

# A Comparative Study of Two Techniques for Analyzing Software Measurement Data<sup>1</sup>

Sandro Morasca  
Politecnico di Milano  
Dipartimento di Elettronica e Informazione  
Piazza Leonardo da Vinci 32  
I-20133 Milano  
morasca@elet.polimi.it  
<http://www.elet.polimi.it/~morasca>

Günther Ruhe  
Fraunhofer Gesellschaft  
Institute for Experimental Software Engineering  
Sauerwiesen 6  
D-67661 Kaiserslautern  
ruhe@iese.fhg.de

## Abstract

Careful analysis of Software Engineering measurement data is essential in deriving the right conclusions from performed experiments. Different data analysis techniques may provide data analysts with different and complementary insights into the studied phenomena. In this paper, two data analysis techniques – Rough Sets and Logistic Regression – are compared, from both the theoretical and the experimental points of view. In particular, the empirical study was performed as part of the ESPRIT/ESSI project CEMP on a real-life maintenance project, the DATATRIEVE project carried out at Digital Engineering Italy. The two data analysis techniques are different in nature: Logistic Regression uses a statistical approach, while the Rough Sets analysis technique does not. We have applied both techniques to the same data set. The goal of the experimental study was to determine the major factors affecting reliability and reusability in the application context. Results obtained with either analysis technique are discussed and compared, to identify commonalities and differences between the two techniques. Finally, both analysis techniques are evaluated with respect to their weaknesses and strengths.

**Keywords:** Data Analysis, Rough Sets, Logistic Regression, Empirical Studies, Software Maintenance.

## 1. Introduction

Careful data analysis is necessary for deriving the right conclusions from collected experimental data. Also, experimenters should make the most out of the available data sets, which are usually smaller in Software Engineering applications than in other fields. No "silver bullet" technique exists answering all questions for different situations and circumstances in an optimal way. On the contrary, it is reasonable to expect that different techniques will have context-sensitive strengths and weaknesses. Furthermore, different insights will likely be gained from the different assumptions underlying them and from the use of different analysis processes. Therefore, we believe that it is important to study, assess, and compare several different techniques.

In this paper, we will study the application of two techniques – Logistic Regression [HL89] and Rough Sets [Paw91] – to the analysis of measurement data of a real-life maintenance project, the DATATRIEVE™ project carried out at Digital Engineering Italy. The experiment was performed as a part of the CEMP project [CEMP95], an ESPRIT/ESSI project funded by the European Union. The overall project was devoted to the customized establishment of goal-oriented measurement programs which were based on the *Goal/Question/Metric* (GQM) paradigm [BW84],[BR88]. GQM was applied and evaluated by itself in a case study replicated across three major companies. In all participating projects, goals related to reliability and reusability – quality aspects of maintenance processes important for all three companies – were studied, to increase consistency and comparability of results across the CEMP project.

---

<sup>1</sup> Supported by ESPRIT/ESSI project # 10358, Customized Establishment of Measurement Programs (CEMP)

Logistic Regression [HL89] is a classification technique based on maximum likelihood estimation that was originally developed for biomedical applications and has already been successfully used in a few software engineering applications (for instance, see [BMB94]). Based on the values of a given set of explanatory variables for the attributes of an object, Logistic Regression estimates the probability that the object may be classified as belonging to one of the possible categories. Therefore, Logistic Regression is quite different from other regression techniques (e.g., linear regression), which aim at finding out to what extent there exists a relationship of a specified kind (e.g., linear) between explanatory and dependent variables.

Rough Set theory [Paw91] is a new and promising data analysis approach which has been successfully applied in many real-life problems of various areas, e.g. medicine, pharmacology, business, banking. While no additional information such as independence of explanatory variables or assumptions on distribution of data is required, conclusions can be drawn even for small sized data sets. As a result, computation of cause-effect relationships in a minimal knowledge representation is performed.

Logistic Regression and Rough Sets are different in that Logistic Regression is based on statistics, while the Rough Sets approach is not. The objective of the paper is to

- identify major influence factors of adaptive maintainability for the studied environment, which will be independently carried out with either analysis technique,
- compare similarities and differences of results,
- assess both analysis techniques with respect to their context-specific weaknesses and strengths.

The remainder of this paper is organized as follows. In Section 2, the practical problem is described. This covers experimental design and other context information as a prerequisite for later analysis of experimental data. Logistic Regression is described in Section 3, and necessary background from Rough Set theory is given in Section 4. In Section 5.1, we list a set of evaluation criteria for data analysis techniques, to provide the grounds for a theoretical assessment and comparison of both approaches. In Sections 5.2 and 5.3, we describe the analysis results we have obtained by applying Logistic Regression and Rough Sets, respectively, to the maintenance data resulting from the CEMP project. The results are used to compare both approaches and get insights into their weaknesses and strengths, as we show in Section 5.4. Summary and concluding remarks follow in Section 6.

## **2. The Experimental Environment**

Our experiment concerned the transition of the DATATRIEVE™ product from version 6.0 to version 6.1. The following two activities were being carried out during that transition.

- Corrective maintenance: failures reported from customers were being fixed.
- Adaptive maintenance: DATATRIEVE™ was being transferred from platform OpenVMS/VAX (version 6.0) to platform OpenVMS/Alpha (version 6.1).

At the time of the experiment, the DATATRIEVE™ team was composed of six people. The DATATRIEVE™ product was originally developed in the BLISS language. Recently, some parts have been added or rewritten in the C language. Therefore, the overall structure of DATATRIEVE™ is composed of C functions and BLISS subroutines. The empirical study of this paper reports on the BLISS part, by far the bigger one. In what follows, we will use the term "module" to refer to a BLISS module, i.e., a set of declarations and subroutines usually belonging to one file. More than 100 BLISS modules have been studied.

We followed the measurement process used for all application experiments within the CEMP project. This measurement process is based on the Quality Improvement Paradigm (QIP) [BCR94], an evolutionary paradigm that supports quality improvement by means of reuse of experience. More details, along with guidelines for the application of the process, may be found at [CEMPwww] and in [CEMP95]. In particular, the *Goal/Question/Metric* (GQM) paradigm [BW84] [BR88] was used as a part of this QIP-based process to iteratively define metrics for those attributes which are important in the context of the measurement goals. This is the GQM goal we defined:

**Analyze** version 6.1 of DATATRIEVE™ *(object of study)*  
**for** **the** **purpose** **of** understanding  
*(purpose)*  
**with respect to** the impact of modifications from version 6.0 to version 6.1 on reliability *(quality focus)*  
*(quality focus)*  
**from** **the** **viewpoint** **of** the project leader  
*(viewpoint)*  
**in** **the** **following** **environment:** Digital Italy – Gallarate  
*(environment)*

The five dimensions of measurement goals help precisely identify what is investigated (*object of study*), why (*purpose*), with respect to which specific attribute (*quality focus*), for whose benefit (*viewpoint*), and in what context (*environment*). GQM goals are later refined into questions and metrics, to capture all relevant factors<sup>2</sup> in a quantitative fashion.

Here, we will list the relevant explanatory variables that will also be used in the experimental comparison between Rough Sets and Logistic Regression. Other explanatory variables were also investigated, but our experimental analysis showed that they were not relevant with respect to the quality focus of interest.

- **LOC6.0:** number of lines of code of version 6.0.
- **LOC6.1:** number of lines of code of version 6.1.
- **AddedLOC:** number of lines of code that were added in version 6.1, i.e., they were not present in version 6.0.
- **DeletedLOC:** number of lines of code that were deleted from version 6.0, i.e., they were no longer present in version 6.1.
- **SizeDifferenceRate:** relative difference in size between versions, i.e.,  $(LOC6.1 - LOC6.0) / LOC6.0$ .
- **DifferentBlocks:** number of different blocks between versions 6.0 and 6.1.
- **ExpansionRate:** rate of expansion from versions 6.0 to version 6.1, i.e.,  $AddedLOC / (DeletedLOC + AddedLOC)$
- **ReuseRate:** percentage of lines of code of version 6.0 reused in version 6.1, computed as  $(LOC\ 6.0 - DeletedLOC) / LOC\ 6.0$ .
- **ModificationRate:** rate of modification, i.e.,  $(AddedLOC + DeletedLOC) / (LOC6.0 + AddedLOC)$ .
- **ModuleKnowledge:** subjective variable that expresses the project team's knowledge on modules (low or high).

### 3. A Concise Introduction to Logistic Regression

---

<sup>2</sup>An intermediate document, the abstraction sheet, was used to refine the GQM goal, and as a basis to derive the questions. Details about this augmentation to the GQM paradigm can be found in [HOR96].

Here, we provide a concise description of Logistic Regression. For a more comprehensive introduction, the reader may refer to [HL89]. Logistic Regression is a classification technique whose aim is to estimate the probability that an object belongs to each specific class, based on the values of the explanatory variables. As such, Logistic Regression is different from other regression techniques (e.g., linear regression), whose goal is to determine whether there is some form of functional dependency (e.g., linear) between explanatory variables and dependent variable. Logistic regression does not assume any strict functional form to link explanatory variables and the probability function. Instead, this functional correspondence has a flexible shape, that can adjust itself to several different cases. Logistic regression is based on maximum likelihood and assume that all observations are independent.

Here, we will address the case of a dependent variable  $Y$  which can only take two values 0 and 1 and any number of explanatory variables  $X_i$ . The multivariate Logistic Regression model is defined by the following equation (if it contains only one independent variable, then we have a univariate Logistic Regression model):

$$\pi(X_1, X_2, \dots, X_n) = \left[ e^{(C_0 + C_1 X_1 + C_2 X_2 + \dots + C_n X_n) Y} \right] / \left[ 1 + e^{(C_0 + C_1 X_1 + C_2 X_2 + \dots + C_n X_n)} \right]$$

where  $\pi(X_1, X_2, \dots, X_n)$  is the probability that  $Y = 0$  (therefore  $1 - \pi(X_1, X_2, \dots, X_n)$  is the probability that  $Y = 1$ ). In our case study, Faulty6.1 will be the dependent variable, i.e., Faulty6.1 = 0 for a module with no faults and Faulty6.1 = 1 for a module with at least one fault. The measures we have collected on the DATATRIEVE™ modules (see the list at the end of Section 2) will be the explanatory variables. The curve describing the relationship between  $\pi$  and any single  $X_i$ —i.e., under the assumption that all other  $X_j$ 's are constant—has a flexible S shape which ranges between the following two extreme cases.

- (1) A horizontal line, when variable  $X_i$  is not significant (probability  $\pi$  is a constant with respect to  $X_i$ ).
- (2) A vertical line, when variable  $X_i$  alone is able to differentiate between the case  $Y = 0$  and the case  $Y = 1$ . In other words, based on the value of  $X_i$  alone, one can perfectly predict whether  $Y = 0$  or  $Y = 1$ . A vertical line is the limiting case of a steep slope, which shows that small variations of  $X_i$  have a relevant impact on the probability  $\pi$ .

We will use the following three types of statistics to describe the experimental results.

$C_i$ 's, *the regression coefficients*, estimated via the optimization of a likelihood function. The likelihood function is built in the usual fashion, i.e., as the product of the probabilities of the single observations, which are functions of the explanatory and dependent variables (whose values are known in the observations) and the coefficients (which are the unknowns).

$p$ , *the statistical significance of the logistic regression coefficients*, which provides an insight into the accuracy of the coefficient estimates. The level of significance of the logistic regression coefficients provides the probability that the coefficient is different from zero by chance. Historically, a significance threshold ( $\alpha$ ) of  $\alpha = 0.05$  (i.e., 5% probability) has often been used in univariate analysis to determine whether a variable is a significant predictor. The larger the level of significance, the larger the standard deviation of the estimated coefficients, the less believable the calculated impact of the coefficient. The significance test is based on a likelihood ratio test [HL89], commonly used in the framework of logistic regression.

$R^2$ , *the goodness of fit*, not to be confused with least-square regression  $R^2$ —they are built upon very different formulae, even though they both range between 0 and 1 and are similar from an intuitive perspective. The higher  $R^2$ , the higher the effect of the model's ex-

planatory variables, the more accurate the model. However, as opposed to the  $R^2$  of least-square regression, high  $R^2$ s are rare for logistic regression.  $R^2$  may be described as a measure of the *proportion of total uncertainty* that is attributed to the model fit. (The interested reader may refer to [HL89] for a detailed discussion of this issue.)

#### 4. Basic Concepts of Rough Set Theory

To make the paper self-contained, we here provide a concise introduction to the theory of Rough Sets. The interested reader may refer to [Paw91], [PGSZ95] for a comprehensive introduction to the subject and [Ruh96] for its application to goal oriented measurement in Software Engineering. Rough Set theory assumes that attributes of objects (i.e., BLISS modules in our application case) are measured on an ordinal or nominal scale. For Software Engineering measurement, this is often met quite naturally (e.g., degree of experience of project team, degree of communication, programming language). However, in all remaining cases it is necessary to subdivide each attribute's domain into subclasses (nominal scale) or subintervals (ordinal scale). In what follows, we will assume that discretization has been carried out. It is understood that the discretization used may affect the experimental results. The influence of different discretization techniques on quality of results has been investigated in [CG94]. We will assume that the set of attributes is subdivided into a set of explanatory variables and one dependent variable  $Y$  (in our case, the number of faults in version 6.1).

Indiscernability is the main concept of Rough Set theory. Two objects (i.e., two modules in our application) are said to be *indiscernible* with respect to a subset  $P$  of explanatory variables if both objects have the same values for the same explanatory variables in  $P$ . Indiscernability is an equivalence relation between objects, whose equivalence classes are sets of indiscernible objects. Any finite union of such equivalence classes is called a  *$P$ -definable set*, i.e., a set that can be defined based on  $P$ 's explanatory variables.

We will illustrate the Rough Set approach by an example. In Figure 1, let the outer oval represent a set of modules. Suppose that the area within the inner oval represents the set, which we will denote as  $S_0$ , of those modules that do not contain faults and that the area outside the inner oval represents the set, which we will denote as  $S_1$ , of those modules that do contain faults. Suppose also that each of the subsets of modules identified by means of the grid of vertical and horizontal lines (these subsets are labelled by the letters A, B, ..., Z) represents an equivalence class of the indiscernibility relation defined by  $P$ . For instance, all of the modules in the area labelled by A have the same values for the explanatory variables of  $P$ . Any union of the letter-labeled equivalence classes is a  *$P$ -definable set*.

Two  $P$ -indiscernible objects may be associated with two different values of the dependent variable. This situation is called an inconsistency. In our case, such a situation arises when a non-faulty module and a faulty module have the same values for  $P$ 's explanatory variables. For instance, in Figure 1, consider the modules in the class labelled by F, which have the same values for the explanatory variables. Some of the modules in class F are nonfaulty, while the others are faulty. This is true for all the modules in the equivalence classes over the boundary of the inner oval, i.e., those subsets labelled by F, G, H, I, K, N, O, R, S, T, U, V. For the sake of classification, one would like to have a small subset of explanatory variables able to precisely identify the minimal subsets  $S_{Y_i}$  of objects, each of which contains only objects with the same value  $Y_i$  for the dependent variable. In our application case, we would like to use a subset of explanatory variables to classify modules as nonfaulty (i.e., in set  $S_0$ ) or faulty (i.e., in set  $S_1$ ).

{ EMBED Word.Picture.6 }

Figure 1. Rough Sets:  $P$ -definable sets.

The Rough Sets approach handles inconsistencies by providing lower and upper approximations of the right subsets. Given a set  $P$  of explanatory variables and a subset of objects  $S_{Y_i}$ , the greatest  $P$ -definable set that is contained in  $S_{Y_i}$  is called a  $P$ -lower approximation of  $S_{Y_i}$ . Therefore, all of the objects belonging to the  $P$ -lower approximation of  $S_{Y_i}$  also belong to  $S_{Y_i}$  with certainty. Likewise, the smallest  $P$ -definable set that contains  $S_{Y_i}$  is called a  $P$ -upper approximation of  $S_{Y_i}$ . All of the objects of  $S_{Y_i}$  also belong to its  $P$ -upper approximation. Therefore, the  $P$ -upper approximation of  $S_{Y_i}$  contains all those objects that may be classified as belonging to  $S_{Y_i}$ , based on the set  $P$  of explanatory variables. The difference set ( $P$ -upper approximation -  $P$ -lower approximation) is called a boundary region. Elements of the boundary region cannot be classified as members of the set  $S_{Y_i}$  with certainty by using only the given set of attributes. For instance, in Figure 1, the  $P$ -lower approximation of  $S_0$  is given by the union of the four subsets labelled by L, M, P, Q; the  $P$ -upper approximation of  $S_0$  is given by the union of the subsets labelled by F, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V. The boundary region of  $S_0$  is therefore the union of those subsets across the boundary of the inner oval, i.e., those labelled by F, G, H, I, K, N, O, R, S, T, U, V. In the same way, the  $P$ -lower approximation of  $S_1$  is given by the union of the subsets labelled by A, B, C, D, E, J, W, X, Y, Z. The  $P$ -upper approximation of  $S_1$  is given by the union of the subsets labelled by A, B, C, D, E, F, G, H, I, J, K, N, O, R, S, T, U, V, W, X, Y, Z. The boundary region is the same as the boundary region of  $S_0$ .

In general, some of the explanatory variables in  $P$  may be discarded. For instance, suppose that a subset  $Q \subseteq P$  of explanatory variables defines the same indiscernibility equivalence classes as the whole  $P$ . Then, the explanatory variables in the set difference  $P-Q$  are said to be *redundant*, i.e., do not contribute in any way to the precision of the classification. Minimal subsets of explanatory variables containing no redundant attributes are called *reducts*. In general, there may be more than one reduct, depending on the set of explanatory variables  $P$ . The intersection of all reducts, called the *core*, provides the most important explanatory variables, i.e., those that are essential to classify all objects. However, the core may be empty.

Knowledge obtained from Rough Set based analysis of experimental data is represented by means of production rules, which describe the relationships between premises and conclusions. The premise of a production rule is the conjunction of predicates of the form "explanatory variable = value." The conclusion of a production rule is of the form "dependent variable = value." Each rule is associated with its *absolute strength*, defined as the number of objects of the data set that satisfy its premise. To reflect different frequencies of occurrence of the different values for the dependent variable, we introduce the measure of *relative strength*. It is defined as the ratio of the number of objects (modules) that satisfy the premise of the rule to total number of objects that have the value of dependent variable of the consequence of the rule. Absolute and relative strengths provide an idea of the importance of the considered rule in explaining the behaviour of the dependent variable based on the explanatory variables. Background and algorithms to generate production rules based on reducts are described in [Paw91]. We remark that rule derivation may not deliver unique results. The reason is that rule derivation can be performed by starting from different reducts. Rules derived from the lower approximation are certainly valid and are called *deterministic rules*. Rules derived from the boundary region are possibly valid and are called *non-deterministic rules*.

## 5. Comparison of Results

### 5.1 Conceptual Criteria for Comparing the Analysis Approaches

As background for later evaluation, we list a set of desirable requirements for data analysis approaches should satisfy and compare the Rough Sets (RS) approach and Logistic Regression (LR) with respect to these assessment criteria. The list includes the criteria suggested in [BBT92].

1. *Avoidance of restrictive assumptions (relationships between variables, probability density distribution on the independent and dependent variable ranges, independence among explanatory variables).*

**LR:** No strict assumption is necessary on the functional relationship between independent and dependent variables, as Logistic Regression curves are able to approximate several different functional forms. Independence of observations is necessary to apply Maximum Likelihood techniques. However, independence of observations is also assumed true for the vast majority of statistical techniques. Data sets of larger size lead to higher confidence in the results.

**RS:** No assumption is necessary with respect to the probability distribution of variables and independence between variables. No assumption is necessary even on the size of the data set. Rough Set analysis can be applied to even small-size data sets.

2. *The modeling process needs to capture the impact of integrating all explanatory variables.*

**LR:** Univariate analysis is first carried out. Based on the results obtained, multivariate models are built by means of a multistage process.

**RS:** All explanatory variables are taken into account at the same time. Therefore, the Rough Sets approach may be considered as a kind of multivariate analysis. Rough Sets analysis allows for detection of core and redundant variables.

3. *Robustness to outliers.*

**LR:** This technique is less sensitive to outliers than other statistical techniques, though some kind of preliminary outlier analysis must be carried out.

**RS:** Outliers are reflected as "outlier rules," i.e., rules with low strength, which do not influence other rules in any form.

4. *Ability to handle interdependencies among explanatory variables.*

**LR:** Interdependencies are detected by means of statistical techniques that allow data analysts to check for correlations or associations between variables.

**RS:** Computation of reducts (see Section 4) provides information on the core and redundant explanatory variables. Therefore, Rough Sets analysis allows the identification of a set of essential explanatory variables (core) and a set of variables that depend on those in the core.

5. *Reliability of each individual prediction.*

**LR:** Given the values for the explanatory variables of an object, Logistic Regression provides an estimate for the probability of that object to be, say, faulty. In addition, Logistic Regression provides an estimate for the variance of this probability. Therefore, the reliability of each prediction can be assessed.

**RS:** There is no reliability measure from the original approach. Quality of prediction is measured by using different techniques such as ten-fold cross validation. Each individual rule is accompanied by its absolute and relative strength.

#### 6. *Management of inconsistent pieces of information*

**LR:** Inconsistent pieces of information are handled in a probabilistic way. Logistic Regression provides an estimate of the probability of an object to be classified in each of the possible classes of the explanatory variable. Therefore, Logistic Regression also tells how likely it is for the value of the explanatory variable of an object to be due to random.

**RS:** Inconsistencies are handled by introduction of lower and upper approximations, which originally motivated development of Rough Set theory.

#### 7. *Need for Discretization*

**LR:** Logistic Regression may be applied to interval or ratio level data—in addition to nominal and ordinal data—so there is no need for discretization.

**RS:** Rough Sets based analysis uses explanatory variables defined on a nominal or ordinal scale. Granularity and discretization techniques clearly influence analysis results.

#### 8. *Manipulation of different levels of granularity*

**LR:** Even though discretization is not needed, it may be carried out in Logistic Regression. A reason for this may be the way the data has been collected, to make the results less sensitive to possible data collection problems, because of which the actual data collected may be imprecise.

**RS:** Degree of accuracy of results is determined by the number of intervals. However, Rough Sets-based computations assume a low (about five) number of intervals. This often reflects degree of accuracy available from software engineering experimental data.

#### 9. *Dependencies on scales*

**LR:** The analysis can be carried out regardless of the type of scale.

**RS:** There is no dependency on type of scale, after discretization has been carried out.

## 5.2 **Logistic Regression: Analysis Results**

We summarize the experimental results by means of a multivariate model, based on the explanatory variables that were found significant and able to explain a relevant part of uncertainty on the dependent variable in the univariate analysis. More details may be found in [HL89]. Here is the multivariate model that we identified (for short,  $\pi = \pi(\text{AddedLOC}, \text{ModRate}, \text{ModKnow})$ )<sup>3</sup>:

$$\log(\pi/1-\pi) = -11.65 + 0.0286 \text{ AddedLOC} + 17.11 \text{ ModificationRate} + 3.53 (\text{ModuleKnowledge} - 1) - 0.06 \text{ AddedLOC} * \text{ModificationRate}$$

The term  $+3.53 (\text{ModuleKnowledge} - 1)$  shows how Logistic Regression deals with ordinal explanatory variables. The difference between the actual value (ModKnow) and a reference value (the numeric value 1, which is assumed to represent "high" knowledge of a module) is used as the actual covariate.  $\text{AddedLOC} * \text{ModificationRate}$ , being the product of two explanatory variables, is called an interaction term. Interaction terms are used to check whether the

---

<sup>3</sup>This formula shows the Logistic Regression equation of Section 3 in an equivalent form, to highlight the expression containing the explanatory variables.

combined effect of two variables has an impact on the dependent variable that is not captured by a purely linear model. The  $R^2$  we obtained is 0.46, which is quite high for Logistic Regression. In column "Estimate (std dev)," Table 1 reports on the estimates for the Logistic Regression coefficient and their standard deviation (in parentheses). Column "p" reports on the significance of the coefficients of the multivariate model. All of the explanatory variables of the multivariate model we have identified have a very high significance. For instance, the probability that the impact of AddedLOC on the dependent variable is due to chance is 0.0006, i.e., 0.06%.

| Term             | Estimate (std dev) | p      |
|------------------|--------------------|--------|
| Intercept        | -11.65 (2.78)      | 0.0000 |
| AddedLOC         | 0.0286 (0.0084)    | 0.0006 |
| ModRate          | 17.11 (5.72)       | 0.0028 |
| ModKnow[2]       | 3.53 (1.13)        | 0.0018 |
| AddedLOC*ModRate | -0.06 (0.024)      | 0.0123 |

Table 1. Logistic Regression: multivariate model.

When building the Logistic Regression equation, we have chosen to weight each module according to its number of faults. The rationale is that each (non) detection of a fault may be considered as an independent event. As a consequence, nonfaulty modules were weighted 1. At any rate, few faulty modules existed in the data set with more than one fault. Therefore, this choice does not lead to any dramatic change in results, though it somewhat biases the Logistic Regression equation towards faulty modules. In other words, the Logistic Regression equation obtained provides better results in classifying modules as faulty.

### 5.3 Rough Sets: Analysis Results

Experimental data were analyzed by the Rough Set-based data analysis system RoughDAS, developed at Institute of Computing Science of Technical University Poznan [SSt92]. Cluster analysis techniques have been used to carry out the discretization of experimental data into discrete variables in an automated way. The results are given in Table 2. In the Rough Sets approach, we have used the explanatory variables used in the Logistic Regression analysis, including the term AddedLOC\*ModificationRate, which was found relevant in the Logistic Regression analysis.

| Explanatory variable      | #inter-vals | int1 (=‘low’)   | int2 (=‘medium’) | int3 (=‘high’)  |
|---------------------------|-------------|-----------------|------------------|-----------------|
| ModificationRate          | 2           | [0.4 ... 30.7)  |                  | [30.7 ... 77.9] |
| SizeDifferenceRate        | 2           | [-0.12 ... 0)   |                  | [0 ... 1.5]     |
| ReuseRate                 | 2           | [52.0 ... 85.3) |                  | [85.3 ... 99.8] |
| AddedLOC*ModificationRate | 3           | [0 ... 42.4)    | [42.4 ... 87.8)  | [87.8 ... 317]  |
| DifferentBlocks           | 2           | [1 ... 22)      |                  | [22 ... 96]     |
| DeletedLOC                | 3           | [2 ... 59)      | [59 ... 129)     | [129 ... 594]   |
| ExpansionRate             | 2           | [37.3 ... 38.7) |                  | [38.7 ... 83.6] |

Table 2. Discretization of explanatory variables.

Discretization of the number of faults in a module  $m$  in version 6.1 leads to the discretized dependent variable Faulty6.1 with values  $Faulty6.1 = 0$  if  $m$  contains does not faults, and  $Faulty6.1 = 1$  if module  $m$  contains faults.

Application of Rough Sets analysis has yielded only deterministic rules, whose premises are mutually exclusive. Determination of core attributes and reducts is one of the most important results of Rough Set based data analysis (see Section 4). Eight core variables were found: the explanatory variables of Table 2 plus ModuleKnowledge.

Nineteen rules were generated, all of which are deterministic ones. In Table 3, we report a few of the rules with greatest relative strength. In this table, we show the four rules with the greatest relative strength for nonfaulty modules (Faulty6.1 = 0) and the three rules with the greatest relative strength for faulty modules (Faulty6.1 = 1).

| Premise  | Consequence   | Relative Strength |
|--|---------------|-------------------|
| (ModificationRate=low) & (ExpansionRate=high) & (DeletedLOC=low)   | Faulty6.1 = 0 | 47.1%             |
| (ReuseRate=low) & (ModuleKnowledge=high)   | Faulty6.1 = 0 | 17.6%             |
| (SizeDifferenceRate=low) & (ReuseRate=low)   | Faulty6.1 = 0 | 16.8%             |
| (SizeDifferenceRate=high) & (DifferentBlocks=high) & (AddedLOC*ModificationRate=low)                           | Faulty6.1 = 0 | 15.1%             |
| (ModificationRate=high) & (SizeDifferenceRate=high) & (ModuleKnowledge=low) & (AddedLOC*ModificationRate=high) | Faulty6.1 = 1 | 27.2%             |
| (ReuseRate=high) & (AddedLOC*ModificationRate=high)  | Faulty6.1 = 1 | 18.2%             |
| (ModificationRate=high) & (SizeDifferenceRate=high) & (DeletedLOC=high) & (ModuleKnowledge=low)                | Faulty6.1 = 1 | 18.2%             |

Table 3. Rules with greatest relative strength generated by the Rough Sets data analysis.

Among these rules, the first one is the most powerful one. It states that if modification rate is low, expansion rate is high, and number of deleted lines of code is low then number of faults in version 6.1 is expected to be zero.

As a guideline for the relative importance of the different attributes, we have ranked them according to their number of occurrences. Using '>' for order relation between pairs of attributes such that A>B means that A is a stronger influence factor than B results in

DeletedLOC(73.1%)>ModificationRate(52.3%)>ReuseRate(50.8%)>  
 ExpansionRate(46.9%)>AddedLOC\*ModificationRate(40.8%)>  
 SizeDifferenceRate(33.8%)>ModuleKnowledge(26.9%)> DifferentBlocks(19.3%)

where the number of occurrences is given in parentheses. We have also combined different explanatory variables and looked for the degree of rule coverage from these combinations. Therein, a rule is covered, if at least one of the considered attributes from the chosen subset occurs in the premise of the rule. The question is, which combinations have highest coverage for different cardinalities of attribute subsets. Here are the best attribute combinations of cardinalities 1, 2, and 3 are described:

cardinality=1: DeletedLOC  
 cardinality=2: DeletedLOC, ModificationRate  
 cardinality=3: DeletedLOC, ModificationRate, ReuseRate

#### 5.4 Comparative Analysis of Results

We will use the following three indices to assess and compare the results obtained with the two data analysis techniques, as follows:

1. *Overall Correctness*: Proportion of modules that have been classified correctly. This parameter provides information on how well a technique classifies modules, regardless of the category in which the modules have been classified.
2. *Faulty Module Completeness*: Proportion of faulty modules that have been classified as faulty. This parameter provides information on how many of the faulty modules have been correctly identified by the data analysis technique. Conversely, this parameter provides information on the risk of not having identified faulty modules, and, therefore, not having tested or inspected them more carefully.
3. *Faulty Module Correctness*: Proportion of modules that have been classified as faulty and were faulty indeed. This parameter provides information on the efficiency of a technique, i.e., on the proportion of modules that are potential candidates for further verification. Conversely, this parameter provides information on the proportion of modules that are not faulty and have undergone further verification anyway.

The idea underlying the use of indices 2 and 3 is that faulty modules may be considered more relevant than nonfaulty ones in this application context and in several others. Faulty modules may cause serious failures, i.e., if faults are not removed. Therefore, faulty modules should undergo additional verification activities, besides those carried out on all modules.

We now provide the classification results obtained with Logistic Regression and Rough Sets.

##### ***Logistic Regression.***

Classification results are summarized in Figure 2 and Table 4. We have assumed a threshold of  $p = 0.5$  to predict a module as faulty, i.e., a module was predicted to be faulty only if the estimated probability for it to be faulty exceeded 0.5. We want to remark that the choice of this threshold value is always a subjective matter. Other choices are possible (e.g., the actual rate of faulty modules), which may yield different classification results.

|                      | Predicted NonFaulty 6.1 | Predicted Faulty 6.1 | Total |
|----------------------|-------------------------|----------------------|-------|
| Actual NonFaulty 6.1 | 80%                     | 11.5%                | 91.5% |
| Actual Faulty 6.1    | 2.3%                    | 6.2%                 | 8.5%  |
| Total                | 82.3%                   | 17.7%                | 100%  |

Table 4. Logistic Regression: classification results.

Table 5 shows the values for the three assessment parameters we have introduced:

| Overall Completeness | Faulty Module Completeness | Faulty Module Correctness |
|----------------------|----------------------------|---------------------------|
| 86.2%                | 73%                        | 35%                       |

Table 5. Logistic Regression: assessment indices.

{ EMBED Word.Picture.6 }

Figure 2. Logistic Regression: classification results.

### **Rough Sets**

Ten-fold cross validation test was performed to validate quality of prediction results. This test divides the objects into ten disjoint subsets of equal size. Each subset is excluded from the data set, one at a time. The remaining set of objects is used to generate rules, which are used to classify the objects of the excluded subset. Accuracy of classification can then be assessed, since the actual classification of the objects in the excluded subset is known. The corresponding classification results obtained by means of Rough Sets analysis are summarized in Figure 3 and Table 6. Three values are present for Predicted Faulty 6.1, as follows

- Predicted Faulty 6.1 = 0 for modules predicted as nonfaulty
- Predicted Faulty 6.1 = 1 for modules predicted as faulty
- Predicted Faulty 6.1 = -1 for modules for which no prediction was made.

{ EMBED Word.Picture.6 }

Figure 3. Rough Sets: classification results.

|                      | Not Classified | Predicted NonFaulty 6.1 | Predicted Faulty 6.1 | Total |
|----------------------|----------------|-------------------------|----------------------|-------|
| Actual NonFaulty 6.1 | 0.8%           | 86.9%                   | 3.8%                 | 91.5% |
| Actual Faulty 6.1    | 2.3%           | 3.9%                    | 2.3%                 | 8.5%  |
| Total                | 3.1%           | 90.8%                   | 6.1%                 | 100%  |

Table 6. Rough Sets: classification results.

The above results lead to the results for our three assessment parameters shown in Table 7.

| Overall Completeness | Faulty Module Completeness | Faulty Module Correctness |
|----------------------|----------------------------|---------------------------|
| 89.2%                | 27.1%                      | 37.7%                     |

Table 7. Rough Sets: assessment indices.

Based on the experimental results, the maintainers have to make decisions on which modules should undergo further verification activities. There are two ways in which the maintainers may take into account the information obtained by means of the above analysis.

*Case a.* The maintainers may decide that all unclassified modules do not deserve any further treatment. In other words, unclassified modules are treated as nonfaulty ones. Table 8 shows the corresponding classification results.

|                      | Predicted NonFaulty 6.1 | Predicted Faulty 6.1 | Total |
|----------------------|-------------------------|----------------------|-------|
| Actual NonFaulty 6.1 | 87.7%                   | 3.8%                 | 91.5% |
| Actual Faulty 6.1    | 6.2%                    | 2.3%                 | 8.5%  |

|       |       |      |      |
|-------|-------|------|------|
| Total | 93.9% | 6.1% | 100% |
|-------|-------|------|------|

Table 8. Rough Sets-Case a: new classification results.

The new values for the three assessment parameters are in Table 9.

|                      |                            |                           |
|----------------------|----------------------------|---------------------------|
| Overall Completeness | Faulty Module Completeness | Faulty Module Correctness |
| 90%                  | 27.1%                      | 37.7%                     |

Table 9. Rough Sets-Case a: new assessment indices.

*Case b.* The maintainers decide that all unclassified modules—and all modules classified as faulty—should be further verified. Therefore, it is as though the maintainers classify as faulty all those unclassified modules. Table 10 shows the corresponding classification results.

|                      |                         |                      |       |
|----------------------|-------------------------|----------------------|-------|
|                      | Predicted NonFaulty 6.1 | Predicted Faulty 6.1 | Total |
| Actual NonFaulty 6.1 | 86.9%                   | 4.6%                 | 91.5% |
| Actual Faulty 6.1    | 3.9%                    | 4.6%                 | 8.5%  |
| Total                | 90.8%                   | 9.2%                 | 100%  |

Table 10. Rough Sets-Case b: new classification results.

The new values for the three assessment parameters are in Table 11.

|                      |                            |                           |
|----------------------|----------------------------|---------------------------|
| Overall Completeness | Faulty Module Completeness | Faulty Module Correctness |
| 91.5%                | 54.1%                      | 50%                       |

Table 11. Rough Sets-Case b: new assessment indices.

Concluding, the results in Tables 6 and 7 are the ones that more closely conform to the ideas behind the classification indices we use. On the other hand, the results in Tables 10-11 are those that maintainers are more likely to use for practical purposes. For notational convenience, in what follows, we will use the abbreviation RS(a) to denote the results in Tables 8 and 9, and RS(b) to denote the results in tables 10 and 11. Results in tables 6 and 7 will be referred to as RS(0).

### ***Comparison.***

Before proceeding into the details of the comparison between the two techniques, we want to warn the reader that it is clearly impossible to extract conclusive and general evidence about strengths and weaknesses of the two approaches from only one data set. Many more numerical and comparative studies are necessary for drawing reliable conclusions. At any rate, based on the above experimental results, these are the preliminary comparisons that we may draw from our experience.

1. RS(0), RS(a), and RS(b) all classify modules as either faulty or nonfaulty better than Logistic Regression. However, the difference in Overall Correctness is quite small.
2. RS(0) and RS(a) perform much worse than Logistic Regression in terms of Faulty Module Completeness, and RS(b) still performs worse than Logistic Regression. These results are only partially due to the fact that, in the Logistic Regression analysis, each faulty module was weighted by the number of faults it contained. Actually, the fact that Logistic Regression provides more accurate results for faulty modules by weighting modules as we did may be considered an advantage of Logistic Regression, since faulty modules are the important ones for our application purposes. Weighting modules by the number of their faults will not change the classification results obtained by the Rough

Sets approach. Only relative strength of rules and number of occurrences of attributes in rules are affected.

In addition, a reclassification such as that of Case b may be practically used when the number of unclassified modules is not high. If this is not the case, the maintainers may not decide to further verify large numbers of modules about whose correctness the Rough Sets approach provides no information. Therefore, the Rough Sets approach appears less effective in identifying those module that may be considered the most critical ones.

3. On the other hand, RS(0), RS(a), and RS(b) perform better than Logistic Regression in terms of Faulty Module Correctness. Therefore, the Rough Sets data analysis allows the maintainers to verify a smaller number of modules that will turn out to be nonfaulty, therefore reducing the cost for additional verification.
4. The Rough Sets approach uses many more explanatory variables than Logistic Regression. Data collection is an expensive and time-consuming activity. Therefore, one should also take into account this cost in the evaluation of the techniques, in addition to the costs due to the failures caused by those faulty modules have not been identified as such and the unnecessary verification activities for those nonfaulty modules that have been erroneously identified as faulty. However, not all of the explanatory variables that appears in the rules obtained by means of Rough Sets may require additional data collection effort. For instance, Reuse, which appears in the Rough Sets rules and does not appear in the Logistic regression equation, can be computed based on data that are also used for the computation of DeletedLOC. On the other hand, the computation of DifferentBlocks requires data that are not used to compute other explanatory variables appearing in the rules. Therefore, the difference in data collection cost may not be too large.
5. Logistic Regression and Rough Sets agree on some of the most important factors. ModificationRate appears as an important explanatory variable in both of them. By examining correlations between explanatory variables, one finds that AddedLOC and DeletedLOC are highly correlated (linear correlation = 0.91). One of them appears in RS and the other one appears in LR<sup>4</sup>. The most important differences are due to the fact that ModuleKnowledge appears in the Logistic Regression multivariate model as an important explanatory variable, while it is not as important in the Rough Sets analysis, and that ReuseRate does not appear as a variable in the Logistic Regression multivariate model.
6. The Rough Sets approach does not provide reliable indications on the significance of prediction, while the Logistic Regression approach is able to evaluate the confidence with which results have to be considered.
7. A weakness of RS is its dependency on discretization methods, which may influence results considerably. One contribution to this problem is to devise automatic discretization methods, e.g, based on clustering techniques or the minimal class entropy method [CG94]. Independently, inclusion of experienced people from the application domain is mandatory for interpreting experimental data in Software Engineering.

The fact that the both techniques agree on some of the important attributes may be thought encouraging, for small data sets, for which the results obtained by Logistic Regression may not have high significance. Overall, the fact that LR performs better should be expected anyway, since the data set was not of a particularly small size.

---

<sup>4</sup>Building the Logistic Regression multivariate model requires checking for correlations or associations between explanatory variables. If two highly correlated explanatory variables are used in the same multivariate model, then their coefficients are likely to be unstable and not to be both significant. This is a side effect of the fact that both explanatory variable actually capture the same phenomenon.

## 6. Summary and Conclusions

We have studied and compared Logistic Regression and Rough Set, two techniques for the analysis of software measurement data. Our study shows that both techniques are able to identify a similar set of relevant factors. This means that we may have greater confidence in the results provided by either technique, all the more since the two techniques have very different mathematical bases. This provides us with greater confidence in making decisions based on the experimental results. We have compared the two techniques by means of their classification ability, and we have discussed their strengths and weaknesses. At any rate, it must be kept in mind that the primary aim of our empirical study was *not* to build a very accurate predictive model. Instead, our main objective was to get better insights into the maintenance environment, as the *purpose* field of our GQM goal says. We believe that it would have been unrealistic to start a measurement program with a *prediction* goal. This is possible only after the important factors that affect the studied phenomenon have been identified. In turn, this is possible only after several empirical studies, each of which refines the previous ones, have been carried out.

From our empirical study a strong preference between the two techniques can not be derived. Both techniques, and others, will have to be studied and compared in different application environments and on data sets with different sizes and characteristics. Therefore, our future activities will encompass further empirical studies, to assess strengths and weaknesses of Logistic Regression, Rough Sets, and other data analysis techniques.

## Acknowledgements

We greatly appreciate the computational support delivered by Robert Susmaga from Technical University of Poznan for performing computations with RoughDAS. We also want to thank the DATATRIEVE™ team for their support. Last, but not least, success of the whole project was based on the permanent discussion among all participants from Robert Bosch GmbH, CEFRIEL, Digital Equipment SPA, Schlumberger RPS and Software-Technology-Transfer-Initiative Kaiserslautern.

## References

- [BBT92] L.C. Briand, V.R. Basili, and W.M. Thomas. A Pattern Recognition Approach for Software Engineering Data Analysis. *IEEE Transactions on Software Engineering*, 18(11): 931-942, Nov. 1992.
- [BCR94] V.R. Basili, G. Caldiera, and H.D. Rombach. Experience Factory. In J.J. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 1, pages 469-476. John Wiley & Sons, 1994.
- [BMB94] L. Briand, S. Morasca, and V.R. Basili. Defining and Validating High-Level Design Metrics. *Technical Report*, CS-TR-3301, University of Maryland, November 1994. Submitted for publication.
- [BR88] V.R. Basili and H.D. Rombach. The TAME Project: Towards Improvement-oriented Software Environments. *IEEE Transactions on Software Engineering*, SE-14(6): 758-773, June 1988.
- [BW84] V.R. Basili and D.M. Weiss. A Methodology for Collecting valid Software Engineering Data, *IEEE Transactions on Software Engineering*, SE-10(6): 728-738, Nov. 1984.
- [CEMPwww] <http://uomo.informatik.uni-kl.de:2080/Cemp/CEMP.html>
- [CEMP95] Customized Establishment of Measurement Programs (CEMP), December 1995. ESSI Project#10358, Final Report.
- [CG94] M.R. Chmielewski and J. Grzymala-Busse. Global Discretization of Continuous Attributes as Preprocessing for Machine Learning. *Proceedings of Third International Workshop on Rough Sets and Soft Computing*, 1994, p.294-301.

- [HL89] D.W. Hosmer, Jr., and S.Lemeshow. Applied Logistic Regression. John Wiley & Sons, Inc., 1989.
- [HOR96] B. Hoisl, M. Oivo, D.H. Rombach, G. Ruhe, F. van Latum, R. van Solingen. Shifting to Goal-Oriented Measurement in Industrial Environments, submitted to IEEE Software.
- [Paw91] Z. Pawlak. Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic, Dordrecht, 1991.
- [PGSZ95] Z. Pawlak, J. Grzymala-Busse, R. Slowinski and W. Ziarko. Rough Sets. Communications of the ACM 38(1995), 89-95.
- [Ruh96] G. Ruhe. Rough Set Based Data Analysis in Goal-Oriented Software Measurement. Proceedings of the Third Symposium on Software Metrics, March 1996 in Berlin, IEEE Computer Society Press, 10-19.
- [SSt92] R. Slowinski and J. Stefanowski. Rough DAS and Rough-Class Software Implementations of the Rough Set Approach. In R. Slowinski, editor, Intelligent Decision Support: Handbook of Applications and Advances of the Rough Set Theory, pages 445-456. Kluwer Academic Publishers, 1992.