Some Open Problems in Software Project Data Analysis

Akito Monden
Nara Institute of Science and Technology
akito-m@is.naist.jp

Abstract

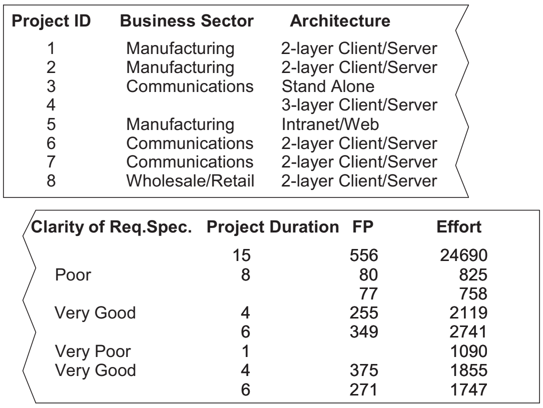
This paper focuses on typical difficulties in software project data analysis, and proposes topics for workshop discussion.

1. Introduction

In the past decade, the author has been analyzing various multivariate data sets of software projects typically shown in Figure 1. This paper describes typical problems the author had faced in analyzing such multivariate data.

A data set like Figure 1 is often called "software engineering data repository," typically collected and by a project management office or a quality assurance office in a software company. One of such data sets available to researchers is the ISBSG (International Software Benchmarking Standards Group) repository [1], which consists of more than 3000 projects each having about 100 project features, collected from more than 20 countries. Also, NASA IV&V Metrics Data Program [2] provides access to data repositories, which consist of about 10 projects each having software metrics and the associated error data at the function/method level.

Such data sets often suffer from problems come from human factors in data collection, typically; (1) definitions of variables are not strict enough, (2) reliability of measurement widely varies, and (3) a lot of missing values and outliers exist. Because of these problems, researchers need to spend a great effort on "cleaning up" the data sets before they are ready to conduct statistical analyses. Moreover, even after such cleaning, the data sets pose some inherent difficulties in statistical analysis; (1) mixture of quantitative and qualitative variables, (2) hidden relationship among variables, and (3) non-Gaussian distribution of metrics values. Below this paper addresses these problems in detail.



Project ID	Business Sector	Architecture
1	Manufacturing	2-layer Client/Server
2	Manufacturing	2-layer Client/Server
3	Communications	Stand Alone
4		3-layer Client/Server
5	Manufacturing	Intranet/Web
6	Communications	2-layer Client/Server
7	Communications	2-layer Client/Server
8	Wholesale/Retail	2-layer Client/Server

Clarity of Req.Spec.	Project Duration	FP	Effort
	15	556	24690
Poor	8	80	825
		77	758
Very Good	4	255	2119
	6	349	2741
Very Poor	1		1090
Very Good	4	375	1855
	6	271	1747

Figure 1: An example of a project data set.

2. Problems ahead of Analysis

2.1 Soundness of Data Definition

It is often the case that definitions of variables are not strict enough. For example, suppose we have a variable "clarity of requirement specifications" of four categories "very good", "good", "poor" and "very poor", it is quite difficult to give a strict definition for each category. A data analyst needs to consider such vagueness of variable definitions.

2.2 Reliability of Measurement

Even though a variable was well defined, measurement could be unreliable. For example, the function points measured in an upstream development phase often do not correctly describe the functional size of a finished product since the product functionality often grows up as a project progresses. Measuring the effort and the number of developers are also very difficult because of development outsourcing.

Another side of difficulty comes from human factors. For example, measuring SLOC excluding automatically generated code lines is often very difficult; thus, some people may not care about the generated code in measurement. For another example, people tend to enter a "default value" rather than entering a real value when using a data measurement tool. For example, typical bug tracking systems require developers to enter the "bug severity," but people may enter "3 = medium" always just because it is a default value. Therefore, an analyst needs to be aware of how the data is measured in the field.

2.3 Missing Values

Generally, a software project data set contains a lot of missing values, while many statistical analysis methods and data mining methods require a data set having non missing values. Therefore, an analyst often needs to do some value imputation. For example, for a nominal scale variable, a new category "unknown" can be added to assign missing values. For an ordinal/ratio scale variable, a median or a mean value can be assigned to a missing value. Also, there are more sophisticated imputation methods such as k-nearest neighbor method.

However, value imputation should not be applied if a variable or a project has too many missing values. In such a case, an analyst should consider deleting some of projects and variables from the data set.

An analyst also needs to consider not doing any imputation since some statistical methods allow missing values, e.g. scatter diagram and association rule mining.

2.4 Outliers

Outliers are also inherent in a software project data set. One root cause is a human error, e.g. recording "person hours" as "person months" by mistake. Another root cause comes from the nature of project individuality, e.g. a project may have double effort because of project failure and recovery. Before doing any analysis, we need to (at least) be aware of outliers in a data set.

3. Difficulties in Analysis

3.1 Mixture of Quantitative and Qualitative Variables

A project data set usually contains both quantitative (ratio scale or interval scale) and qualitative (ordinal scale or nominal scale) variables, while many statistical methods, e.g. regression analysis, cannot handle the both at the same time. An analyst needs to do some scale translation before an analysis.

3.2 Hidden Relationship among Variables

There are a lot of hidden relationships among variables; and, such relations make analysis very difficult. For example, if a statistical analysis revealed that the development productivity had high correlation with the outsourcing percentage, this does not directly mean the high outsourcing percentage results in low productivity because high outsourcing percentage projects often have a large team size, and the large team size usually causes low productivity.

3.3 Biased Value Distribution

Many quantitative variables do not follow the Gaussian distribution. It is partly because small projects (or modules) exist much more than large projects (modules). Since many statistical methods assume the Gaussian distribution, an analyst must be careful in applying statistical methods, and should consider using non-parametric methods.

4. Summary

This paper described several difficulties that the author had faced when analyzing a software development data set. In the workshop, the author wish to discuss on the effective way of analysis.

Acknowledgement

This work is being conducted as a part of Stage Project, the Development of Next Generation IT Infrastructure, supported by Ministry of Education, Culture, Sports, Science and Technology.

References

[1] International Software Benchmarking Standards Group: ISBSG Estimating, Benchmarking and Research Suite, http://www.isbsg.org/

[2] NASA IV&V Facility Metrics Data Program, http://mdp.ivv.nasa.gov/