LOGIC BASED RULES FOR PROJECT ASSIGN PREDICTION

Ashu Singla*, Raman Maini*

*University College of Engineering, Punjabi University Patiala, Punjab, India – 147002
ashsinglaoct@gmail.com

Abstract: Project assigning is a major decision for any software organization like IT companies to their developer employees. The uncertain domain of assessment has been in need of a reliable and consistent system to help simplify the decision making process. Logic rules provides a completely different, unorthodox way to approach a control problem. This method focuses on what the system should do rather than trying to understand how it works. One can concentrate on solving the problem rather than trying to model the system mathematically, if that is even possible. This almost invariably leads to quicker, cheaper solutions. This research work will help in analysis assign project to a developer according to different factors like reliability, his hardware interaction, programming level, configuration etc the decision factor helps in making decision above how to select accurate developer for the project. Using the algorithm has been defined, new rules defined, according to different rules added and take average decision based on data collected by different categories of the employees.

Keywords: Logic rules, Decision support system, Visual basic, Iterative Dichotomiser

INTRODUCTION

It is very difficult to evaluate factors for any organization to assign the projects. The main problem of the research work is that to evaluate different factors about a new employee (developer). So in very short time it is difficult to analyze to whom we will assign the project or whom not. As human brain is capable of analyzing few factors regarding new employee, because every case is new or different even for average experienced person. The question though is how to resolve the pressure from human perception in making a judgment. Data mining is making changes to the entire make up of our skills and comfort zones in information analysis [1]. Data mining is a new type of exploratory and predictive data analysis whose purpose is to establish systematic relations between variable when there are no prior expectations as per to nature of those relations. Data mining is a process of extracting previously unknown, valid, potentially useful and hidden patterns from large datasets. As the amount of data stored in company databases is increasing rapidly, in order to get required benefits from large data sets and to find hidden relationships between variables using different data mining techniques developed and used. Clustering and decision tree are most widely used techniques for future predictions. Decision trees are analytical tools used to discover rules and relationships by systematically breaking down and sub dividing the information contained in data set. Decision trees are composed of a hierarchy of “if-then” statements in which the synchronous Tree construction approach and the and the partitioned tree construction approach [6]. Traditionally approach attempts to develop analysis system based on the logic using two structures: traditional reasoning of all inputs that map to one single output and stage wise reasoning of input parameters in accordance with their importance. These models are advanced but sometimes complex and can only be understood by specialists [8][9]. The expert rules were constructed using the reasoning in order to adequately analysed the inputs. This paper investigates way of logic implementation for a project assigns predictor. It’s a basically decision support system which helps in decision-making regarding a new project assignment to the developer. The approach considers all risk influencing input parameters in single stage of the decision making process. Member functions have been plot according to the different input variables and their rules are defined. From different rules added, decision has been taken according to decision as accepted or rejected. For accurate results studies are presented.

The rest of this paper is organized as follows. Section 2 deals with the overview of the research work, section 3 and 4 describes about the small introduction of decision trees and ID3 algorithm. Section 5 is having the results and the last section concludes the finding of the proposed research work and future or scope of the research work.

OVERVIEW OF THIS WORK

In company’s database there are information about each developer like their name, their programming level, their compatibility with the configuration, how much a person can be reliable to a company, and he or she can interact with the hardware or not, the language in which they can perform well and many more. The proposed model makes prediction about when a new developer is recruited then which project is assigned to him or her. It is preliminary that the assigned new project will be depend on the qualities as attributes which he
has but the previous data will help to take a decision in which area he or she should be assigned.

Following the methodology shown the stages involved in the process of this work. Mainly the input data is taken and then in the required format as the software needs then applied the ID3 algorithm. Further ID3 takes many steps and then the output will be the logic rules, which will be the result.

![Decision Tree Diagram](image)

**ID3 ALGORITHM BASIC**

ID3 is a simple decision tree-learning algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node [3][4]. A measure called information gain, which will be used to decide which attribute to test at each node. Information gain is itself calculated using a measure called entropy, which we first define for the case of a binary decision problem and then define for the general case.

Given a binary categorization, C, and a set of examples, S, for which the proportion of examples categorized as positive by C is P(positive), and the proportion of examples categorized as negative by C is P(negative), then the entropy of S is:

\[
\text{Entropy}(S) = -P(\text{positive}) \log_2 P(\text{positive}) - P(\text{negative}) \log_2 P(\text{negative})
\]

\[(i)\]

P(positive): proportion of positive examples in S
P(negative): proportion of negative examples in S

ID3 ALGORITHM BASIC

ID3 is a simple decision tree-learning algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node [3][4]. A measure called information gain, which will be used to decide which attribute to test at each node. Information gain is itself calculated using a measure called entropy, which we first define for the case of a binary decision problem and then define for the general case.

Given a binary categorization, C, and a set of examples, S, for which the proportion of examples categorized as positive by C is \(P(\text{positive})\), and the proportion of examples categorized as negative by C is \(P(\text{negative})\), then the entropy of S is:

\[
\text{Entropy}(S) = -P(\text{positive}) \log_2 P(\text{positive}) - P(\text{negative}) \log_2 P(\text{negative})
\]

\[(i)\]

P(positive): proportion of positive examples in S
P(negative): proportion of negative examples in S

Note that the more uniform is the probability distribution, the greater is its information. You may notice that entropy is a measure of the impurity in a collection of training sets.

Information Gain: We now return to the problem of trying to determine the best attribute to choose for a particular node in a tree[10]. The following measure calculates a numerical value for a given attribute, A, with respect to a set of examples, S. Note that the values of attribute A will range over a set of possibilities which we call \(\text{Values}(A)\), and that, for a particular value from that set, v, we write \(S_v\) for the set of examples which have value v for attribute A.

The information gain of attribute A, relative to a collection of examples, S, is calculated as:

\[
\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v=1}^{n} \left( \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \right)
\]

\[(ii)\]

The algorithm terminates either when all the attributes have been exhausted, or the decision tree perfectly classifies the problem. It uses information gain to measure the attribute to put in each node, and performs a greedy search using this measure of worth. The algorithm goes as follows:

Given a set of examples, \(S\), categorized in categories \(c_i\), then:

1. Choose the root node to be the attribute, A, which scores the highest for information gain relative to S.
2. For each value v that A can possibly take, draw a branch from the node.
3. For each branch from A corresponding to value v, calculate \(S_v\). Then:
   - If \(S_v\) is empty, choose the category \(c_{\text{default}}\), which contains the most examples from \(S\), and put this as the leaf node category which ends that branch.
   - If \(S_v\) contains only examples from a category \(c\), then put \(c\) as the leaf node category, which ends that branch.
Otherwise, remove A from the set of attributes which can be put into nodes. Then put a new node in the decision tree, where the new attribute being tested in the node is the one which scores highest for information gain relative to $S_v$. This new node starts the cycle again, with $S$ replaced by $S_v$ in the calculations and the tree gets built iteratively like this.

**DESIGN OF THE PREDICTOR TOOL AND RESULTS**

In the process of designing a predictor the most important task is to identify those factors that contribute primarily to a software company’s decision concerning providing projects to a right person for developing. In order to identify the process and influencing the factors those contributes to a developer’s assessments; the analysis work of an experienced manager in a software company was observed. After discussion we identified the main factors are programming level, configuration, reliability, hardware interaction, computer language etc are important factors on which theis decision can be based.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming level</td>
<td>Terran, Protoss, Zerg</td>
</tr>
<tr>
<td>Configuration</td>
<td>Good, medium, low</td>
</tr>
<tr>
<td>Reliability</td>
<td>high, normal</td>
</tr>
<tr>
<td>Hardware interaction</td>
<td>true, false</td>
</tr>
<tr>
<td>language choosen</td>
<td>offline, online</td>
</tr>
</tbody>
</table>

Table2: Parameters chosen and relative inputs

Here the technical terms are taken by the industry for the programming level such as Terran for the technical leader with exp of 10-15 years, Zerg as on particular platform or any particular tool like oracle, mysql and Protoss use algo’s and mathematical tools. The impact levels of various parameters can be set or change according to the system of the different organizations [11][12].

The GUI based tool is developed according to different rules added, which is developed using Visual Basic language as shown in fig 2 and corresponding decision by this tool in fig 3.

**CONCLUSION**

A logic-based predictor is developed to assist Software Company in decision-making. Decision tree learning algorithm has been successfully used in expert systems in capturing knowledge. The main task performed in these systems is using inductive methods to the given values of attributes of an unknown object to determine appropriate classification according to decision tree rules. A simple GUI based applicatin, which is developed using visual basic, is simple and easy to use. The main focus of this depend upon the impact factors of attributes, it is very clear to analysis decision whether a project is assign or not for this application. From this research paper, another students can take the idea about the application of data mining in new research areas.

**ACKNOWLEDGMENT**

I would like to thank my Research Guide Raman Maini, Associate Professor, University College of Engineering, Punjabi University Patiala for his valuable assistance, help and guidance during the research process.
REFERENCES

[6] Elzbieta K.Trybus, Ginter Trybus . A Brief introduction to the field of data mining.CA91330 USA

SHORT BIODATA OF ALL THE AUTHOR

Raman Maini, received B.Tech(Computer Science & Engineering) from Beant College of Engineering, Gurdaspur, Punjab, India in 1999 and M.Tech( Computer Science & Engineering) from PAU, Ludhiana, India , in 2002. He got Merit certificate in his M.Tech thesis at PAU He is currently working as an Assistant Professor in Computer Engineering at University College of Engineering, Punjabi University, Patiala, India. He is a life member of ISTE (Indian Society of Technical Education), India, IETE (Institution of Electronics & Telecommunication Engineers), India. His current area of research is Computer Vision (Specialty Noise Reduction in Medical Images, Edge Detection and Image Enhancement.

Ashu Singla, B.Tech(Computer Science & Engineering) from RIMT Engineering college, Mandi Gobindgarh, Punjab, India in 2009 and currently in M.Tech( Computer Engineering) from at University College of Engineering, Punjabi University, Patiala, India.