An Approach to Outlier Detection of Software Measurement Data using the K-means Clustering Method *

Kyung-A Yoon, Oh-Sung Kwon, Doo-Hwan Bae
Software Engineering Lab., Department of EECS
Korea Advanced Institute of Science and Technology, Daejon 305-701, Korea
{kayoon, oskwon, bae}@se.kaist.ac.kr

Abstract

The quality of software measurement data affects the accuracy of project manager's decision making using estimation or prediction models and the understanding of real project status. During the software measurement implementation, the outlier which reduces the data quality is collected, however its detection is not easy. To cope with this problem, we propose an approach to outlier detection of software measurement data using the k-means clustering method in this work.

1. Introduction

The quality of software measurement data affects the accuracy of project manager's decision making using estimation or prediction models and the understanding of real project status. During the software measurement implementation, the invalid data and outlier which reduce the data quality can be collected by various reasons. The invalid data which is described as invalid data type, scale and range can be found by checking their format according to their related definition in software measurement specification. However, the detection of outlier is not easy. As the related work, the method of noisy instance detection using ensemble filter which consists of 25 classifiers [1] proposed, however it can be applied to the only classification problem.

In this paper, we propose an approach to outlier detection of software measurement data using the k-means clustering method. Conceptually, we define the *outlier* as the software data whose value is inconsistent with the main characteristics and relationships which majority data's attribute values have. In addition, the outlier is redefined practically with the viewpoint of the k-means clustering method by categorizing into the external and internal outlier. We performed the case study with real project data of a financial company in Korea. The remainder of this paper is organized as follows. The k-means clustering method is briefly introduced as background and the outlier is redefined in Section 2. Section 3 explains the overall procedure of outlier detection and Section 4 describes the case study. Finally, we conclude with future works in Section 5.

*This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement).(IITA-2006-(C1090-0603-0032))

2. K-means clustering and outlier

2.1. K-means clustering

Generally, statistical methods such as regression model are more popular than data mining methods in our area. Their results are highly accurate if the target data fits a specific distribution model. However, many data sets do not fit one particular model [2] and the software measurement data do not either. On the other side, data mining methods do not constrain the data to be fitted some specific model. By considering the characteristics of software measurement data, we choose the k-means clustering method to the outlier detection method. According to the survey work [2], the k-means clustering is an unsupervised method and can support high-dimensional data. Given a data matrix composed of observations and variables, the objective is to cluster the observations into groups that are internally homogeneous and heterogeneous from group to group [3]. The k of the k-means clustering method indicates the number of groups which is established a priori by expert. To calculate the degree of homogeneity and heterogeneity, the k-means clustering method employs the Euclidean distance as a measure of the similarity between observations and groups. The distance function D using the Euclidean distance is like eq.(1):

$$D = \sum_{i=1}^{k} \sum_{j \in S_i} |x_j - \mu_i|^2 \quad (1)$$

0-7695-2886-4/07 $20.00 © 2007 IEEE
DOI 10.1109/ESEM.2007.49

where there are k clusters S_i, $i = 1, 2, ..., k$ and μ_i is the centroid or mean point of all the points $x_j \in S_i$. We eliminate the detail operational flow of the k-means algorithm to save the space.

2.2. Outlier

In Section 1, we defined outlier as the software data which is inconsistent with the majority data. The concept of *outlier* needs to be redefined with the viewpoint of the k-means clustering algorithm to establish the outlier detection method using it. A survey [2] mentions that the point is outlier if it lies outside all clusters. With the same concept, we redefine an outlier by categorizing into these things:

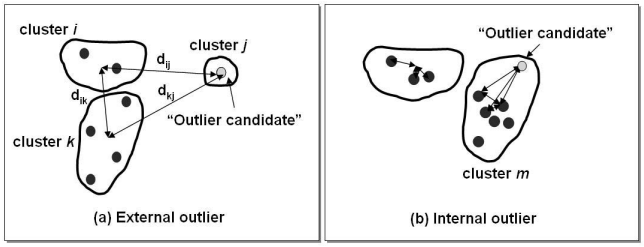

Figure 1. Two kinds of outlier.

(a) External outlier The outlier is an element of the group which has few data and is located far from other groups. The (a) of the Figure 1 illustrates that the *cluster j* has low similarity between other clusters because it contains only one element and the distance d_{ij} and d_{kj} are longer than d_{ik}.

(b) Internal outlier Compared with the distance of the other data in the same group, the outlier is located far from the centroid. The (b) of the Figure 1 illustrates that the gray element has low similarity between other elements because it has long distance in the *cluster m*.

3. Procedure of outlier detection using the k-means clustering method

Figure 2 presents our overall approach. It consists of the following three phases:

Phase 1. Data preparation First, the target observations and attributes should be determined to improve the accuracy of result and the performance of k-means clustering algorithm. Second, if the original data set has missing data, its handling activity must be done. Because the software measurement data is not easy to be collected and its size usually is small, we impute the maximum likelihood data which is anticipated by

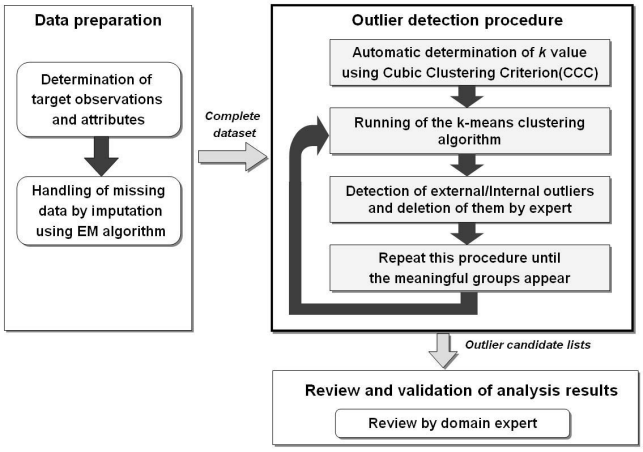

Figure 2. Overall approach.

Expectation Maximization(EM) algorithm [4] to the missing data.

Phase 2. Outlier detection procedure In this phase, the outlier detection procedure using the k-means clustering method is performed with the complete data set resulted from the previous phase. First, the k value should be determined to run the k-means clustering algorithm. Finding the proper k value is difficult and heuristic, so we decide it by referring to the value of Cubic Clustering Criterion(CCC) [5]. Second, the k-means clustering algorithm runs with the decided k value. After completion, the expert searches the external and internal outliers in the clustering results. If the other groups are meaningful after eliminating the outliers, stop this procedure. However, if the other groups are needed to be recalculated, repeat from running the k-means clustering algorithm without the detected outliers.

Phase 3. Review and validation In the domain expert's aspect, the results of previous phase are only the candidates list of outlier. Therefore, this phase should be performed to find the true outlier by considering the domain knowledge.

4. Case study

4.1. Software project data set

Our empirical case study used the software project data which were measured for two years in a financial company in Korea. Since this company acquired the CMMI level 3 certification, the software project information has been measured continuously. Before transferring to the CMMI level 4, it needs to have confidence in the quality of software

measurement data and the working results. Most of influential measurement data are collected by hand and recorded by means of a measurement tool. The original data set consists of 112 observations and 36 attributes. In the data preparation phase, we excluded the observation which has more than 70% missing values and whose measurement unit is mismatch with the definition in measurement specification. In addition, we reduced the number of target attributes according to their importance in this company. The missing values appear in 3 attributes of 2 observations, so they are imputed by new values created by EM algorithm. After the data preparation phase, 104 target observations and 24 attributes are remained like Figure 3.

Category	Counts	Attributes
Project environment	7	Scale, Lifecycle model, Methodology, Development type, Platform etc.
Cost	7	Effort, Max team size, Reused code size, New development code size, Cost of test etc.
Quality	4	Total number of defects, Defect density etc.
Delivery	1	Duration
Productivity	2	Productivity, Document productivity
Others	3	Measures related to the accuracy of prediction

Figure 3. The target attributes of data set.

4.2. Results

We used SAS Enterprise Miner 4.3 to execute the k-means clustering algorithm. It supports the automatic determination of the k value based on CCC, the 6 popular clustering criterions, and various graphical analysis view. After iterating the outlier detection procedure three times, we finally detected total 14 outliers that consist of 10 external outliers and 4 internal outliers. Except the detected outliers, we analyzed the general characteristics of the remained 90 observations which are clustered to four groups like Figure 4. The detected outliers are inconsistent with the majority data's characteristics as Figure 4. They mainly have extreme values or inconsistent relationships between some attributes. During the review and validation of analysis result phase, the domain expert selected two normal data among 14 outlier candidates. He analyzed the source of 12 outliers with the related project artifacts and gave two important feedbacks like these:

- The main source of 5 outliers are inferred that the developers who came from a supplier company were not accustomed to the development process of this company. Two outliers were made by the data collector's mistake and the others are the extreme values.
- The results of outlier detection using clustering should have more detail information of the reason why the target observation is outlier on which criterion.

Cluster	Scale	Lifecycle	Methodology	Domain	Type
1	M/L	V-model	CBD	S	M
2	M/L	V-model	INF	S	D
3	S/M	V-model	INF	A	M
4	S/M	V-model	INF	I	M

Cluster	Platform	Database	Duration (day)	Effort (person-hours)
1	Unix	DB2	78.5	3319.35
2	Windows	other	101.8	3300.47
3	Host	HDB	57.9	1343.93
4	Unix	DB2	50.5	980.85

* Scale: S(Small), M(Medium), L(Large)
* Methodology: CBD(Component-based Development), INF(Information-based Development)
* Domain: S(Server-side), A(Account-side), I(Information-side)
* Type: M(Maintenance), D(New Development)

Figure 4. The clustering result of final iteration.

5. Conclusions

In this paper, we proposed an approach to outlier detection of software measurement data using the k-means clustering method. To improve the quality of measurement data, organizations have to detect outliers, find the source of them, and try to prevent it. We found 11.5% outliers among 104 observations in our case study. This work is our preliminary work to detect outliers of software measurement data using clustering method. We plan to improve our work for increasing the detection accuracy.

References

[1] Taghi M. Khoshogoftaar, Vedang Joshi, and Naeem Seliya, "Detecting Noisy Instances with The Ensemble Filter: A Study in Software Quality Estimation", *International Journal of Software Engineering and Knowledge Engineering v.16 no.1 pp.53-76*, 2006

[2] Victoria J. Hodge and Jim Austin, "A Survey of Outlier Detection Methodologies", *Artificial Intelligence Review pp.85-126*, 2004

[3] Paolo Giudici, *Applied Data Mining: Statistical Methods for Business and Industry*, Wiley, 2003

[4] Ingunn Myrtveit, Erik Stensrud, and Ulf H. Olsson, "Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods", *IEEE Transaction on Software Engineering, v.27 no.11 pp.999-1013*, 2001

[5] Warren Sarle, "Cubic Clustering Criterion", *SAS(R) Technical Report A-108, SAS Institute Inc.*, 1983 Wiley, 1987