Journal of King Saud University – Computer and Information Sciences (2017) **29**, 93–102

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Fuzzy inferencing to identify degree of interaction in the development of fault prediction models

Rinkaj Goyal *, **Pravin Chandra**, **Yogesh Singh**

University School of Information and Communication Technology, Guru Gobind Singh Indraprastha University, Sector 16 C, Dwarka, Delhi 78, India

Received 8 October 2014; revised 24 December 2014; accepted 24 December 2014
Available online 3 November 2015

KEYWORDS

Software fault prediction;
Fuzzy inference system;
Influential metrics;
Object oriented metrics

Abstract The software fault prediction models, based on different modeling techniques have been extensively researched to improve software quality for the last three decades. Out of the analytical techniques used by the researchers, fuzzy modeling and its variants are bringing out a major share of the attention of research communities. In this work, we demonstrate the models developed through data driven fuzzy inference system. A comprehensive set of rules induced by such an inference system, followed by a simplification process provides deeper insight into the linguistically identified level of interaction. This work makes use of a publicly available data repository for four software modules, advocating the consideration of compound effects in the model development, especially in the area of software measurement.

One related objective is the identification of influential metrics in the development of fault prediction models. A fuzzy rule intrinsically represents a form of interaction between fuzzified inputs. Analysis of these rules establishes that Low and NOT (High) level of inheritance based metrics significantly contributes to the F-measure estimate of the model. Further, the Lack of Cohesion of Methods (LCOM) metric was found insignificant in this empirical study.

1. Introduction

Fault-proneness of a software module is an external metric and a fault prediction model applies a modeling technique to the internal software metrics to predict fault-proneness (Catal and Diri, 2007; Fenton and Neil, 1999). Due to the increased practice of object oriented technology in industry, an extensive usage of object oriented metrics has been proposed, and efforts have concentrated on building models that predict defective modules (Arisholm et al., 2010). These metrics not only indicate the complexity of an object and its association (interaction) with other objects, but also measure different characteristics of a quality model. The widely used Chidamber and Kemerer (CK) metrics along with a metric, Lines of Code (LOC) are used in this paper to conduct empirical study.

New fault prediction models may be developed from statistically validated improved models reported earlier or one may

* Corresponding author. Tel.: +91 8826020315.
E-mail addresses: rinkajgoyal@gmail.com (R. Goyal), chandra.pravin@gmail.com (P. Chandra), ys66@rediffmail.com (Y. Singh).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.12.008
1319-1578

develop their own customized models. The growth of such customized models reflects the necessities of a software project and applies the expertise of people affected in the operation (Weyuker et al., 2007).

Fuzzy based models translate the subjective understanding and expertise of the processes into mathematically exposable figures and rules to generate systems with some degree of uncertainty. The use of fuzzy logic in experimental software engineering to model various aspects of software evolution process is increasingly attaining recognition of the research community (Ahmed and Muzaffar, 2009; Aljahdali and Sheta, 2011; Bouktif et al., 2010; Chiu, 2011; Engel and Last, 2007; Gray and MacDonell, 1997; Khoshgoftaar and Seliya, 2003; MacDonell, 2003; Meneely et al., 2008; Ozcan et al., 2009; Pandey and Goyal, 2009; So et al., 2002; Verma and Sharma, 2010). The present study discusses the use of fuzzy inference mechanism to recognize the most notable rules in the development of fault prediction model using GUAJE framework (Alonso and Magdalena, 2011a,b; Alonso et al., 2012).

This study makes use of the data set of four software modules available in the NASA data repository for experimental usage (Jureczko and Madeyski, 2010).

Though in this paper, we derive knowledge from data itself. Nevertheless, fuzzy inference mechanism permits adding of experts experience in the formulation of fuzzy rules.

In this paper, interaction between variables is not conceptualized as a multiplicative term (generally used in regression analysis), rather the type of interaction analyzed here is moderated by FIS operators. The rest of this paper is organized as follows: Section 2 presents the core ingredients of a fuzzy inferencing system and semantics of a typical fingram; Section 3 describes the data set and FIS simulation parameters;Section 4 presents the results derived, and further elaborates the accuracy measures the developed FIS. Our conclusion is presented in Section 5.

2. Fuzzy inference system and semantics of a fingram

The core elements of a data driven fuzzy inference systems are:

1. Fuzzification module: alters crisp inputs into fuzzy values using membership function. In the construction of data driven fuzzy inferencing system, fuzzy partitioning of crisp values of the variables of data set carries out this fuzzification process.
2. Fuzzy inference systems (FIS): accomplish input–output mapping of linguistically communicated information in the form of rules. Different potential building algorithms induce these rules by learning from data.
3. Simplification module: holds the number of rules small and keeps consistent rule base. Though this module is optional in fuzzy inferencing process, GUAJE framework provides improved readability of generated fuzzy rules using this module.
4. Defuzzification: transcribes fuzzy outputs back into crisp values.

Fig. 1 elucidates the core elements of data driven fuzzy inference system discussed above (Guillaume and Charnomordic, 2011). Admitting that the expert knowledge

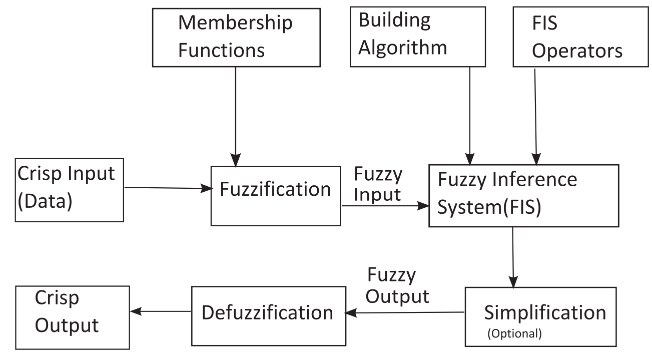

Figure 1 Core elements of a fuzzy inference system.

is not accessible, the diagram does not explicitly confer the feedback perspective from the user and rule induction takes place through available information simply.

The simulation environment utilized in this study creates a number of fuzzy partitions of various sizes exercising three mechanisms namely regular partition, k-means algorithm with different numbers of groups and hierarchical fuzzy partitioning (hfp). Incorporation of criteria like partition coefficient (PC), partition entropy (PE) and Chen index (CI) determines suitable partition. A reliable partition minimizes the entropy and maximizes the partition coefficient and the Chen index (Guillaume and Charnomordic, 2011).

To induce rules from the fuzzified input data, the experimentation puts to use the Fast Prototyping Algorithm (FPA). FPA generates rules based on the basis of the rule matching degree and the number of records in the data set.

The patterns generated by FPA are quite large in number. Thus, simplification module protects the strongest interactions and rules out other variables, which are seeming in some rules only. Derived FIS uses normalized entropy as a criterion for classification problems to hold the balance between precision and interpretability.

Results obtained from the aforementioned fuzzy inference mechanism make use of fingrams for visual analysis of strong interactions and other useful outcomes.

In fingram, a circular node enacts a rule with its size as the comparative magnitude of the covered data samples. Classified colors of circles describe different classes of data set. For example, in Fig. 2. There are two classes in the data set depicted by the legend in Fig. 2.

Classes 1.0 and 2.0 represent non-faulty and faulty data sample, respectively, and same terminology and legend is used in the interpretation of all fingrams constructed in this paper.

Each circle represents the values of cov, G, Ci which are explained in as follows (Pancho et al., 2013a):

1. *Coverage of a node (cov)*: Ratio of covered data samples to the total number of samples (Relevant in interpreting fingram).
2. *Goodness (G)*: Indicates the goodness of a rule to classify data samples. Its value varies from −1 to 1. A negative value signifies the lower number of data samples correctly classified.
3. *Relative coverage of a node (Ci)*: This refers to the coverage of a rule corresponding to an output class. It is the proportion of the number of data samples covered by that rule to the total number of data samples in that class.