DIGGING HIGH RISK DEFECTS OUT IN SOFTWARE ENGINEERING

Jin-Cherng Lin, Kuo-Chiang Wu

Abstract: Data mining implies "digging through tons of data" to uncover implicit information. Software defect is an essential characteristic of software development process, offering much information about software quality assurance. Based on the information supplied by defects, it can help a software team with capability of software quality assurance. In order to dig through lots of defects to uncover implicit information of defects, this paper integrates PCA and DA, combined with the relationship among each defect attribute index, and then we can dig critical factor out of software defects. These critical factors are the vital few defects affecting software quality, warning programmer to stress concentration on getting rid of these vital few defects.

Key words: Prime Components Analysis (PCA) and Discriminate Analysis (DA)

1. INTRODUCTION

The earlier defect is discovered in software, the lower the developing cost is. The higher the software quality is, the lower the maintenance cost after software released is.

Defect is the byproduct in software development process. Usually, defect may result in software products not satisfying requirement of user. Defect indicates error existed in program, for example, syntax error, spelling error or wrong program statement. Defect may be also errors existed in design, even in requirement and specification or other documentations. However, software defect is an elemental feature of software development process, offering much information about software quality. Based on the information supplied by defects, it can help a team with capability of SQA (Software

Quality Assurance). SQA is an important strategy for safeguarding the design, production and support of software. It is imperative that all defects to be dug out for SQA.

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data Mining can be defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [1] and "The science of extracting useful information from large data sets or databases" [2]. Data mining, also referred to Knowledge Discovery in Database (KDD), is a pivotal step involving the extraction of interesting patterns (such as knowledge rules, constraints, and regularities) from a set of data sources. The patterns obtained are used to describe concepts, analyze associations, build classification and regression models, cluster data, model trends in time-series, and detect outliers. Likewise, the goal of this paper is to construct defects mining for software quality to analysis all defects, indicating the critical factors that affect the software quality.

The software development process is complicated and uncertain. Of factors all, the most are surrounded by defect problems, and therefore it is necessary to collect data of defect and have statistical analysis, especially in defect mining. This paper is trying to solve issues relating to the defect mining for software quality. This paper employs PCA and DA as techniques of defects mining in order to dig out vital few defects affecting software quality assurance, warning programmer to stress concentration on getting rid of these vital few defects.

2. EXISTENCE OF DEFECTS DEPENDENCY IN COLLECTING DATA

If testers do not care about accuracy at classifying defects, programmer will not precisely find location of defects. Consequently, in software testing process, tester will classify found defects so as to effectively manage defects.

Perhaps testers classify defects according to their subjective judgment, which causes dependency among different defect attributes. In other words, testers sometimes classify a certain kind of defects into different defect attributes. A certain kind of defects is classed as class A, but the same kind defect is classed as class B in another instances. The reason why they mistake is tester forgets or mistakes the judging rule of classifying in different instance; that is to say, tester is busy at working in searching and classifying defects, so it is unavoidable for tester sometime cannot see where

they went wrong. In short, too many classifying rules of defect attributes will bring about many mistaking chance. Consequently, defects classifying is too complex (many) to easily manage them, but too rough (few) to precisely locate them.

Another problem is that different defect attributes are caused by the same error and different defect attributes have the same defect frequency. This puzzling situation is like a ripple effect; for instance, defect in called subprogram introduced new defects to calling program. Besides, many other conditions also cause ripple effect Maybe only one source defect exists in source program, but they introduce a lot of new born defects. Repeated calculation of defect frequency will cause redundant information, while redundant information will cause errors in subsequent statistical analysis.

Dependency and redundant information should be overcome at once. PCA is a multivariate statistical analysis, which can reduce dependency (or collinear) and redundant information among different defect attributes (refer to [3]). Furthermore, PCA is able to transform many indexes to few independent comprehensive indexes. The nature of PCA is to make the high dimension system best integrated and to avoid objectively determining weight of each index (refer to [3]). Considering many factors affecting defect attribute, PCA is a relatively feasible method, especially for handling data dependency.

DA is also a multivariate statistical analysis, which a classified and predicate original data (refer to [4-6]). DA is classified and predicated by discriminate function, and the discriminate coefficient of discriminate function reflects the importance of the factor in discriminate function (refer to [4-6]). For example, the discriminate coefficient of erroneous logic is much greater than other factors, which is the major factor to evaluate the whole software quality. The discriminate coefficient of erroneous specification is much smaller than other factors, and no consideration can be made in evaluating the whole software quality. We can effectively find vital few defect attributes by discriminate coefficient of DA.

3. CASE STUDY OF DEFECT MINING

This case is a plan for defects management, whose goal is to design an approach, indicating the key factor affecting software quality. Vital few defects are the key factor of software quality; therefore concentration should be stressed on vital few defects. First we will study how to standardize the gathered defect data and find out five prime components of defect through

PCA. Then establish discriminate function by five prime components of defect differentiates the importance of each defect attribute, and then vital few defects attributes can be found through discriminate function. Furthermore, vital few components were found by helps of vital few defects attributes. In vital few components, we will dig higher risk class out in OO environment (see section 4).

In the first column of table 1 is an acronym of defects attributes, which will be explained in table 2. In other words, there is a correspondence of abbreviation between table 1 and table 2. The other columns are the value of standardized original data (see section 3.1).

3.1 Standardized original data

In Table 1, the original data is obtained by number of defect density (defects/KLOC), complexity, or other method. A method for collection of data was determined by tester. In other words, different measuring methods will apply different dimensions, so we should pay attention to eliminating the effects of different dimensions. Therefore, we need to standardize original data. This case achieved normalization of original data by **Z-score**, which has an expression like:

$$Z=\frac{X-\bar{X}}{\hat{\alpha}}$$

Where X =original data $\bar{X}$ =mean value $\hat{\alpha}$ =standard deviation. Before the following PCA and DA treatment, original data should be calculated by Z-score first (see Table 1).

3.2 Prime Component Analysis of Defect Attributes

Originally there are 43 defect attributes in this case. However, some defect attributes rarely happen, so they are not listed in statistical analysis, with only 29 defect attributes listed in statistical analysis (see Table 1).

Then, based on correlation matrix of evaluation index, PCA can be conducted after VARIMAX rotation. The communalities of each index in defect type got from factor loading matrix indicate the total variance contribution that each index has made to defect attribute, and from this we can get index weights. Calculating the percentage of communalities in total communalities and then transforming the weight to number value between 0 and 1, we get the index weight.

PCA for the defect attribute in this project demonstrates that eigenvalues of five primary components are greater than 1, and is maintained for its explanation to most of total variances (75%)(see Table 2). From communalities in each defect attributes, we can see five primary components can explain the variance of LE, IOLE, MLT, BTSUI, DIDI, LSI and VIFNP, whose variance is greater than 90%; it can explain the variance of BTI, LPO, MCT, LAOS, WAOP and ID, whose variance is greater than 80%; it can explain the variance of IVT, VRWN, LPWI and DIND, whose variance is greater than 70%; it can explain the variance of DCII, DHE, ODW, MSL, VEWL, CPWS and ASD, whose variance is greater than 60%; while the variance explanation of five prime component FSWDW, IPS, AD, RLWD and WVBC is less than 60%. It is thus clear that five prime component can explain the variance of most defect attribute index.

Prime components arranged in order according to eigenvalues. The first prime component (PCA1), together with LE, IOLE, LAOS, LSI, MLT, MCT, BTI, BTSUI, DIDI, VIFNP, IVT,VRWN, LPWIS, DHE,DIDN,CPWS and WVBC, has relatively high factor loading (greater than 0.6), moreover, these indexes have obvious relevance with erroneous logic. Therefore, we call the first prime component erroneous logic factor. The second prime component (PCA2), together with IPS and MSL, has relatively high positive loading, and it has negative loading with DCII. All these indexes demonstrates erroneous specifications index, therefore, we call the second prime component (PCA2) as erroneous specifications factor. The third prime component (PCA3), together with FSWDN, VEWN, DDW and RLWD, has relatively high factor loading, of which VEWL, RLW, FSWDW and ODW have obvious relevance with erroneous data accessing. Therefore, we call this prime component erroneous data accessing. The forth prime component (PCA4), together with WAOP and LPO, has relatively high factor loading, which demonstrates erroneous arithmetic. Therefore, we call it as erroneous arithmetic factor. The fifth prime component (PCA5), together with ID and AD, has relatively high factor loading, and is called erroneous documentation.

We can get the scores of each measurement factor based on coefficient of factor scores of PCA and standardization of original variable. Table 3 (the second phase of our case) shows prime component scoring of defect types under different milestones. Variance analysis demonstrates that in five prime components only erroneous arithmetic factor doesn't have obvious differences under different milestones, while the other four primary components have obvious differences under different milestones.

3.3 Discriminate Analysis of Defect Type

Stepwise discriminate analysis in discriminate analysis goes to discriminate function by the smallest Wi1ks statistic. The variable in discriminate function F is used as the criterion of screening variables. When F>3.84, the variable is introduced; or when F<2.71, the variable is removed [4][5][6].

Stepwise discriminate analysis of five primary components demonstrates that erroneous arithmetic factor <2.71 among different milestones, therefore factor of erroneous arithmetic is removed from discriminate function, and the whole software quality discriminate function is:

LSI, VIFNP, MCT, ODW and ASD can be screened as software quality evaluation index, of which LE, BTSUI and LSI are key indexes to indicate software quality.

$Y_{\text{TOTAL QUALITY}}$ =1.9298* erroneous logic -0.0589* erroneous specification -0.4699* erroneous data accessing -0.4498* erroneous documentation

Discriminate coefficients of a certain factor demonstrate its importance in discriminate function. Discriminate coefficients of erroneous logic are far greater than other factors, and it is the major factor to evaluate the whole software quality. Discriminate coefficients of erroneous specification are much less than other factors, and no consideration can be made in evaluating the whole software quality.

Stepwise discriminate analysis of defect attributes index that composes erroneous logic factors demonstrates that the discriminate coefficient of LE, BTSUI and LSI is relatively high and can be regarded as erroneous logic evaluation index, and discriminate function is as follows:

Y_{LOGIC} =-3.47880* LE+1.6776* BTSUI + 1.2477* LSI + 0.5385*VFFNP + 0.3766* MCT + 0.3037* DIDN + 0.2578* DHE + 0.1586* LSI

Stepwise discriminate analysis of defect attributes index that composes erroneous data accessing factors demonstrates that OWD has relatively high discriminate coefficient and can be regarded as erroneous data accessing evaluation index, and discriminate function is as follows:

$Y_{\text{DATA ACCESS}}$ =0.9087* OWD + 0.3178* VEWL

Stepwise discriminate analysis of defect attributes index that composes erroneous documentation factors demonstrates that ASD has relatively high discriminate coefficient and can be regarded as erroneous documentation evaluation index, and discriminate function is as follows:

$Y_{\text{DOCUMENTATION}}$ =0.8978* ASD + 0.4187* AD + 0.2379* ID

You can get the full paper from us by e-mail (d9206006@ms2.ttu.edu.tw).