ASSOCIATION ANALYSIS OF SOFTWARE MEASURES

W. PEDRYCZ*†, G. SUGI* and M. G. CHUN†
* Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada T6G 2G7
† Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland
† School of Electrical and Electronics Engineering, Chungbuk National University, Sun 58, Cheongjuang, Haeng Dae-gu, Cheongju, 361-763 Korea

Received 11 September 2001
Revised 2 November 2001
Accepted 28 November 2001

Software measures (metrics) provide software engineers with an important means of quantifying essential features of software products and software processes such as software reliability, maintainability, reusability and size. Software measures interact between themselves. Some of them may be deemed redundant. Software measures are used to construct detailed prediction models. The objective of this study is to pursue an association analysis of software measures by revealing dependencies (associations) between them. More specifically, the introduced association analysis is carried out at the local level by studying dependencies between information granules of the software measures. This approach is contrasted with a global level such as e.g., regression analysis. We discuss the role of information granules as meaningful conceptual entities that facilitate analysis and give rise to a user-friendly, highly transparent environment.

Keywords: Software measures; associations; rules; fuzzy sets; information granules; rule-based models.

1. Introduction

Intelligent Data Analysis (IDA) and data mining [2, 6, 9] have emerged as a new and promising direction of research that intends to make sense of multivariable data. The agenda of IDA (as well as data mining) is very broad. IDA provides with an insight into the data by revealing the main trends and relationships between variables. By analyzing data in the IDA environment, one derives useful patterns that can be refined and used afterwards in the form of detailed models. In this way, those patterns help make predictions. In a nutshell, IDA attempts to operate at a level that is comfortable to the human user performing at the level of specificity that allows the user to build a global picture about the underlying structure in data sets. The duality of the quantitative-qualitative approach is particularly appealing in Software Engineering. Being abstract to a very high extent, not being guided
by any laws of physics and exhibiting a high level of variability combined with a uniqueness of software projects, software artifacts (products and processes) and their experimental measurements require careful treatment. In particular, modeling has to be carried out at various levels of specificity and produce models of varying level of specificity (granularity) by revealing different levels of details. Software measures, cf. [1, 3, 12, 16] are crucial descriptors of software products and processes that help quantify their complexity and assess effort required to maintain software or make some essential changes to it.

In this study, we pursue a concept of associations formed at the level of information granules [5-7, 13-16] regarded as basic building blocks used in data analysis. Roughly speaking, associations are highly homogeneous and experimentally strongly justifiable information granules capturing the essence of given data at hand. To illustrate the concept, refer to Fig. 1 which illustrates various clouds of two-dimensional experimental data. When analyzing this data set from a designer’s perspective, we may easily identify several regions of concentration of data emerging in a form of some clusters (associations).

![Fig. 1. An example data set visualising some well-formed regions of high density of data along areas of low density and poorly visible structure.](image)

As seen in Fig. 1, in some regions, the data points clearly exhibit high density and take on some well-defined shape (say, hyperspherical or single-line structure). In some other regions, there are few scattered points with no clear and strong dependency. Intuitively, one can delineate some highly populated regions and view them as highly representative to the entire data set. Subsequently, all such associations can be subject to further detailed analysis and form the basis for construction of a detailed model quantifying specific dependencies between the variables. It becomes apparent that association analysis can be viewed as a preliminary step in data analysis that may afterwards be refined by building detailed models. In this sense, constructing data associations may serve as blueprint of further, more detailed and mathematically specific models (say, regression models, correlation coefficients, neural networks and alike). In Software Engineering we are concerned with a panoply of detailed quantitative and more general yet qualitative models. The proposed association analysis bridges a gap between these two approaches. Information granules help conceptualize the associations by selecting the relevant level of granularity (namely, a level of sets or fuzzy sets). By being structure-free, associations support a general, more qualitative view at essential modeling pursuits in Software Engineering not necessarily proceeding immediately with detailed models (say nonlinear functions) for the software measures encountered in the problem.

This paper is organized into 8 sections. First, we concentrate on the design of data associations (Sec. 2). We start by discussing how information granules are built over a single variable (software measure). The study concentrates on fuzzy sets as a convenient form of information granulation. A method of fuzzy equilibration supports the development of fuzzy sets. In the sequel, we construct multidimensional (multivariable) information granules and show that they can be regarded as fuzzy relations. In Sec. 3, we argue that all associations exhibiting a strong experimental support are organized in the form of an agenda of associations. Associations can be converted into rules (conditional “if-then” statement). This step is discussed in Sec. 4. Further reduction process of rules leading to a concise description of the relationships is studied in Sec. 5. A detailed characterization of the software measures is included in Sec. 6 including their relevance and discriminatory aspects. Conclusions are covered in Sec. 7.

This study exhibits a strong experimental slant. We use a well-known MIS data set [4] to illustrate fundamental concepts and discuss all computational facets of the design as well as reveal interesting association patterns in the data set itself. This data set consists of a number of commonly used software measures (such as lines of code, McCabe complexity, bandwidth, etc.) describing a collection of software modules. The data set also includes a number of changes made to the individual modules during its testing phase.

2. The Design of Data Associations

In this section, we define a concept of information granules and reveal how these are used in the design of associations. The proposed development methodology involves two key phases: (1) building basic information granules for each individual variable, and (2) forming associations with the use of these information granules.

2.1. Building information granules for a single variable

For each variable (measure), we specify a certain number of information granules. By information granules we mean a collection of elements that are collected together owing to their similarity or functional coherences. Information granules exhibit a well-defined semantics easily comprehended by humans and therefore could serve as useful descriptors of the problem. For instance, when talking about lines of code, it is convenient to talk about small, medium, and large size of code rather than using a single number, say 70K code. These terms like small, medium and alike are examples of information granules: they are easy to understand, highly descriptive, and handy when communicating findings about the given data set. Similarly, they
are helpful in making design decisions. The way in which we proceed at the formal end may vary. Such information granules can be represented as sets, fuzzy sets, rough sets, shadow sets, etc. The choice of fuzzy sets is of interest in this study. Fuzzy sets describe concepts (information granules) that exhibit a phenomenon of gradual membership. The term "small size of code" falls under this category; definitely we do not have any single numeric value, say 60K that can distinguish between the size of code being small and not small. One may state that this dichotomization sounds very superficial. Instead, we can envision smooth boundaries capturing the essence of this concept. See Fig. 2 showing two example membership functions of the small size of code.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{example_membership_functions.png}
\caption{Information granules of small size of code as an example of a fuzzy set; note that its membership function captures an effect of continuous changes in membership grades (a) nonlinear membership function; (b) piecewise (trapezoidal) membership function.}
\end{figure}

The question arises as to the determination of the fuzzy sets (information granules). First, we have to decide on the form of the membership functions that describe a way in which the membership grades vary over the space. The simplest model is a triangular or trapezoidal fuzzy set. In the case of triangular fuzzy set, we have

a clear interpretation as illustrated in Fig. 2: the modal value is the most likely (typical) value of the information granule (in this case we are dealing with a range of values between 0 and 20 KLOC) whereas "a" and "b" (here equal to 20 and 100, respectively) are the lower and upper bounds, respectively. The membership changes in a nonlinear fashion between the bounds and the typical value.

While each fuzzy set comes with a transparent semantics, its parameters need to be adjusted and should reflect the nature of the experimental data which these fuzzy sets have to granulate. Fuzzy equalization [8] forms a simple yet efficient algorithm of determining the parameters of the fuzzy sets. In a nutshell, the concept of fuzzy equalization originates from the idea of fuzzy events [13]. Consider a fuzzy set \( A \) defined over some universe of discourse \( X \). In this universe, we have a family of experimental numeric data that are conveniently described by a certain probability density function (pdf) \( p(x) \). Then the probability of the fuzzy event (fuzzy set \( A \)) is computed in the form

\[ P(A) = \int_X A(x)p(x)dx. \]  

(1)

Essentially, \( P(A) \) is an expected value of \( A \). Interestingly, the probability of the fuzzy event carries a straightforward interpretation. A fuzzy set can be viewed as meaningful if its probability \( P(A) \) is equal or exceeds a certain critical value \( \alpha \), namely \( P(A) \geq \alpha \). If this holds, we say \( A \) is experimentally justified (valid). The monotonicity property holds: if \( A \subseteq A' \) (that is the granularity of \( A \) is higher than that of \( A' \)) then the corresponding probabilities satisfy the monotonicity condition: \( P(A) \leq P(A') \).

Now let us consider a family of fuzzy sets, say \( A = \{ A_1, A_2, \ldots, A_n \} \) is given. To make each \( A_i \) meaningful (in the above sense of experimental justification), we request that there is an equal level of experimental evidence behind each of them equal to \( 1/\alpha \), that is

\[ P(A_i) = 1/\alpha. \]  

(2)

According to the above equalization condition, fuzzy sets become more specific (detailed) in the regions of \( X \) where the pdf attains some local maximum. On the other hand, in the areas of low values of pdf, we need fuzzy sets of broader support to gain sufficient experimental evidence (see Fig. 3).

When dealing with a series of triangular fuzzy numbers with a 1/2 overlap between successive members of \( A \) with the first as well as the last fuzzy set being defined by a trapezoidal membership function, the equalization algorithm can be shortly described as follows.

[Step 1] Specify the number of elements of linguistic terms in \( A \), \( \alpha \)
[Step 2] Start from the lower bound of the universe \( X \) denoted by \( \alpha_{\text{min}} \).
[Step 3] Keep moving "a" towards higher values (starting from \( \alpha = \alpha_{\text{min}} \)) and calculate the value of the integral (1). Stop once the integral (1) has reached
the value equal to $1/(2c)$ that is

$$\int_{-\infty}^{\infty} A_1(x)p(x)\,dx = 1/(2c).$$

(3)

[Step 4] Determine the upper bound, $\phi$, of the support of $A_1$ in such a way that the probability of the fuzzy event becomes equal to

$$\int_{-\infty}^{\phi} A_1(x)p(x)\,dx = 1/(2c).$$

(4)

[Step 5] For the triangular fuzzy sets (starting from $A_2$ and $A_3, A_4$, etc.), compute the probability of the respective fuzzy events such as

$$\varphi = \int_{-\infty}^{\phi} A_2(x)p(x)\,dx.$$  

(5)

[Step 6] Optimize the upper bound of the support (3) of the same fuzzy set such that it satisfies the condition (refer to Fig. 4)

$$\int_{-\infty}^{\phi} A_3(x)p(x)\,dx = 1/c - \varphi.$$  

(6)


It is worth stressing that we are usually furnished with discrete data rather than continuous probability density functions. In this case, the equalization algorithm should use the sums rather than integrals. For instance, for the data $\{x_1, x_2, \ldots, x_N\}$, the first expression (3) assumes the form

$$\sum_{k=x_1 \text{ to } c} A_1(x_k) = N/(2c).$$

(7)
In the sequel, the optimization is carried out with respect to the unknown parameter (a) in the manner already described.

Proceeding with the MIS data and using the equalization algorithm, the resulting fuzzy sets are illustrated in Fig. 5. These are represented in the form of triangular membership functions for one selected software measure, namely LOC and the number of changes. It is evident that the size (spread) of the fuzzy set is very much affected by the density of experimental data. A concentration of experimental data yields a narrow range of the membership functions in the corresponding regions of high values of the probability function. And, conversely, if the data are sparse in some regions, the fuzzy set grows bigger to compensate for that and gain the required level of experimental evidence.

Table 1 summarizes the fuzzy sets for all software measures. In this table, we exploit the standard notation used in fuzzy sets, namely a triangular fuzzy number (set) is fully described by three parameters (lower and upper bound and a modal value) whereas a trapezoidal fuzzy set is characterized by four parameters. Referring to Fig. 6 that provides us a detailed illustration of these parameters. Further on, we will be describing these fuzzy sets by a list of three or four numbers, say \( A = (a, m, b), B = (a, m, n, b), \) etc.

In the ensuing experiments and analysis, we have decided to proceed with four fuzzy sets for each software measure as these information granules assume a transparent interpretation. The lowest one can be interpreted as low, while the fourth one depicts high values of the respective software measure. Similarly, the two intermediate fuzzy sets can be interpreted as those modeling intermediate medium values of the software measure.

![Fig. 6. Triangular and trapezoidal fuzzy numbers (fuzzy sets) — a basic notation.](image)

2.2. Building associations as multidimensional experimental structures

The fuzzy sets defined in the individual coordinates (spaces) are generic components forming associations. Consider "n" variables in the data set. The fuzzy sets in each coordinate is denoted by \( A_1, A_2, \ldots, A_n \) (first variable), \( B_1, B_2, \ldots, B_n \) (second variable), etc. Formally speaking, an association \( A \) is a Cartesian product of any combination of the fuzzy sets for each variable. Formally, the association \( A \) comes in the form

\[
A = A_1 \times B_1.
\]

The operation of combining the membership grades is realized using a \( t \)-norm [8]. In particular, one may consider a minimum operation as one of the plausible options. This gives rise to the expression

\[
A(x, y) = (A_1 \times B_1)(x, y) = \min(A_1(x), B_1(y)).
\]

An illustration of the associations in the case of two variables is illustrated in Fig. 7. The association \( A \) is uniquely specified by a sequence of indexes identifying the coordinates of the fuzzy sets. Thus, we may characterize \( A \) through a sequence of indexes \( (i_1, i_2, \ldots, i_n) \) where \( i_k \) describes an index of the fuzzy set in the \( k \)-th coordinate (variable) contributing to this specific association. Say, for \( n = 3 \), a certain association \( B \) comes in the form \( B = (2, 4, 1) \).

In light of the introduced design procedure, several interesting features are worth emphasizing:
Association Analysis of Software Measures

3. Forming an Agenda of Associations

The number of all possible associations is tremendous. With "n" variables and "p" information granules (fuzzy sets or sets) defined for each of them, we end up with \( p^n \) possible associations. Only a small fraction of these is meaningful that is justified (supported) by the experimental evidence (data at hand). For a uniform distribution of data (which is usually not the case), the probability of data supporting each association is \( \frac{1}{2^p} \). Most of the associations are void meaning that there is no data behind them. Naturally, a straightforward criterion to select meaningful associations would be to quantify a level of experimental evidence behind them. Two measures of experimental support are of interest here.

1. \( \sigma \)-count [6]: For association \( A \), refer to (8) we compute

\[
\sigma(A) = \sum_{k=1}^{N} \min(A_i(x_k), B_j(y_k))
\]

2. Cardinality: This definition applies to the support of \( A \) and involves counting all elements of the data set falling under the support of \( A \). No membership grades are used in this case. The computations follow the formula

\[
\text{card}(A) = \sum_{k=1}^{N} \min(\chi_A(x_k), \chi_B(y_k))
\]

where \( \chi_A \) and \( \chi_B \) are the characteristic functions of the supports of the corresponding fuzzy sets, namely

\[
\chi_A(x) = \begin{cases} 1 & \text{if } x \in \text{supp}(A) \\ 0 & \text{otherwise} \end{cases}
\]

(let us recall that by support of the fuzzy set we mean all elements of the universe of discourse over which the membership function attains non-zero values).

The most meaningful associations are combined in the form of an agenda, that is a collection of associations with the highest values of the cardinality and arranged...
in decreasing order with respect to the cardinality value. The size of the agenda is
selected in such a way that all associations there cover a certain percentage of the
overall data set (say, 70%). As emphasized, the number of all possible associations
is high. A brute-force enumeration can work for 10-12 variables and 3-4 fuzzy sets
in each variable. Quite quickly, this approach is not feasible. Fortunately, there is a
simple solution that helps overcome the problem. A complete enumeration can be
combined by a simultaneous pruning technique, which avoids exploring combinations
of fuzzy sets not supporting enough data. The pruning is possible because the
cardinality (and \( \sigma \)-count) decreases its value once more variables come into play.
Moreover, at each stage of expanding the association, see Fig. 9, some combinations
are pruned and the related expansions are not pursued any longer. This immensely
reduces the number of possibilities that need to be investigated.

For the MIS data under study we have considered all possible combinations (this
is feasible as we have used 4 fuzzy sets for each of the 12 variables — this altogether
giving rise to \( 4^{12} \) combinations). The plot of the \( \sigma \)-count for each of them is shown

Fig. 9. The pruning of associations in their process of successive developments; note that at each
level when adding one extra variable, only a portion of the most meaningful Cartesian products
is retained.

in Fig. 10. Note that there are a small number of them assuming significant values
that qualify them as meaningful associations.

50 of the most dominant (experimentally relevant) associations are listed in
Table 2. The left-hand side of the table summarizes all those associations, each
of them in the form of a string of fuzzy sets (their labels), say 1 1 1 . . . 1. On the right-hand
side of the table, the levels of coverage of the associations (provided in terms of
their cardinalities) are included. The coverage is determined for the training set
(which is 2/3 of the overall MIS dataset). Also given is an average coverage of the
associations on the testing set (the testing set is randomly selected out of the entire
dataset; note that 1/3 of it has not been used for training purposes. The result on
the testing set is averaged over 10 runs).

This table reveals interesting findings from the standpoint of Software Engineering.
We note that the association with the strongest experimental support involves
all software measures of low values (1 1 . . . 1). Second, the association between high
values of the software measures (4 4 . . . 4) is also strongly supported at the numeric
ground. Interestingly, the third group of associations is the one with intermediate
linguistic values of the software measures (viz., those involving 3s and 2s). The associ-
ations with the low values of the software measures (with 1s) and few 2s (those
shown at the bottom of Table 2) are weakly supported on the experimental
ground.

4. From Associations to Rules

Associations are direction-neutral local structures in which we do not distinguish be-
tween input (independent) and output (dependent) variables. This obviously leads
to a significant level of generality of such constructs. In addition to being direction-
free, the associations are structure-free. They do not confine themselves to any
Table 2. The most dominant associations along with their data coverage characterization for the training and testing portion of the data set.

<table>
<thead>
<tr>
<th>Associations</th>
<th>Training Data</th>
<th>Testing Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 4 4 4 1 1 1 1 4 1 1 1 1</td>
<td>20</td>
<td>24.2</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 3 4 4 4 4 1</td>
<td>23</td>
<td>12.8</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>17</td>
<td>5.8</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>20</td>
<td>8.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>16</td>
<td>7.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>15</td>
<td>7.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>16</td>
<td>6.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>14</td>
<td>5.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>12</td>
<td>4.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>11</td>
<td>3.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>10</td>
<td>3.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>9</td>
<td>3.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>8</td>
<td>2.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>6</td>
<td>1.6</td>
</tr>
<tr>
<td>4 4 4 4 4 4 4 4 4 4 4 4 4</td>
<td>5</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Data Coverage(%) | 55 | 56.6 |

Specific type of analytical relationships between the software measures. Treating associations, and the agenda, in particular, as a starting point for any detailed modeling pursuit, several main directions can be envisioned:

(a) identifying rules
(b) constructing detailed parametric models for each association

In contrast to associations, rules are direction-sensitive structures. Variables have to be declared as independent or dependent. The independent variables come as a condition part of the rule. The dependent variables appear in the conclusion part. Once we have identified the variables, each association can be viewed as an individual rule

— if condition1 and condition2 and ... and conditionn then conclusion1 and conclusion2 and conclusionm

where r + p = n. In contrast to associations, rules cannot be treated individually but have to be analyzed and utilized en block. Some rules resulting from the above transformation of the associations could be in conflict. Two rules are said to be in conflict if they have identical condition parts and have different conclusion parts.

Considering the MIS data, it is natural to treat all software measures as independent variables and view the number of changes as a dependent variable. This gives rise to the rules of the form

— if software measure1 and software measure2 and ... and software measurep then number of changes

As we have four fuzzy sets of the number of changes, this naturally splits the set of associations into four subsets of rules; in this scenario some of the rules become inconsistent and have to be excluded (or left intact as associations), see Table 3.

5. Reduction of Rulebases

The rules come as a direct transformation of the associations. Their number could be reduced through a merging process. For instance the rules

— if A1 and B1 then C1
— if A2 and B1 then C1

can be merged into a single, more general rule in which one of the conditions is a disjunction of two information granules (assuming that A1 and A2 are two adjacent fuzzy sets)

— if (A1 or A2) and B1 then C1

The generalization of this nature reduces the number of all rules and could be automated to some extent. The crux of the optimization approach dwells on a well-known Quine-McCluskey reduction scheme commonly encountered in digital
Table 3. Rules constructed on the basis of the association; the highlighted rows indicate two inconsistent rules.

<table>
<thead>
<tr>
<th>LOC</th>
<th>Cl</th>
<th>Customer Level</th>
<th>Maturity</th>
<th>Project</th>
<th>Help</th>
<th>MM</th>
<th>NF</th>
<th>V(0)</th>
<th>BW</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 11. The representation of the rules in the K-map.

Medium, medium-large) are made adjacent on the Karnaugh map (K-map), say $A_1 = 00, A_2 = 01, A_3 = 11$, and $A_4 = 10$. Using the above coding scheme, the original rules occupy single entries in the K-map as shown in Fig. 11.

Apparently, some rules are adjacent and when combined they form prime implicants (note that each rule is just an implicant). The Quine-McCluskey method determines all prime implicants and thus helps us carry out the simplification (generalization) of the rules in an automatic fashion. Following Fig. 11, the prime implicants lead to the reduced rules

- if $A_2$ and $(B_2$ or $B_3$) then $C_1$
- if $A_3$ and $(B_3$ or $B_4$) then $C_1$

Some further generalization is still possible (even though the original simplification method does not cope with this phenomenon). We can merge the two rules by generalizing the first condition to the form

- if $A_2$ and $(B_2$ or $B_3$ or $B_4$) then $C_1$

One should stress that the above merging phase has to be handled separately and it is not supported by the Quine-McCluskey method.

Table 4 shows the number of associations for each fuzzy set of the number of changes. Following this scheme, the number of associations has been reduced to 34.

Table 5 shows a detailed insight into the prime implicants for the fuzzy set of high number of changes (that is, represented by the fourth fuzzy set).

Using this simplification method, the prime implicants $p_2$ and $p_3$ cannot be reduced any further. However, we can merge them manually producing the association of the form

$4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 4 \ 2 \ or \ 3 \ or \ 4$

Following the simplification method combined with further refinement, we obtain the rules summarized in Table 6.
Table 4. The reduction effect obtained through the use of Quinlan-McCluskey simplification method: the last column shows a number of implicants building a minimal coverage.

<table>
<thead>
<tr>
<th>Fuzzy set of the number of changes</th>
<th>Number of associations</th>
<th>Number of prime implicants</th>
<th>Minimal coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5. Enumeration of prime implicants for the fuzzy set of high number of changes.

<table>
<thead>
<tr>
<th>Prime implicant</th>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>p2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>p3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>p4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>p5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 6. The final rules after the automated and manual simplification.

<table>
<thead>
<tr>
<th>LOC</th>
<th>CL</th>
<th>Toler</th>
<th>Terms</th>
<th>Meta</th>
<th>Deter</th>
<th>N</th>
<th>Np</th>
<th>NF</th>
<th>VQG</th>
<th>BW</th>
<th>RC</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>29.84%</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>29.87%</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>29.07%</td>
</tr>
</tbody>
</table>

Table 7. Correlation coefficients for the software measures for the four fuzzy sets of the number of changes (a) small number of changes (first fuzzy set) - (b) high number of changes (fourth fuzzy set).

(a) LOC | CL | Toler | Terms | Meta | Deter | N | Np | NF | VQG | BW | RC | NC |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.93</td>
<td>0.94</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) LOC | CL | Toler | Terms | Meta | Deter | N | Np | NF | VQG | BW | RC | NC |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(c) LOC | CL | Toler | Terms | Meta | Deter | N | Np | NF | VQG | BW | RC | NC |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.89</td>
<td>0.89</td>
<td>0.89</td>
<td>0.89</td>
<td>0.89</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 7 (Cont'd)

<table>
<thead>
<tr>
<th>LOC</th>
<th>Cl</th>
<th>C</th>
<th>Total</th>
<th>Test</th>
<th>Model</th>
<th>Del.</th>
<th>N</th>
<th>Nv</th>
<th>VN</th>
<th>SW</th>
<th>NW</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.06</td>
<td>0.03</td>
<td>0.84</td>
<td>0.65</td>
<td>0.36</td>
<td>0.88</td>
<td>0.08</td>
<td>0.95</td>
<td>0.94</td>
<td>0.83</td>
<td>0.82</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>0.01</td>
<td>0.72</td>
<td>0.38</td>
<td>0.12</td>
<td>0.74</td>
<td>0.12</td>
<td>0.75</td>
<td>0.76</td>
<td>0.58</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td>0.09</td>
<td>0.06</td>
<td>0.79</td>
<td>0.41</td>
<td>0.19</td>
<td>0.82</td>
<td>0.19</td>
<td>0.82</td>
<td>0.82</td>
<td>0.65</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.03</td>
<td>0.83</td>
<td>0.64</td>
<td>0.35</td>
<td>0.87</td>
<td>0.07</td>
<td>0.93</td>
<td>0.94</td>
<td>0.82</td>
<td>0.82</td>
<td></td>
</tr>
<tr>
<td>0.06</td>
<td>0.04</td>
<td>0.77</td>
<td>0.42</td>
<td>0.20</td>
<td>0.81</td>
<td>0.20</td>
<td>0.81</td>
<td>0.81</td>
<td>0.64</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.03</td>
<td>0.83</td>
<td>0.65</td>
<td>0.35</td>
<td>0.86</td>
<td>0.15</td>
<td>0.91</td>
<td>0.92</td>
<td>0.79</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>0.06</td>
<td>0.04</td>
<td>0.77</td>
<td>0.42</td>
<td>0.20</td>
<td>0.81</td>
<td>0.20</td>
<td>0.81</td>
<td>0.81</td>
<td>0.64</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.03</td>
<td>0.83</td>
<td>0.65</td>
<td>0.35</td>
<td>0.86</td>
<td>0.15</td>
<td>0.91</td>
<td>0.92</td>
<td>0.79</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>0.06</td>
<td>0.04</td>
<td>0.77</td>
<td>0.42</td>
<td>0.20</td>
<td>0.81</td>
<td>0.20</td>
<td>0.81</td>
<td>0.81</td>
<td>0.64</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>0.05</td>
<td>0.03</td>
<td>0.83</td>
<td>0.65</td>
<td>0.35</td>
<td>0.86</td>
<td>0.15</td>
<td>0.91</td>
<td>0.92</td>
<td>0.79</td>
<td>0.79</td>
<td></td>
</tr>
</tbody>
</table>

The coverage of the rules included there sums up to over 100% which is not surprising as the fuzzy sets standing in the associations overlap. The same table of final rules is converted into a graphical representation, Fig. 12, which produces an overall visual "signature" of the rules and shows how the fuzzy sets are distributed across the rules.

Table 7 summarizes the correlation coefficients between the software measures and the number of changes. The strength of correlation varies from rule to rule. By analyzing these values we can gain a better view at the strength of dependencies between some variables and the "stability" of such relationships. For example, LOC is strongly coupled with BW for the second fuzzy set of the number of changes while this relationship is weak for the third fuzzy set of changes. At the same time the dependency between LOC and v(C) is constantly high and does not vary too much.

For each rule, we have designed "local" regression models expressing the number of changes as a function of the software measures (independent variables). The performance of these models is shown in Fig. 13. For comparison, Fig. 14 includes the results for a single (global) regression model.

6. Characterization of Software Measures and Their Granulation

The rule table helps us to evaluate software measures as to their discriminatory aspects, that is, their ability to discriminate between the categories of the number of changes made to the software module. With this regard, two quantitative descriptors are proposed.

6.1. Relevance of software measure

The intent of this index is to quantify how relevant the software measure is. Intuitively, if the measure appears less specific in the rule then its relevance is lower. Consider as an illustrative example the following three rules:

Fig. 12. Graphical visualization of the rules; the vectors (information granules) occurring in each rule are marked as black boxes.

Fig. 13. Performance of the local regression models for successive information granules of the number of changes (testing data set).
We count the number of values assumed by each measure in all these rules. Thus for measure-1, we get 4 values, for measure-2, 3 values and for measure-3, 6 values. Then the relevance order is measure-2, measure-1, and measure-3 (which exhibit the lowest relevance among these three).

6.2. Discriminatory abilities of software measures quantified through an entropy measure

We are interested in determining the discriminatory capabilities of the software measures. As each measure comes with a finite number of information granules, one can also attach local discriminatory abilities of each of such granules. Again, a simple example clarifies the point. The table of the rules is given below.

<table>
<thead>
<tr>
<th>Measure-1</th>
<th>Measure-2</th>
<th>Measure-3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1, 4</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3, 4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1, 3</td>
<td>1, 2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

The table is re-organized by focusing on the individual fuzzy sets of each software measure and counting how these fuzzy sets help discriminate between the classes. The arrangement of all these counts is shown below.

<table>
<thead>
<tr>
<th>Class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure-1</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>0.61</td>
</tr>
<tr>
<td>Fuzzy set 1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fuzzy set 2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Measure-2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Fuzzy set 3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Fuzzy set 4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Measure-3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1.5</td>
</tr>
<tr>
<td>Fuzzy set 5</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Fuzzy set 6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fuzzy set 7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Subsequently, for each fuzzy set we compute an entropy measure. For instance, for the first fuzzy set describing measure-1, we get

$$H = -3/\log_2(3/4) - 1/\log_2(1/4) = 0.81$$

If the given fuzzy set occurs for only one class it is very discriminatory. In this case the entropy function achieves zero. On the other hand, if the fuzzy set is attached to each class in a rather uniform manner, then the entropy takes on high values and this becomes a clear indicator of a lack of discriminatory power of the fuzzy set of the given measure.

The calculations of the entropy for each software measure leads to the results collected in Table 8. These results reveal an interesting pattern: most of the software measure exhibit similar entropy values (around 8) with an exception of the bandwidth measure (BW) whose entropy is far higher (648). This quantifies its quite limited discriminatory abilities. The entropy values for the remaining features are not low. This could suggest that each of the measures considered separately may not be sufficient to discriminate between the number of changes. Such abilities are also visible at the qualitative end by scanning the original rules. This finding is supported by the correlation analysis, see Table 9 where it is obvious that BW and the number of changes correlate quite poorly with the correlation coefficient being equal to 0.20.
Table 9. Correlation matrix of the software measures and the number of changes.

<table>
<thead>
<tr>
<th>LOC</th>
<th>CL</th>
<th>Tchar</th>
<th>Tforms</th>
<th>MehAR</th>
<th>Dchar</th>
<th>N</th>
<th>NF</th>
<th>V(G)</th>
<th>BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.99</td>
<td>0.98</td>
<td>0.88</td>
<td>0.96</td>
<td>0.94</td>
<td>0.93</td>
<td>0.83</td>
<td>0.88</td>
<td>0.77</td>
</tr>
<tr>
<td>2</td>
<td>0.96</td>
<td>0.96</td>
<td>0.83</td>
<td>0.93</td>
<td>0.93</td>
<td>0.91</td>
<td>0.83</td>
<td>0.83</td>
<td>0.77</td>
</tr>
<tr>
<td>3</td>
<td>0.89</td>
<td>0.89</td>
<td>0.91</td>
<td>0.82</td>
<td>0.81</td>
<td>0.81</td>
<td>0.78</td>
<td>0.81</td>
<td>0.75</td>
</tr>
<tr>
<td>4</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>5</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
</tr>
<tr>
<td>6</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>7</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>8</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>9</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>10</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>11</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
</tr>
<tr>
<td>12</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>13</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>14</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>15</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>16</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>17</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
</tr>
<tr>
<td>18</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>19</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>20</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>21</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
</tbody>
</table>

7. Conclusions

We have shown that the association analysis is a fundamental mechanism of data analysis especially for those phenomena that do not exhibit linearity. Software systems fall very much under this category. The agenda of essential (experimentally justifiable) associations has been constructed and then converted into a list of rules that is further compressed by adapting some well-known techniques of Boolean functions. The detailed analysis of rules with regard to their consistency has been also completed.

The framework and findings of this study are of direct interest to the pursuits of quantitative Software Engineering in several meaningful ways:

✓ the association analysis cast in terms of linguistic terms brings the results in the tangible and user-oriented fashion
✓ owing to their underlying linguistic character, the results are concise and easy to interpret
✓ we have quantified the relevance of the software measures with regard to their ability to describe software modules
✓ the distinction between associations and rules is essential to the better understanding of the nature of the software data. The rule-based modeling helps capture and describe some "local" characteristics of the software data. This feature is particularly important in light of the nonlinear characteristics of the data.

Acknowledgments

The support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Alberta Software Engineering Research Consortium (ASERC) is gratefully acknowledged.
References