considered program
- NI = total number of Insert instructions in the considered program
- ND = total number of Delete instructions in the considered program
- NU = total number of Update instructions in the considered program
- NT = total number of used Tables in the considered program
- NN = total number of Nestings in the considered program
- Where = total number of Where clauses in the considered program

The dependent variable is the time of maintenance (TIME) of the program. In this study we consider that the "time of maintenance" is the time devoted to maintenance tasks from the moment when the application was installed until the end of its second version. Time is measured in minutes with an application done specifically to manage maintenance: a button is pressed at the beginning and end of each operation. The time that appears in the tables is the total time of the maintenance (it can have one or several operations).

## 3. Historical Data Collected

The system where the data is collected is an application developed and maintained by the Provincial Center of Informatics (CENPRI), belonging to the Provincial Council of Ciudad Real (Spain). The CENPRI develops and maintains software for its own management, for the management of the 100 municipalities of the province and for managing the relationships with citizens. This includes tax collection, property registers, public infrastructures, permissions for private work, salaries and personnel, etc. These programs interface with several applications from banks, the Ministry of Finance and the Regional Government.

The quality of the software developed by CENPRI was recognized at the national level, being the recipient of an award by the Ministry of Industry and Energy two years ago. A recent evaluation [14] of the maturity level of its Maintenance Process according to the IT Service Capability Maturity Model [13] placed this organization at the second CMM level (Repeatable). Although the reader may feel that this is not an adequate level for an organization like this, the reality is that Public Entities probably have a set of special characteristics that may make the reaching of higher levels unnecessary. This idea is of course quite discussable, and so we put some of our reflections in the afore-mentioned reference [14].

The analyzed system consists of 143 programs written in CA–OpenIngres/4GL, and it is mainly composed of sentences of data-manipulation written in SQL. The programs have three general functions: (1) to manage the relationships of citizens with the Local Public Administration (provincial and municipal councils), for instance, taxes, work permits, etc; (2) to manage the internal operation of the administration (payroll) and (3) manage the communication of these Public Entities with banks. The system was completely developed by the same team that maintained it. These factors may be considered as a constant and, therefore, the reliability of this study can be high [4].

The data of the 143 programs can be represented in this way:

Table 1. Historical data collected.

| NAME | NS | NI | ND | NU | NN | NT | WHERE | TIME |
|---|---|---|---|---|---|---|---|---|
| timer_on.osq | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5 |
| compr_li.osq | 1 | 0 | 0 | 0 | 2 | 5 | 27 | 65 |
| consulta.osq | 12 | 0 | 0 | 0 | 3 | 10 | 65 | 130 |
| cont000.osq | 3 | 0 | 0 | 0 | 1 | 3 | 5 | 30 |
| cont100.osq | 1 | 0 | 0 | 0 | 1 | 3 | 5 | 30 |
| cont101.osq | 6 | 3 | 1 | 2 | 2 | 8 | 39 | 95 |
| cont102.osq | 4 | 0 | 0 | 1 | 2 | 6 | 14 | 75 |
| cont103.osq | 7 | 1 | 0 | 1 | 4 | 12 | 73 | 160 |
| cont104.osq | 1 | 0 | 0 | 0 | 1 | 3 | 5 | 30 |
| cont105.osq | 2 | 1 | 0 | 0 | 0 | 4 | 5 | 30 |
| cont110.osq | 0 | 0 | 0 | 0 | 0 | 1 | 5 | 5 |
| cont120.osq | 2 | 1 | 0 | 0 | 3 | 4 | 30 | 65 |
| cont121.osq | 7 | 0 | 0 | 1 | 4 | 7 | 67 | 105 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

## 4. Analysis of the Historical Data Using Descriptive Statistics

A linear multivariable regression model was initially applied to the historical data. We found that only three of the seven metrics collected had a meaningful correlation with the dependent variable, TIME. In [12] a detailed description of the experiment can be found including several coefficients and further discussion. The

linear equation for predicting maintenance time is:

$$TIME = 10.01 \times NN \ + \ 10.22 \times NT \ + \ 0.08 \times WHERE \ - \ 9.84 \qquad (1)$$

As can be seen, only NN, NT and WHERE are significantly correlated to TIME. The correlation coefficients are:

Table 2. Correlation of the metric values with TIME.

| Metric | Correlation with TIME |
|---|---|
| NS | 0.29 |
| NI | 0.39 |
| ND | 0.13 |
| UN | 0.08 |
| NN | 0.88 |
| NT | 0.99 |
| WHERE | 0.96 |

## 5. Analysis of the Same Historical Data with Data-Mining Techniques

The field of data mining, like statistics, concerns itself with "learning from data" or "turning data into information". The context encompasses statistics, but with a somewhat different emphasis. Data miners are often more interested in understandability than accuracy or predictability *per se*. Thus, there is a focus on relatively simple interpretable models involving decision rules, decision trees, association rules, and so forth. Statistical methods can provide valid information, although in some cases, large amounts of data require another treatment to discover understandable knowledge and reduce the computational costs.

If we obtain a globally weighted linear regression, we can confirm that only the last three parameters have influence over the variable TIME (see Eq. (1)). The average mean error of this regression summed over the set of examples is 3.8622 (6.28% relative mean error) by using a 10-fold cross-validation.

We also tackle the problem with a regression-tree-based method called M5, a prediction-oriented technique, as well as with HIDER, which finds a set of decision rules by means of an evolutionary algorithm. Decisions rules add value to the knowledge obtained by prediction techniques, since the discretized variable TIME is related to intervals or values of the metrics.

### 5.1. *Data analysis with M5*

M5 is a well-known tool for data mining researchers [15], which builds trees whose leaves are composed by a multivariate linear model, analogous to piecewise linear functions. M5 chooses the test (nodes no leaves) over the attribute that maximizes

```
–NT <= 5.5 :
|   NN <= 1.5 :
|   |   |   NT <= 1.5 : LM1
|   |   NT > 1.5 :
|   |   |   NT <= 3.5 : LM2
|   |   |   NT > 3.5 :
|   |   |   |   WHERE <= 8.5 : LM3
|   |   |   |   WHERE > 8.5 : LM4
|   NN> 1.5 : LM5
+–NT> 5.5 : LM6
```

with:
LM1: TIME = 5.48
LM2: TIME = -10.2 + 10.1NN + 9.88NT
LM3: TIME = 31.4
LM4: TIME = 42.1
LM5: TIME = -15.2 + 13.8NN + 9.64NT
LM6: TIME = -11.3 + 10.3NN + 11NT - 0.887NI

Fig. 1. Results provided by M5'.

the expected error reduction as a function of the standard deviation of the variable or output parameter.

We have applied a version of M5 called M5' to the historical data. M5' is implemented in the WEKA library [18]. The application of M5' over the data (Fig. 1) shows a regression tree that decreases the error rate to 2.8 (4.5% in relative terms), therefore giving a more accurate prediction than linear regression.

Six categories of the dependent variable (TIME) are found by M5' (LM1 to LM6). TIME in some of these categories receives a constant value, and in others must be calculated with a linear equation. The interpretation of Fig. 1 is the following:

- If NT is less or equal to 1.5 and NN is less or equal to 1.5 then TIME = 5.48.
- If NT is greater than 1.5 and less or equal than 3.5 and NN is less or equal to 1.5 then TIME $= -10.2 + 10.1 \times NN + 9.88 \times NT$.
- If NT is greater than 3.5 and less or equal than 5.5 and NN is less or equal to 1.5 and WHERE is less or equal than 8.5 then TIME = 31.4, and so on ...

As additional information, the collected data was categorized by M5 as follows:

| | |
|---|---|
| LM1: 21 programs | LM2: 34 programs |
| LM3: 7 programs | LM4: 7 programs |
| LM5: 27 programs | LM6: 47 programs |

### 5.2. *Data analysis with HIDER*

HIDER [2] is a system based on Evolutionary Algorithms (EA) that provides a decision rule set from a labelled database. A decision rule can be a condition as follows:

$$\textit{If } p_1 \in [2.1, 3.7] \;\; \textit{and } p_2 \in [6.3, 9.8] \;\; \textit{and } p_3 = 4 \textit{ then variable is } A$$

where $p_i$ are independent parameters (metrics in this case) and *variable* is a dependent variable (TIME) and $A$ is an interval of values for the last one.

The EA provides us with a method for finding good solutions in a complex search space where parameters do not have an obvious relationship. The application of EAs to a learning problem requires the selection of an internal representation of the space to be searched and the definition of an external function that assigns fitness to candidate solutions. Both components are critical for the successful application of the EAs to the problem of interest. The algorithm chooses the best individual of the evolutionary process, transforming it into a rule, which is used to eliminate covered data from the training file. In this way, the training file is reduced for the following iteration. A termination criterion can be reached when more examples to cover do not exist.

#### 5.2.1. *Discretizing the dependent variable*

To apply HIDER, the dependent variable (TIME) must be made discrete into a set of disjoint categories, for which a number of techniques can be used, such as:

- The k Nearest Neighbour (k–NN) technique, for example, finds the three categories shown in Fig. 2, assuring that the last three independent variables (NN, NT and WHERE) have influence on the metric distance [7]. The accuracy results were similar to that of linear regression, although the error rate was slightly greater. To minimize the absolute error, linear regression tends to better adjust to the large values of the variable TIME, losing accuracy for small values (above all, 5 and 10). In fact, k–NN showed the opposite behaviour, being more precise with smaller values. This behaviour is due to the presence of many small values with high frequency and the difficulty of estimation when large values have frequency equal to one. k–NN has discretized TIME based on the values inclined to make errors.
- A manual discretization was made due to the aforementioned behaviour and to the desire of comparing HIDER with M5'. In the same way that M5' finds six

| | |
|---|---|
| **Label A**, when **TIME <= 40** | (74 records) |
| **Label B**, when **40 <TIME <= 105** | (33 records) |
| **Label C**, when **TIME >105** | (36 records) |

Fig. 2. Discretizing the dependent variable with the k–NN technique.

| | |
|---|---|
| **Label A**, when **TIME <= 10** | (29 records) |
| **Label B**, when **10 <TIME <= 25** | (20 records) |
| **Label C**, when **25 <TIME <= 40** | (25 records) |
| **Label D,** when **40 <TIME <=75** | (23 records) |
| **Label E,** when **75 <TIME <=150** | (21 records) |
| **Label F,** when **TIME >150** | (25 records) |

Fig. 3. Manual discretization of the dependent variable.

different regressions, we have made the variable TIME discrete in six intervals (the same number found by M5', see Fig. 3) with approximately the same number of records. These intervals are in Fig. 3.

### 5.2.2. *Applying HIDER*

Once the output parameter has been discretized (according to Fig. 3) and assigned a class label to each interval for the output parameter, we applied HIDER for extracting rules task. HIDER's output is a set of decision rules which indicates what intervals for the input parameters (metric values) are able to predict the class (discretized output variable TIME). HIDER has a user–defined parameter to adjust the expected error rate during the learning process. When we are interested in producing few rules, a greater error rate is allowed; if we need more accuracy, this parameter will be equal to zero and the number of rules will therefore be higher.

HIDER is able to provide two sorts of results: a single and hierarchical rule set that allows the classification of all class labels, or a set of rules for each class label independently. The first case is useful when the purpose is to classify new unseen data[a] and therefore the least number of rules is preferred. The hierarchical structure of rules implies a lesser number of rules although with a loss of understandability (Fig. 4). The second case provides a comprehensive knowledge about the influence of the input parameters on the discretized variable for each class label independently. In this case the rules are nor hierarchic because each label has a rule set (see Fig. 5).

In [1] HIDER was compared to the well-known tool for supervised learning C4.5 [16], as a way to show the quality of the results. For this study, HIDER had a better performance than C4.5 since the number of rules was smaller and more records were covered.

The aim of our analysis is to achieve a better knowledge about the relationship among the metrics used and the TIME value. For this reason, a rule set was found for each one out of the six labels independently. Thus, HIDER was run six times, for which the initial evolutionary population only contained individuals of a unique

[a]Unseen data are the examples used in the test phase, in contrast with the examples used during the training phase, which have been considered to build the rule set.

class or label. Taking a population size of 100 and 100 generations, the computational cost is very low (less than a minute on a Pentium II 450 MHz). Table 3 shows the rules obtained after applying HIDER to the data.

If conditions(1) the TIME is Label A
Else If conditions(2) then TIME is Label D
Else If conditions(3) then TIME is Label B
Else If conditions(4) then TIME is Label D

Fig. 4. Hierarchical set of rules for every label.

Hierarchical set
If conditions(1) then TIME is Label A
Else If conditions(2) then TIME is Label A
Else If conditions(3) then TIME is Label A
Disjunctive rule
If conditions(1) or
conditions(2) or
conditions(3) then TIME is Label A

Fig. 5. Hierarchical set for a single label is equivalent to one disjunctive rule.

Table 3. Results provided by HIDER.

| NS | NI | ND | NU | NN | NT | WHERE | Label | Records/errors |
|---|---|---|---|---|---|---|---|---|
| | | | | =0 | >=2 | | A | 25/0 |
| | | | | =0 | =3 | | B | 13/0 |
| | >=1 | | | =0 | >=4 | | C | 6/1 |
| | | | | >=1 | >=1 | | A | 3/0 |
| | | | | >=1 | >=5 | ∈ [12,29] | D | 3/0 |
| | | | | =1 | =2 | | B | 6/0 |
| | | | | =1 | >=3 | >=18 | C | 13/1 |
| | >=1 | | | =1 | >=3 | | C | 11/1 |
| | | | | ∈ [2,3] | >=5 | >=17 | D | 17/1 |
| | =0 | | | >=2 | >=4 | >=18 | C | 5/1 |
| | | >=2 | | >=2 | >=6 | ∈ [17,39] | D | 18/1 |
| | | | | =2 | >=3 | >=10 | C | 5/1 |
| | | >=2 | | >=2 | ∈ [4,6] | >=39 | D | 18/1 |
| | | | | =3 | >=3 | | D | 2/0 |
| | | | | | ∈ [4,5] | >=15 | C | 8/1 |
| | | | | | ∈ [6,11] | >=35 | E | 19/1 |
| >=7 | | | | | >=12 | | F | 24/0 |

Table 4. Results of all the techniques.

| NAME | TIME | LR | M5' | HIDER Category | Right |
|---|---|---|---|---|---|
| borra_u.osq | 75 | 73.53 | 74.64 | ∈ (40, 75] | 1 |
| compr_li.osq | 65 | 63.44 | 60.76 | ∈ (40, 75] | 1 |
| consulta.osq | 130 | 127.59 | 129.6 | ∈ (75, 150] | 1 |
| cont000.osq | 30 | 31.23 | 29.54 | ∈ (25, 40] | 1 |
| cont100.osq | 30 | 31.23 | 29.54 | ∈ (25, 40] | 1 |
| cont101.osq | 95 | 95.06 | 94.64 | ∈ (75, 150] | 1 |
| cont102.osq | 75 | 72.62 | 75.3 | ∈ (40, 75] | 1 |
| cont103.osq | 160 | 158.68 | 161.01 | >150 | 1 |
| cont104.osq | 30 | 31.23 | 29.54 | ∈ (25, 40] | 1 |
| cont105.osq | 30 | 31.44 | 31.4 | ∈ (25, 40] | 1 |
| cont110.osq | 5 | 0.78 | 5.48 | <=10 | 1 |
| cont120.osq | 65 | 63.47 | 65 | ∈ (40, 75] | 1 |
| cont121.osq | 105 | 107.1 | 106.9 | ∈ (75, 150] | 1 |
| ... | ... | ... | ... | ... | ... |

## 6. Discussion of the Results

Table 4 shows some of the results of applying the three techniques to the sample:

- TIME is the observed value for the dependent variable.
- LR is the estimated time using Linear Regression.
- M5' is the estimate time using M5'.
- For HIDER, we show the estimated category and whether or not it is correct. HIDER's predictions are good for 95.10% of cases.

The three estimation methods are very different, but comparisons among them are possible. It is possible to make an objective comparison between M5' and LR using statistical estimates. The "crisp nature" of HIDER makes it hard to compare with the other techniques, although this point will be discussed in depth later.

### 6.1. *LR and M5'*

We compare the goodness of these prediction techniques using the Mean Magnitude of Relative Error (MMRE) and PRED(q), proposed by [6]:

- MMRE is given by Eq. (2). In a sample of size $n$, $\hat{e}_i$ is the estimated value for the $i$th element, and $e_i$ is the actual value. Conte *et al.* suggest then an acceptable value for MMRE is a value less or equal to 0.25.

$$MMRE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| \frac{e_i - \hat{e}_i}{e_i} \right| \tag{2}$$

- PRED(q), being $q$ a percentage, is the number of cases whose estimations are under $q$, divided into the total number of cases. For example, if PRED(0.1)=0.9,

it means that 90% of cases have estimations inside 10% of its actual value. For Conte, an estimation technique is acceptable if PRED(0.25)>=0.75. PRED(q) is calculated using:

$$PRED(q) = \frac{k}{n} \tag{3}$$

where:

— $k$ is the number of elements in the sample whose MMRE is less than or equal to $q$

— $n$ is the number of elements in the sample

With these estimates, the behaviour of both prediction techniques is good. In fact see Table 5:

Table 5. Estimates for LR and M5'.

| | MMRE | PRED (0.25) |
|---|---|---|
| LR | 19.46% | 81.81% |
| M5' | 4.5 % | 95.80% |

It is clear that the goodness of M5' is greater than LR. The origin of the high value of MMRE in LR is in the low values of TIME, since a time less than 1 unit is always predicted in these cases, even though the mean actual time is 5 units.

Moreover, M5' discovers some factors that would remain hidden using only linear regression, such as the relative importance of some of the metrics depending on the characteristics of the program: both techniques only highlight NT, NN and WHERE as representative metrics for effort estimation, but actually NN and WHERE only have influence when the Number of Tables is low (5 or less). Moreover, when the Number of Nestings is 2 or more, then the value of WHERE has no influence. These rules may help programmers to write more maintainable software.

### 6.2. *LR, M5' and HIDER*

Depending on the importance of the accuracy of the estimation (the "quantitative point of view"), it is obvious that maybe to give an interval of TIME as an effort prediction may not be enough. This issue, of course, depends on the actual organization and situation.

Speaking in qualitative terms, the high reliability of the HIDER technique stays in the added-value information provided. In fact, the rules found by HIDER can be textually expressed as follows:

- TIME takes values smaller or equal than 10 if NN and NT take very low values, concretely if both parameters do not add more than 2.
- If the sum of NN and NT is three, then TIME will be between 15 and 25.

- The variable TIME will take values between 30 and 40 in three possible cases depending on NN:
  - — If NN equals 0 and NT is greater than 3 and WHERE is less than 16.
  - — If NN equals 1 and NT takes values between 3 and 5.
  - — If NN equals 2 and NT is smaller than 3 and WHERE is greater than 9.
- The values of TIME are between 45 and 75 in three scenarios:
  - — When NN equals 2 or 3, and NT is smaller or equal than 5, and WHERE is greater than 16.
  - — If ND is smaller or equal than 2, NN is greater than 1, NT is bounded by 6 and WHERE is between 17 and 39.
  - — Also, if NN is greater or equal than 1, NT is greater than 4 and WHERE is between 12 and 29.
- If NT takes values between 6 and 11 and WHERE is greater than 34, then TIME will be between 80 and 150.
- Finally, TIME will be greater than 150 if NS and NT are greater than 6 and 12, respectively.

In summary, NT is the most decisive parameter for TIME: if NT equals 0, 1 or 2, TIME will take values less than 25, also depending on the value of NN. If NT takes values between 3 and 6, then TIME will be between 30 and 75, depending on the values of NN and WHERE. If NT is between 6 and 11 then TIME will be in the [80, 150] range and TIME is greater than 150 when NT takes values greater than 12.

### 6.3. *Prediction capability*

To prove the prediction capability of the analyzed techniques, new unseen data about the maintenance of programs was selected. This data (33 in total) was about programs different from the 143 used to build the model.

Regarding the linear regression, MMRE was 0.3 and for M5, 0.17. The value of PRED (0.25) for LR was 0.7 and for M5, 0.73. The error MMRE was greater than 0.25 ten times for LR and nine times for M5. Six programs out of 33 corresponded to the case TIME = 10, which can be predicted by LR with a value of 0.46 (MMRE = 0.95) and for M5 with a value of 5.48, according to the equation LM1 in Fig. 1 (MMRE = 0.45). This shows the difficulty of predicting the values of the variable TIME. For instance, if the prediction of the regression LM1 had been 7.5 (a value between 5 and 10, the smallest possible values), then MMRE from M5 would be 0.15 and PRED (0.25) = 0.85. These results are excellent due to the characteristics of data.

Regarding HIDER, 28 out of 33 programs were correctly classified, i.e. 84.8%. Analyzing the 5 errors, we can state that although they were considered as errors, the prediction values were very close to the real values. Table 6 shows these errors.

Table 6. Forecasting errors with HIDER.

| Program name | Value of TIME | Interval predicted |
|---|---|---|
| ppla101.osq | 15 | [5,10] |
| ppla103.osq | 10 | (10,25] |
| ppla203.osq | 80 | (40,75] |
| ppla241.osq | 80 | (40,75] |
| ppla300.osq | 50 | (25,40] |

## 7. Conclusions

This paper has presented the results of applying three different techniques (linear regression, M5 and HIDER) for predicting maintenance effort in 4GL programs.

From a quantitative point of view, the M5 technique, based on Data-Mining, has proved to be much better than Linear Regression. When the concrete predicted value for TIME is important, then M5' should be used.

Moreover, as was discussed, rules provided by M5' and HIDER provide further insight into the program characteristics. The values and intervals provided for TIME, together with the metric values, may help programmers and project managers to set thresholds for writing programs with higher maintainability levels.

Estimating the maintenance time of new unseen programs has proved the prediction accuracy of data mining-based tools. The potential of these techniques offers good insight from data to managers of software projects, as the approaches reached very low errors.

Although our results cannot be generalized to all situations, techniques aside from classical Statistics should be explored for building prediction models: not only for a more accurate estimation, but also for the additional knowledge and information which they give.

Other practitioners or researchers can download a spreadsheet for doing this kind of predictions from http://www.inf-cr.uclm.es/www/mpolo/varios/4gl.xls.

## References

1. J. S. Aguilar–Ruiz, I. Ramos, J. C. Riquelme, and M. Toro, An evolutionary approach to estimating software development projects, *Information and Software Technology* **43**(14) (2001) 875–882.
2. J. S. Aguilar–Ruiz, J. C. Riquelme, and M. Toro, Evolutionary learning of hierarchical decision rules, *IEEE Systems, Man and Cibernetics Part B* **33**(2) (2003) 324–331.
3. P. Bourque and V. Côté, An experiment in software sizing with structured analysis metrics, *Journal of Systems and Software* **15** (1991) 159–172.
4. V. Basili, F. Shull, and F. Lanubile, Building knowledge through families of experiments, *IEEE Trans. on Software Engineering* **25**(4) (1999) 453–473.
5. L. C. Briand, S. Morasca, and V. Basili, Property-based software engineering measurement, *IEEE Trans. on Software Engineering* **22**(1) (1999) 68–85.
6. S. D. Conte, H. E. Dunsmore, and V. Shen, *Software Engineering Metrics and Models*, Benjamin/Cummings, 1986.

7. B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, IEEE Computer Society Press, 1991.
8. J. J. Dolado, A study of the relationships among Albrecht and Mark II function points, lines of code 4GL and effort, *Journal of Systems and Software* **37** (1997) 161–173.
9. S. Holloway, *Fourth–Generation Systems: Their Scope Application and Methods of Evaluation*, Chapman & Hall, London, 1990.
10. N. Leavit, Whatever happened to object-oriented databases?, *IEEE Computer* **33**(8) (2000) 16–19.
11. A. Martínez and M. Piattini, Validation of measures to assess the maintainability of SQL programs, in *Proc. 11th ESCOM-SCOPE Conference*, 2001.
12. A. Martínez and M. Piattini, Measuring for database programs maintainability, in *Proc. 11th Database and Expert Systems Applications Conference — DEXA'01*, LNCS, vol. 1873, 2001, pp. 65–78.
13. F. Niessink and H. van Vliet, The Vrije Universiteit IT Service Capability Maturity Model, Technical Report IR-463, Release L2-1.0, Faculty of Sciences, Division of Math and Comp. Sci., Vrije Universiteit Amsterdam, Amsterdam, 1999.
14. M. Polo, M. Piattini, F. Ruiz and M. Jiménez, Assessment of Maintenance Maturity in IT Departments of Public Entities: Two Case Studies, *Product Focused Software Process Improvement*, LNCS, vol. 2188, 2001, pp. 86–97.
15. J. R. Quinlan, Learning with continuous class, in *Proc. 5th Australian Joint Conference on Artificial Intelligence*, World Scientific, 1992, pp. 343–348.
16. J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
17. J. Verner and G. Tate, Estimating size and effort in fourth-generation development, *IEEE Transactions on Software Engineering* **5**(4) (1988) 15–22.
18. I. Witten and E. Frank, *Data Mining Practical: Machine Learning Tools and Tecniques with Java Implementations*, Morgan Kaufmann, 1999.
19. H. Zuse, *A Framework for Software Measurement*, Walter de Gruyter, Berlin, 1998.