An association rule mining method for estimating the impact of project management policies on software quality, development time and effort

María N. Moreno García, Isabel Ramos Román,
Francisco J. García Peñalvo, Miguel Toro Bonilla

Abstract

Accurate and early estimations are essential for effective decision making in software project management. Nowadays, classical estimation models are being replaced by data mining models due to their application simplicity and the rapid production of profitable results. In this work, a method for mining association rules that relate project attributes is proposed. It deals with the problem of discretizing continuous data in order to generate a manageable number of high confident association rules. The method was validated by applying it to data from a Software Project Simulator. The association model obtained allows us to estimate the influence of certain management policy factors on various software project attributes simultaneously.

Keywords: Association rules; Data mining; Software estimation; Project management; Simulation

1. Introduction

Software quality, project duration and development effort are important factors to be kept under control in the software development process. They are interrelated and influenced by many other factors which complicate the monitoring tasks. When managers have to take decisions about a project, they must consider a great number of variables and the complex relations between them. The simulation of software development projects by using dynamic models has contributed to a better knowledge of the influence of these variables and their relations. The *Software Project Simulators* (SPS), based on dynamic models, enable us to simulate the project's behavior and to evaluate the impact of different management policies and other factors. Nevertheless, they have important drawbacks: first, the number of input parameters needed for the simulation, and second, the number of possible combinations of factors influencing the development process that make it difficult to choose the best combination for the desired objectives. An important improvement in SPS is the treatment of the data generated by the simulator by using machine learning and evolutionary algorithms in order to facilitate their use (Aguilar-Ruiz, Ramos, Riquelme, & Toro, 2001; Ramos, Riquelme, & Aroba, 2001). The combination of factors for achieving specific objectives can be learned through the application of these supervised techniques. Such information allows managers to establish the correct management policy taking as reference the model generated by these algorithms.

When machine learning techniques are used, only one output variable is the target of the prediction. In classification problems this variable, named the class attribute, must be discrete. Classification algorithms use classified historical data for inducing a relation model between the class attribute and the other attributes (descriptive attributes).

* Corresponding author. Tel.: +34 923 294400.
E-mail address: mmg@usal.es (M.N. Moreno García).

Later, the model can make predictions about new, unclassified data.

The aim of this work is to present a method for estimating three variables simultaneously. We propose an association rule mining algorithm for building a model that relates management policy attributes to the output attributes quality, time and effort. All the available attributes are continuous, they must thus be split into intervals of values in order to generate the rules. The applicability and interest of the discovered associations depend mostly on how the data is discretized. The success of our method is mainly due to the supervised, multivariable procedure used for discretization. The result is an association model comprised of a manageable number of high confidence rules representing relevant patterns between project attributes. Those patterns provide managers with important information for decision making.

The rest of the paper is organized as follows: next section contains the fundamentals and main works concerning SPS, association rules and data discretization. Section 3 describes the experimental data provided by a dynamic simulation environment. The following section deals with the stage of data preprocessing. The proposed method for association rule mining and its results are presented in Section 5. The evaluation of the associative model obtained is given in Section 6 and, finally, we draw some conclusions.

2. Background

2.1. Software project simulators

The simulation of software project behavior by means of an SPS provides managers with a valuable tool for trying out several policies in order to take better decisions.

A dynamic model consists of a collection of parameters and functions needed for building the simulation environment. It is articulated in mathematical terms by means of a set of differential equations which express restrictions between variables that change over time. These restrictions enable us to analyze cause–effect relations among several project factors, such as management policies and technological, product and process factors. The analysis of the project with SPS can be done before the project start (a priori analysis), during the development (project monitoring) and when the project has finished (post-mortem analysis).

Since the publication of a dynamic model for software projects by Abdel-Hamid and Madnick in 1991 (Abdel-Hamid & Madnick, 1991), many other models and simulation environments for diverse application domains have appeared. In recent years the research in this field has increased, producing significant advances (Kellner, Madachy, & Raffo, 1999; Rodrigues & Williams, 1997). The most complete models are very complex and manage a large number of parameters which should be known previously. However, the numerous combination possibilities of such parameters make it difficult to find the best combination for a specific situation or purpose. These are the main drawbacks for the use of an SPS. In order to solve this problem, reduced dynamic models have been proposed for specific phases of the project (Ramos & Ruiz, 1998; Ruiz, Ramos, & Toro, 2001). Another alternative is the construction of data mining models from SPS data in order to analyze separately the influence of some of the factors related to the management policy on some of the project attributes. A number of data mining techniques, such as machine learning and evolutionary algorithms, can be used for building the models (Ramos et al., 2001) (Aguilar-Ruiz et al., 2001). The main drawback of a model generated by one of these supervised algorithms is that it can be used to estimate the repercussion of different management policies on just one project attribute. In this work we propose an algorithm for mining association rules that provide an associative model that will let managers know simultaneously the influence of policy factors on several project attributes.

2.2. Association rules

Data mining models can be obtained by employing supervised and unsupervised algorithms. Supervised methods require a learning stage for building a predictive model. The target of the prediction is a special attribute called the "label". The model is built from historical labeled data records by encoding the relation between the label and the other attributes; then, the model can be used for making predictions about new, unlabeled data. The two most common supervised modeling methods are classification and regression. If the label is discrete, it is named the class label and the task is called classification; if the label is continuous, the task is called regression. Unsupervised algorithms belong to knowledge discovery modeling. This task is descriptive instead of predictive and the objective is to detect patterns in present data without need of previous learning.

Traditionally, association analysis is considered an unsupervised technique, so it has been applied in knowledge discovery modeling. Recent studies have shown that knowledge discovery algorithms, such as association rule mining, can be successfully used for prediction in classification problems (Hu, Chen, & Tzeng, 2002; Li, Shen, & Topor, 2001; Moreno, García, & Polo, 2004; Wang & Wong, 2003). Patterns that have been extracted from historical data can serve to predict upcoming behaviours.

Since Agrawal et al. introduced the concept of association between items (Agrawal, Imielinski, & Swami, 1993a; Agrawal, Imielinski, & Swami, 1993b) and proposed the Apriori algorithm (Agrawal & Srikant, 1994), many other authors have studied better ways for obtaining association rules from transactional databases. Below, we introduce the foundations of association rules and some concepts used for quantifying the statistical significance and goodness of the generated rules (Padmanabhan & Tuzhilin, 2002).

Consider the set $D = \{T_1, T_2, \ldots T_N\}$ as a relation of N transactions over a set of discrete attributes $A_t = \{a_1, a_2, \ldots, a_m\}$. Let an atomic condition be a proposition of the form $\text{value}_1 \leqslant \text{attribute} \leqslant \text{value}_2$ for continuous attributes and attribute = value for discrete attributes, where value, value_1 and value_2 belong to the set of distinct values taken by attribute in D. Finally, an itemset is a conjunction of atomic conditions or items. The number of items in an itemset is called length. Rules are defined as associations of the form $X \rightarrow Y$, where X and Y are itemsets representing the antecedent and the consequent part of the rule, respectively. The strength of an association rule can be quantified by the following factors:

Confidence or *predictability*. A rule has confidence c if $c\%$ of the transactions in D that contain X also contain Y. A rule is said to hold on a dataset D if the confidence of the rule is greater than a user-specified threshold.

Support or *prevalence*. The rule has support s in D if $s\%$ of the transactions in D contain both X and Y.

Expected predictability. This is the frequency of occurrence of the item Y. So the difference between expected predictability and predictability (confidence) is a measure of the change in predictive power due to the presence of X (Mineset, 1998). Usually, the algorithms only provide rules with support and confidence greater than the threshold values established.

Association rule mining can be applied to obtain useful information from SPS databases. However, most of the algorithms for generating associations rules discover too many patterns, some of them contradictory or irrelevant, but only a reduced number of high confidence rules are valid for efficient decision making. Many methods for obtaining a manageable number of rules with high support and confidence values have been proposed; nevertheless, in many cases, an appropriate discretization of the continuous attributes is more effective than the use of complex association rule algorithms. We proposed a refinement method for obtaining stronger rules that has been successfully applied in the early software size estimation (Moreno, Miguel, García, & Polo, 2004). We used the refinement algorithm to process the SPS data with the purpose of obtaining strong association rules between several project management factors and attributes such as software quality, project duration and costs. The procedure yielded worse results than the simpler method proposed in this work based on a supervised multivariate discretization of the continuous attributes.

2.3. Discretization

The main application domain of association rules is decision support in market management, where they are used for finding products that clients tend to buy together. Attributes involved in these rules are nominal; however, in the project management field the available attributes are continuous. This is a drawback in the use of many data mining techniques that work with categorical and discrete attributes. Therefore, the application of those techniques requires a previous process of discretization consisting of splitting the continuous spectrum of values into intervals. In addition, the discretization of continuous attributes can provide many benefits: the predictive models induced by the algorithms are more accurate and more understandable; the induction process is faster and the analysis of the results can be more comprehensive and helpful (Liu, Hussain, Tan, & Dash, 2002).

Among the great variety of existing discretization algorithms, two simple techniques commonly used are equal-width and equal-frequency, which consist of creating a specified number of intervals with the same size or with the same number of records, respectively. The purpose of the discretized data and the statistical characteristics of the sample to be treated should be kept in mind when an algorithm is selected.

In classification problems an appropriate discretization of the attributes can improve the predictive accuracy. Because of this, specific algorithms considering class information have been developed. The procedure is called *supervised discretization* as opposed to *unsupervised discretization*, which does not consider the classes to establish the intervals. On the other hand, discretization can be *univariate* or *multivariate*. Univariate discretization quantifies one continuous attribute at a time while multivariate discretization considers multiple attributes simultaneously. Most of the supervised methods carry out the univariate discretization considering each attribute separately.

In our study we could not apply traditional supervised discretization because we did not have a class but rather three output variables to estimate, which constitute the consequent part of the rules. Attribute discretization methods for mining association rules have been treated in the literature. Nearly everyone takes the support factor of the rules as the main feature for splitting the attribute values into intervals, that is, they consider the weight of the records in the interval in relation to the total number of records (Srikant & Agrawal, 1996). Recently, several partition methods based on the fuzzy set theory have been proposed (Hong, Kuo, & Chi, 1999). The mined rules are expressed in linguistic terms, which are more natural and understandable. In these works either both the antecedent or consequent parts of the rules are formed by a single item or the consequent part is not fixed. In our case the consequent part must be fixed and both consequent and antecedent parts are itemsets. Hence, a supervised multivariate discretization that considers all the attributes (itemset) of the consequent part of the rules are more suitable.

3. Experimental data description

The data used in this study come from a dynamic simulation environment developed by Ramos et al. (Ramos & Ruiz, 1998; Ruiz et al., 2001). This environment manages

data from real projects developed in local companies and simulates different scenarios. It works with more than 20 input parameters and more than 10 output variables. The number of records generated for this work was 300 and the variables used for the data mining study were those related to time restrictions, quality and technician hiring. The description of the data is given below:

Input variables:

ASIMDY: average delay in the adaptation of the new technicians in days;
HIREDY: average delay in the incorporation of the new technicians in days;
TRNSDY: average delay in the departure of the new technicians in days;
MXSCDX: maximum allowed percentage of delivery time with regard to the initially estimated time.

Output variables:

JBSZMD: effort in technicians-days;
SCHCDT: development time in days;
ANERPT: product quality in errors/tasks.

The aim of this work was to study the influence of the input variables related to the project management policy on the output variables related to the software product and the software process.

4. Data preprocessing

Data mining comprises statistical and visualization techniques that complement the core algorithms. In this work we take advantage of the capabilities of multidimensional visualization tools in order to do an exploratory analysis of the behaviour of the input attributes in relation to the output ones, quality, time and effort. Figs. 1–4 are multidimensional scatterplots, one for each input variable, showing this behavior. The three output variables are represented on the axes and the intervals of values of the input variables are represented by colors. These graphs allow us to recognize general patterns in data such as a strong correlation of the attributes ASIMDY and HIREDY with the three output variables; however, no patterns could be appreciated for the other input attributes, TRNSDY and MXSCDX. Therefore, these attributes were not used in the generation of the association rules.

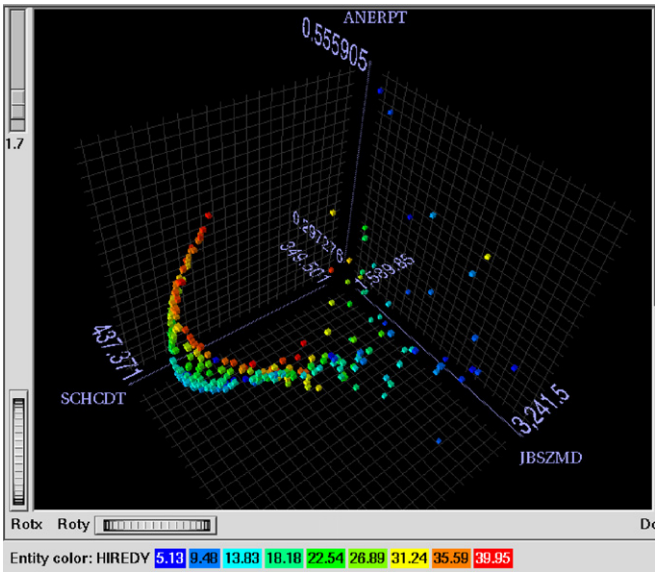

Fig. 2. Behavior of the input variable HIREDY in relation to the output variables.

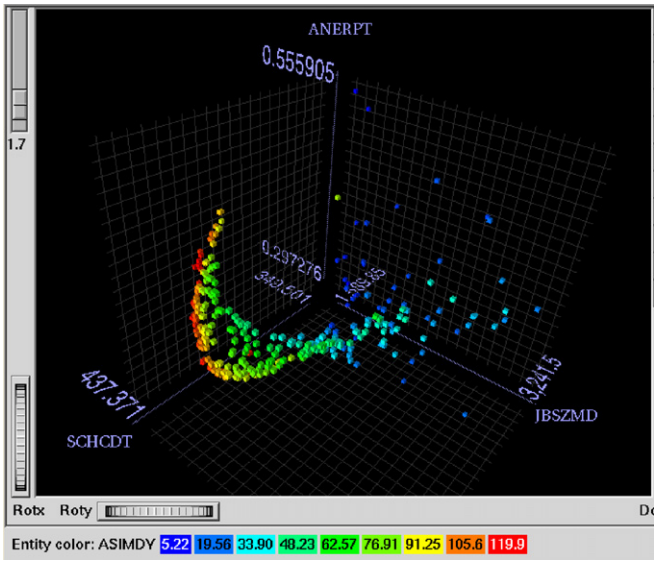

Fig. 1. Behavior of the input variable ASIMDY in relation to the output variables.

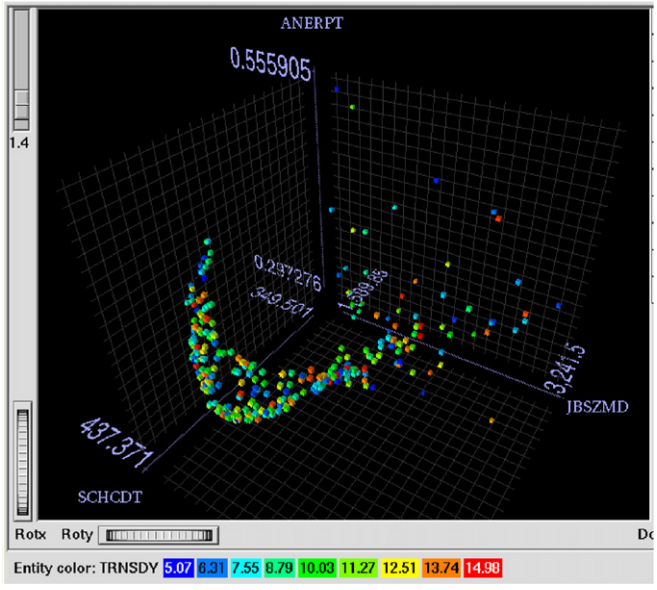

Fig. 3. Behavior of the input variable TRNSDY in relation to the output variables.

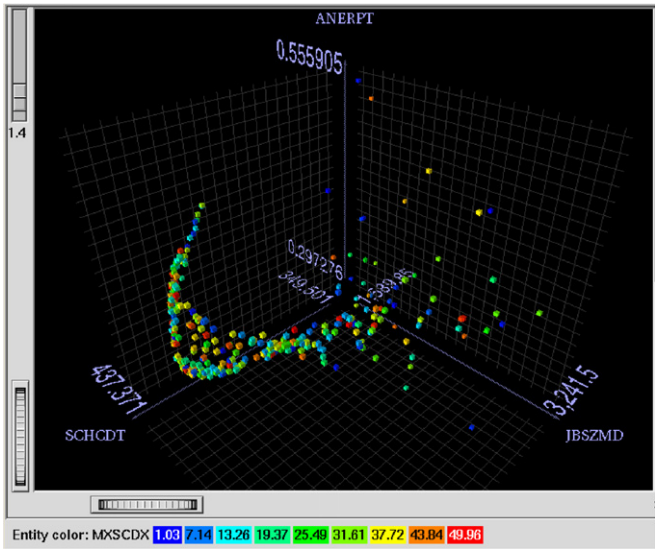

Fig. 4. Behavior of the input variable MXSCDX in relation to the output variables.

We also used representations similar to the one in Fig. 5 for detecting outliers in data, that is, records that lie at an abnormal distance from the sample. The detected outliers were eliminated before applying the algorithms for rule mining.

5. Method for mining association rules

The use of association rules is widespread in the business environment. They have been mostly adopted to target marketing or personalized recommendation services within the e-commerce area. In the project management field unsupervised data mining techniques such as association rules are used less frequently than machine learning techniques which have predictive purposes. However, both categories of methods can be used for extracting knowledge from historical information and applying it in future projects. The main inconvenience in this application domain is the type of data involved. While data in the business field are nominal, in the software project domain they are continuous and they must be discretized. The utility, interest and statistical strength of the rules obtained is highly conditioned by the suitability of the attributes' discretization. The association rule mining method applied in this work to the data from the simulation environment includes, as the main feature, an efficient discretization procedure.

5.1. Discretization procedure

All the attributes that are used in this work to generate association rules are continuous, that is, they can take a wide range of values. In order to reduce the number of rules generated it is necessary to discretize the attributes by splitting the range of values into a manageable number of intervals. Since the obtained association model is used with predictive purposes, we propose a supervised multivariate discretization procedure that contributes to increasing the predictive accuracy of the model by finding the intervals of values of these attributes that produce strong associations.

In order to discretize multiple attributes simultaneously, a clustering technique was applied. Clusters of similar records were built by using the iterative k-means algorithm with a Euclidean distance metric (Grabmeier & Rudolph, 2002). This distance $D(p,q)$ between two points p and q in a space of n dimensions is:

$$[D(p,q)]^2 = \|p-q\|^2 = \sum_{i=1}^{n}(p_i - q_i)^2 \quad (1)$$

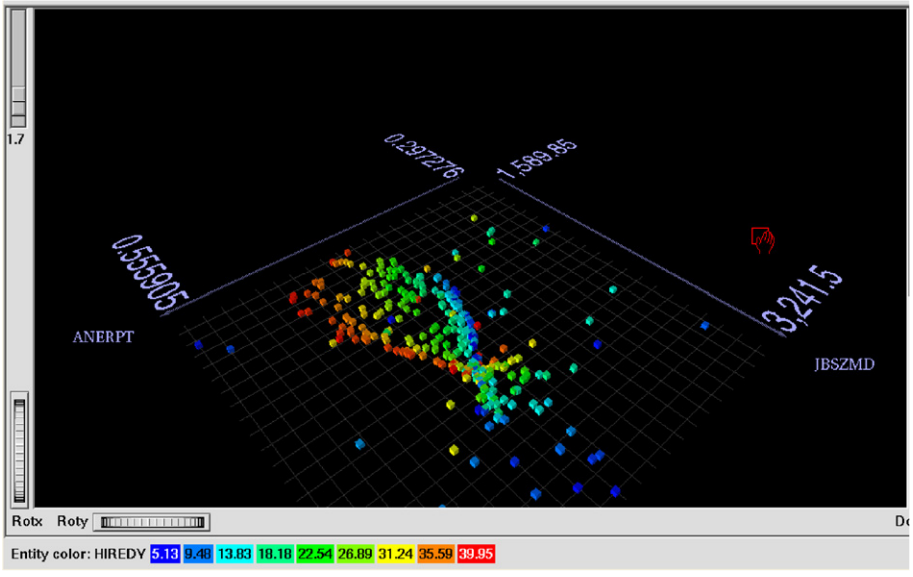

Fig. 5. Graph with outliers.

where p_i and q_i are the coordinates of the points p and q, respectively. In our case, the points are the records to be compared, and the coordinates are the n attributes of each record.

The iterative k-means algorithm takes as input the minimum and maximum number of clusters (k). The selected values in this work were 1 and 10, respectively. This clustering method groups the records in such a way that the overall dispersion within each cluster is minimized. The procedure is the following:

1. The value of the minimum number of clusters is assigned to k.
2. The k cluster centers are situated in random positions in the space of n dimensions.
3. Each record in the data is assigned to the cluster whose center is closest to it.
4. The cluster centers are recalculated based on the new data in each cluster.
5. If there are records which are closer to the center of a different cluster than the cluster that they belong to, then these records are moved to the closer cluster.

Steps 4 and 5 are repeated until no further improvement can be made or the maximum number of clusters is reached.

The clusters were created with a weight for the output variables three times greater than for input attributes. This is a supervised way of producing the most suitable clusters for the prediction of the output variables, which appear in the consequent part of the rules. In this study the clustering algorithm produced three clusters. The distribution of attribute values in the clusters was used for making the discretization according to the following procedure:

1. The number of intervals for each attribute is the same as the number of clusters. If m is the mean value of the attribute in the cluster and σ is the standard deviation, the initial interval boundaries are $(m - \sigma)$ and $(m + \sigma)$.
2. When two adjacent intervals overlap, the cut point (superior boundary of the first and inferior boundary of the next) is placed in the middle point of the overlapping region. These intervals are merged into a unique interval if one of them includes the mean value of the other or is very close to it.
3. When two adjacent intervals are separated, the cut point is placed in the middle point of the separation region.

This procedure was applied for creating intervals of values for every one of the attributes in order to generate the association rules.

5.2. Generating the association rules

Rules representing the impact of project management policies on software quality, development time and effort were generated and visualized by using Mineset, a Silicon Graphics tool (Mineset, 1998). Fig. 6 is a graphical representation of the rules on a grid landscape with left-hand side (LHS) items on one axis, and right-hand side (RHS) items on the other. A rule (LHS → RHS) displayed at the junction of its LHS and RHS itemset relates the itemset

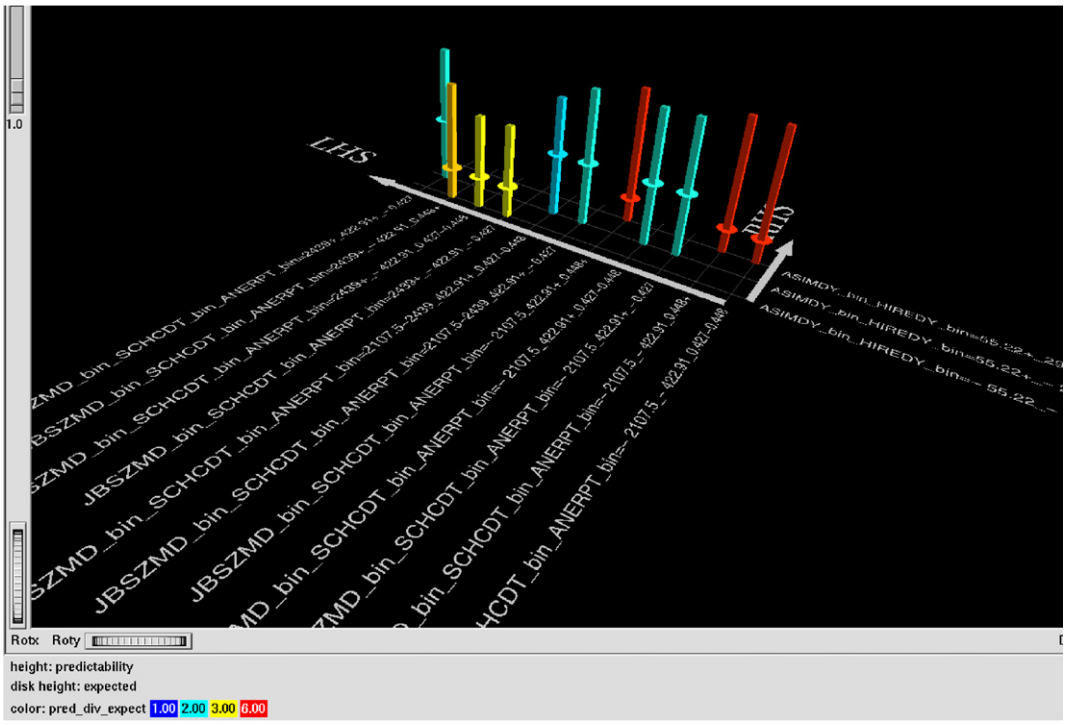

Fig. 6. Graphical representation of the association rules.

Table 1
Association rules and their support and confidence factors

Rule	ASIMDY	HIREDY	JSBZMD	SCHCDT	ANERPT	% Confidence	% Support
1	>55.22	<29.785	>2439	>422.91	<0.427	100	1.14
2	<55.22	<29.785	>2439	<422.91	>0.448	86.67	4.94
3	<55.22	<29.785	>2439	<422.91	0.427–0.448	69.77	11.41
4	<55.22	<29.785	>2439	<422.91	<0.427	69.23	3.42
5	>55.22	<29.785	2107.5–2439	<422.91	0.427–0.448	88.24	5.70
6	>55.22	<29.785	2107.5–2439	>422.91	<0.427	100	16.35
7	>55.22	>29.785	<2107.5	>422.91	>0.448	100	7.60
8	>55.22	<29.785	<2107.5	>422.91	0.427–0.448	100	7.60
9	>55.22	<29.785	<2107.5	>422.91	<0.427	100	10.27
10	>55.22	>29.785	<2107.5	<422.91	>0.448	100	4.18
11	>55.22	>29.785	<2107.5	<422.91	0.427–0.448	100	2.66
Sum						–	**75.27**
Average						**92.18**	–

containing the input attributes with the itemset formed for the output attributes. The display includes bars, disk and colors whose meaning is given in the graph. Fig. 6 shows association rules representing the influence of staff hiring factors (ASIMDY and HIREDY) on software quality (ANERPT), project duration (SCHCDT) and development effort (JBSZMD).

The rules generator does not report rules in which the predictability (confidence) is less than the expected predictability, that is, the result of dividing predictability by expected predictability (pred_div_expect) should be greater than one. Good rules are those with high values of pred_div_expect. We have also specified a minimum predictability threshold of 60%.

Under the conditions described, eleven rules were generated. These are presented in Table 1, which also includes their confidence and support factors.

6. Rules evaluation

The main drawbacks of the association rule algorithms are the generation of uninteresting patterns, the huge number of rules discovered and the low algorithm performance. These three problems can be avoided with an efficient discretization procedure such as the one proposed in this work.

The interestingness of a rule refers to finding rules that are interesting and useful to users (Liu, Hsu, Chen, & Ma, 2000). It can be assessed by means of objective measures such as support and confidence, which capture the statistical strength of a pattern. The more confident a rule is, the more reliable it will be when it is used to carry out predictions. Table 1 contains the confidence factor for the rules obtained. Seven of them have the maximum confidence value (100%) and the remaining rules have high values of this factor, yielding an average value of 92.18%. Therefore, they are good for taking decisions on future projects. On the other hand, the induced associative model is useful if it is comprised of a manageable number of rules and the rule set covers a large percentage of examples (records). The coverage measure is provided by the total support of the rules, that is, the sum of individual supports. In our case study the proposed method gives a model that covers the 75% of the examples with just eleven association rules (see Table 1).

In the study carried out, a reduced number of strong rules were generated. In addition, the rule induction process was very fast, because the association rule algorithm works with a reduced number of intervals of values of the attributes, which are provided by the discretization method. Thus, the associative model obtained, which relates management policy factors to quality, time and effort, provides managers with a useful tool for taking decisions about current or future projects.

7. Conclusions

Some management policy factors have a great influence on the success of a software project; however it is very difficult to know their impact on other project attributes, due to the complex relations existing between them. In this paper we have presented a data mining study of this influence by using data from an SPS based on a dynamic model. We have proposed an association rule mining algorithm for building a model that relates management policy attributes with the output attributes quality, time and effort. The success of the algorithm is mainly due to the supervised multivariate procedure used for discretizing the continuous attributes in order to generate the rules. The result is an association model comprised of a manageable number of high confidence rules representing relevant patterns between project attributes. This allows us to estimate the influence of the combination of some variables related to management policies on software quality, project duration and the development effort simultaneously. Classical machine learning methods can only predict one variable at a time.

The study has demonstrated that the delay in the departure of the new technicians and the maximum allowed percentage of delivery time with regard to the initially estimated time have no appreciable influence on the studied attribute projects. However, factors relating to the incorpo-

ration and adaptation of the new technicians have an important impact, as the associative model obtained shows.

In addition, the proposed method avoids three of the main drawbacks of rule mining algorithms: production of a high number of rules, discovery of uninteresting patterns and low performance.

References

Abdel-Hamid, T., & Madnick, S. (1991). *Software project dynamics: An integrated approach*. Englewood Cliffs, NJ: Prentice Hall.

Agrawal, R., Imielinski, T., & Swami, A. (1993a). Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering, 5*, 914–925.

Agrawal, R., Imielinski, T., & Swami, A. (1993b) Mining associations between sets of items in large databases. In *Proceedings of ACM SIGMOD international conference on management of data*, Washington, DC (pp. 207–216).

Agrawal, R., & Srikant, R. (1994) Fast algorithms for mining association rules in large databases. In *Proceedings of 20th international conference on very large databases*, Santiago de Chile (pp. 487–489).

Aguilar-Ruiz, J. S., Ramos, I., Riquelme, J. C., & Toro, M. (2001). An evolutionary approach to estimating software development projects. *Information and Software Technology, 43*, 875–882.

Grabmeier, J., & Rudolph, A. (2002). Techniques of cluster algorithms in data mining. *Data Mining and Knowledge Discovery, 6*, 303–360.

Hong, T. P., Kuo, C. S., & Chi, S. C. (1999). Mining association rules from quantitative data. *Intelligent Data Analysis*, 363–376.

Hu, Y. C., Chen, R. S., & Tzeng, G. H. (2002). Mining fuzzy associative rules for classifications problems. *Computers and Industrial Engineering, 43*, 735–750.

Kellner, M. I., Madachy, R. J., & Raffo, D. M. (1999). Software process simulation modeling: Why? What? How? *Journal of System and Software, 46*, 91–105.

Li, J., Shen, H., & Topor, R. (2001) Mining the smallest association rule set for predictions. In *Proceedings of IEEE international conference on data mining (ICDM'01)*.

Liu, B., Hsu, W., Chen, S., & Ma, Y. (2000). Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems* (September/October), 47–55.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: an enabling technique. *Data Mining and Knowledge Discovery, 6*, 393–423.

Mineset user's guide (1998). v. 007-3214-004, 5/98. Silicon Graphics.

Moreno, M. N., García, F. J., & Polo, M. J. (2004). Mining interesting association rules for prediction in the software project management area. *Lectures Notes in Computer Science, 3181*, 341–350.

Moreno, M. N., Miguel, L. A., García, F. J., & Polo, M. J. (2004). Building knowledge discovery-driven models for decision support in project management. *Decision Support Systems, 38*, 305–317.

Padmanabhan, B., & Tuzhilin, A. (2002). Unexpectedness as a measure of interestingness in knowledge discovery. *Decision Support Systems, 33*, 309–321.

Ramos, I., Riquelme, J., & Aroba, J.C. (2001). Improvements in the decision making in software projects: Application of data mining techniques. IC-AI'2001.

Ramos, I., & Ruiz, M. (1998). A reduced dynamic model to make estimations in the initial stages of a software development project. In C. Hawkings et al. (Eds.), *INSPIRE III. Process improvement through training and education* (pp. 172–185). London: British Computer Society.

Rodrigues, A. G., & Williams, T. M. (1997). System dynamics in software project management: towards the development of a formal integrated approach. *European Journal of Information System, 6*, 51–56.

Ruiz, M., Ramos, I., & Toro, M. (2001). A simplified model of software project dynamics. *The Journal of Systems and Software, 59*, 299–309.

Srikant, R., & Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In *Proceedings of ACM SIGMOD conference* (pp. 1–12).

Wang, Y., & Wong, A. K. C. (2003). From association to classification: inference using weight of evidence. *IEEE Transactions on Knowledge and Data Engineering, 15*, 764–767.