Measuring the Heterogeneity of Cross-company Dataset

Jia Chen, Ye Yang, Wen Zhang
Institute of Software,
Chinese Academy of Sciences,
Beijing, China

{chenjia, ye, zhangwen}@itechs.iscas.ac.cn

Gregory Gay
Lane Department of Computer Science and
Electrical Engineering,
West Virginia University,
Morgantown, WV, USA

gregoryg@csee.wvu.edu

ABSTRACT

As a standard practice, general effort estimate models are calibrated from large cross-company datasets. However, many of the records within such datasets are taken from companies that have calibrated the model to match their own local practices. Locally calibrated models are a double-edged sword; they often improve estimate accuracy for that particular organization, but they also encourage the growth of local biases. Such biases remain present when projects from that firm are used in a new cross-company dataset. Over time, such biases compound, and the reliability and accuracy of a general model derived from the data will be affected by the increased level of heterogeneity. In this paper, we propose a statistical measure of the exact level of heterogeneity of a cross-company dataset. In experimental tests, we measure the heterogeneity of two COCOMO-based datasets and demonstrate that one is more homogeneous than the other. Such a measure has potentially important implications for both model maintainers and model users. Furthermore, a heterogeneity measure can be used to inform users of the appropriate data handling techniques.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – cost estimation, time estimation

General Terms

Economics, Experimentation, Management

Keywords

Heterogeneous datasets, software effort estimation, parameter comparison, estimation model calibration.

1. INTRODUCTION

Obtaining an accurate estimate of software effort is of great value for project managers as well as other stakeholders. Such an estimate can be used to appropriately arrange various activities of software development, as well as to allocate the suitable amount of resources to these activities. However, obtaining an estimate that is both accurate and reliable is a difficult task. To address this problem, many different software effort estimation methods have been proposed. These methods tend to fall into three categories: expert-based, analogy-based, and model-based estimation.

Model-based estimation, as the name implies, makes use of a mathematical model, such as COCOMO [3, 4], to produce effort estimates. This approach is also known as parametric estimation, as there are several variable parameters within the model that must be determined before that model is used. Initially, the values of these parameters are often supplied by experts or the model's designers. The software engineering industry changes rapidly, and regardless of the model's accuracy, the parameters used to estimate projects from the previous decade are unlikely to remain relevant to modern projects. The regular re-calibration of model parameters, either through a series of new expert judgments or through the use of an Ordinary Least Squares (OLS)-based method, ensures the reliability of a model's estimates.

The calibration of the general COCOMO model has met with several problems, the most major of which is the existence of counter-intuitive – that is, negative – regression coefficients [4, 5, 6, 7]. Such coefficients make little logical or practical sense; a higher level of programmer capability (PCAP) leads to a decrease in the calculated project effort, but a negative coefficient indicates that higher PCAP, unreasonably, increases the effort level.

To adapt a general COCOMO model calibrated from a cross-company dataset to the local environment of a particular firm, it is advised to locally-calibrate the general model, which involves tuning the two constants representing the overall productivity of the firm [4, 5]. While the local model often provides more accurate estimates for that company in the short term, its long-term use can be harmful. If local calibration is highly effective for a company, it logically follows that this company's practices differ from the mainstream (that is, the average practices that can be summarized from the general model) by some significant amount. In other words, firms that benefit from local calibration demonstrate a local bias. These firms may grow content with the accurate estimates provided by their local models, and will be unlikely to change their practices. Over time, even if their models still provide accurate estimates, they will grow more unreliable.

We propose an approach to measuring the heterogeneity of COCOMO datasets by comparing the calibrated parameters of a cross-company dataset with a derived version of that dataset where the effort values have been replaced with those given by locally-calibrated models. This is because we believe that the heterogeneity is aggregated into the estimate given by each local model. By this comparison, we can calculate a measure of the heterogeneity of the original dataset.

2. Related Work

There is a large body of work on comparing software effort estimation models derived from within-company datasets with those derived from cross-company datasets. Kitchenham et al. [1] systematically reviewed 10 such papers with the aim of determining under what circumstances estimation models derived from cross-company datasets are as good as those derived from within-company datasets. It was only certain that models derived from within-company datasets were significantly better (that is, more accurate) than models derived from cross-company datasets when the within-company datasets were small (less than 20 projects) and leave-one-out cross validation was used.

Jeffery et al. [2] compared the accuracy of estimation models derived from the ISBSG repository (a cross-company dataset) with those derived from the dataset of an Australian company called Megatec (a within-company dataset). They found that the model derived from the within-company dataset was significantly more accurate than the model derived from the cross-company dataset. Several papers have supported this conclusion [8, 9], while others have rejected it (see [1] for a complete list).

Rather than pure model comparison, some authors have focused on the preliminary analysis of datasets used to build an estimation model. Kitchenham [11] proposed a procedure for analyzing unbalanced datasets, and helped to explain the difficult situation that happened when COCOMO II was initially calibrated [5]. Based on forward pass residual analysis, the procedure identifies really significant factors, and then produces a better model. Another paper by Liu et al. [10] proposed a rather generic framework for preliminary analysis of cost estimation dataset. Using this framework, analyst can systematically remove outliers and identify dominant variables.

In order to improve the accuracy of estimation models derived from datasets with heterogeneous sources (such as the ISBSG database), Cuadrado-Gallego et al. [12] proposed an automated segmentation process that splits a single parametric model into a number of sub-models. However, the segmentation of a general model will dramatically decrease the maintainability of the general model over time, and lead to the lack of a common basis for comparing estimates produced by different model variants.

We focus on the task of measuring the pure level of heterogeneity in a cross-company dataset and what implications such heterogeneity has for the life cycle management of cost models. Instead of comparing estimation accuracy, we compare calibrated parameters and come to a specific measurement.

3. RESEARCH METHOD

3.1 Overview

We first take the cross-company dataset, called the original dataset, and filter out any unusable within-company subsets (those with less than three projects are too small to make use of). Using the slightly smaller original dataset, we build a second, derived, dataset as follows:

(1) The original dataset consists of several disjoint subsets, each consisting of records from a single organization. We derive a local model for each of those subsets through calibrating the A and B constants by the OLS method.

(2) For each project contained in a within-company dataset, we calculate its effort estimate using the local model for the organization that contributes the project.

(3) Copy the original dataset, replacing the recorded actual effort with the model's estimate. The derived dataset differs from the original dataset only in effort values.

These two datasets (the original and derived datasets) form the basis of the subsequent heterogeneity comparison. We then repeat the following three steps in sequence for both datasets.

(1) Randomly select about 90% of projects from the dataset.

(2) Calibrate the parameters of a general model from selected projects by the OLS method.

(3) Save all of the calibrated parameters as a new element of a predefined array. Each element is a set of values for all calibrated parameters.

After a specified number of trials, we fill two arrays of value-sets for all of the calibrated parameters (one array per dataset). From another perspective, there are two arrays of values for each calibrated parameter. We regard each array as a random sample of the same calibrated parameter. In other words, we actually take a pair of random samples for each calibrated parameter. By means of the statistical test defined in Section 3.4, we determine whether or not the difference of two means of a calibrated parameter is equal to zero. The p-value of the statistical test is used as an indicator of the difference.

Finally, we count the number of calibrated parameters whose p-values are relatively large, compared with others. We use the proportion of calibrated parameters with small p-values as a positive indicator of the degree of heterogeneity. This provides a quantitative measure of the heterogeneity of a cross-company dataset. The foregoing process is applied to both NASA and USC datasets. Since it has a parameter that specifies the times of repetition, this parameter's value is varied in order to test its effect on the results of our measurement.

3.2 Datasets

We examine two datasets in this paper, and explore their varying degrees of heterogeneity. The first is the NASA93 dataset from the National Aeronautics and Space Administration (NASA). The other is a subset of the COCOMO II dataset from the University of Southern California (USC). Both datasets use a variant of the COCOMO software effort estimation model. Both contain effort multipliers and the two COCOMO constants as variable parameters, but the NASA dataset lacks scale factors. Readers are referred to Boehm [3] and Boehm et al [4] for detailed definitions of these effort multipliers and scale factors.

The software size is measured in KSLOC (thousand of logical line of code), and the development effort is measured in PM (Person Month). Table 1 compares some statistics for the software size and development effort of these two datasets. In this table, we see that the NASA dataset has less variety than the USC dataset.

Table 1 Software size and development effort of NASA and USC datasets

	Software Size		Development Effort	
	NASA	USC	NASA	USC
Mean	94.02	130.9	624.41	711.03
S.D.	133.6	236.23	1135.93	1519.3
Min	0.9	2.6	8.4	6
Max	980	1292.8	8211	11400

Projects of the NASA dataset are contributed by several centers that are geographically distributed across the United States, and we treat each center as an individual company. Similarly, projects of the USC dataset are contributed by many organizations, and we treat each organization as an individual company. Thus, we divide each dataset into several disjoint subsets whose projects are contributed by the same company. Each whole dataset is, by definition, a cross-company dataset, and each such subset of it forms a within-company dataset.

3.3 Data Preparation

In the local COCOMO model, there are only two calibrated parameters: A and B. In order to calibrate them by the OLS method from a within-company dataset, there must be no less than three projects in the dataset. As we need to build a local model for each within-company dataset, we have to exclude those within-company datasets whose sizes are less than three. Applying this filter to NASA and USC datasets, we excluded one within-company dataset from the former, and two within-company datasets from the latter. As a result, 91 projects of the NASA dataset are distributed among 4 within-company datasets, and 158 projects of the USC dataset are distributed among 14 within-company datasets. Within-company datasets range in size from 3 to 39 for the NASA dataset, and from 3 to 48 for the USC dataset.

3.4 Statistical Test

As a result of repeatedly calibrating parameters, each of these parameters has two arrays of values (see Section 3.1). One array is from the original dataset, and the other is from the derived dataset. Each array can be regarded as a random sample of the same calibrated parameter and each of its elements as an observation of the sample. Based on these observations, we can calculate the sample mean and sample variance for hypothesis testing.

We do not directly test the null hypothesis that the difference between two means of a calibrated parameter is equal to zero. Instead, we calculate the p-value of such a test, which is defined by the following equations.

$$p = 2\times\left(1-\mathrm{PDF}_{\mathbf{Z}}\left(z\right)\right)$$

$$z = \left|\frac{\overline{x}-\overline{y}}{\sqrt{\frac{s_1^2+s_2^2}{n}}}\right|,\ \mathrm{PDF}_{\mathbf{Z}}\left(z\right) = \mathrm{P}\left\{\mathbf{Z}\le z\right\}$$

$$\mathbf{Z}\sim\mathbf{N}\left(0{,}1\right)$$

Where $\overline{x}$ and $\overline{y}$ are the two sample means of a calibrated parameter, s_1^2 and s_2^2 are the two sample variances of a calibrated parameter, n is the number of observations in a sample, $\mathbf{Z}$ is a random variable of standard normal distribution, and $\mathrm{PDF}_{\mathbf{Z}}$ denotes the Probability Distribution Function of $\mathbf{Z}$.

Note that these two samples are of equal size, because we specify the same number of trials for both datasets. That is, the sample size equals the times of repetition. We do not use the common t-test, because it assumes that two distributions have the same variance, but we observe that the two sample variances are quite different. In the equation for z, we replace the variances of the two distributions with the sample variances respectively, simply because the former is not available and the latter is a good approximation. For our hypothesis, a lower p-value implies that the two means of a calibrated parameter are more probably different from a statistical perspective.

3.5 Measure of Heterogeneity

For each of the calibrated parameters, we calculate the p-values of the statistical test defined in the previous subsection. There are some calibrated parameters whose p-values are relatively smaller than others. We count the number of these small p-values and calculate their proportion with regard to total number of calibrated parameters. This proportion is proposed as a positive indicator for the degree of heterogeneity of a cross-company dataset. The larger the proportion is, the greater the heterogeneity.

$$Heterogeneity = \frac{s}{n}$$

Where, s denotes the number of calibrated parameters that has small p-values, and n denotes the total number of calibrated parameters.

Currently, we define a p-value as being small if it is less than 0.025. The value is chosen because it is a common choice of significance level for hypothesis testing, and it acts as a clear boundary when the heterogeneity is calculated for the two cross-company datasets used in this paper.

4. RESULTS AND DISCUSSION

Two cross-company datasets are used in this paper. One is the NASA dataset, and the other is the USC dataset. We apply the process summarized in Section 3.1 to each of these datasets three times, each time with a different value for the parameter that specifies how many calibration trials are conducted. Doing this, we can see how the p-value derived from statistical test varies with this parameter.

Table 2 Calibrated parameters with large p-values

	NASA			USC	
Trials	B	VIRT	MODP	TEAM	RUSE
36	0.5642	0.534	0.1477	0.1174	0.08853
48	0.5322	0.3397	0.1532	0.06147	0.1527
60	0.5247	0.4184	0.04999	0.02535	0.05405

Table 2 lists the exact number of p-value for the calibrated parameters with large p-values. For most of these calibrated parameters, their statistical tests reject the null hypothesis (that the difference between two means of a calibrated parameter is zero) at a significance level of 0.05.

The USC dataset has a *smaller* proportion (2/25) of calibrated parameters with *large* p-values, compared with the NASA dataset (3/17). Therefore, the former has *greater* degree of heterogeneity than the latter. This result agrees with two characteristics of these two datasets. And these two characteristics help to explain why the USC dataset has a greater degree of heterogeneity than the NASA dataset.

(1) The USC dataset has a greater degree of variety in the software size and development effort than the NASA dataset (see Table 1). These two attributes can substantially influence the values of calibrated parameters.

(2) There are merely 4 within-company datasets in the NASA dataset, but 14 within-company datasets in the USC dataset. It is usually true that a cross-company dataset consisting of more within-company datasets has a greater degree of heterogeneity.

However, these two intuitive characteristics cannot determine the degree of heterogeneity. Suppose a large cross-company dataset consists of many homogeneous within-company datasets, it can demonstrate great variety as long as all the within-company datasets demonstrate it, but little heterogeneity will be measured.

Before calibrating a general estimation model such as COCOMO model, our method may be used to analyze the heterogeneity of the source dataset. Based on the results of analysis, a user can determine whether a cross-company dataset is appropriate for calibrating a general estimation model. In many cases, however, model users and maintainers alike are restricted to what data they have in their possession. If there is a high degree of heterogeneity in the dataset, it is inadvisable to use the data “as is.” Instead, one approach suggested by Cuadrado-Gallego et al. [12] could be used to form a segmented (composite) general model that consists of several sub-models, instead of using an overall (singleton) general model that consists of a single equation. However, this approach cannot be applied to COCOMO, because its definition strictly stipulates that a single equation should be used.

An open research question is the exact definition of small p-value. We suggest that a relative approach should be adopted for determining the appropriate level for small p-value, because the appropriate level largely depends on the method used to calibrate the general estimation model. In this paper, we used the Ordinary Least Squares method and assigned a level of 0.025 to the small p-value. However, this is not universally appropriate - other methods may require different levels for small p-value.

5. CONCLUSIONS

In this paper, we propose a method that statistically compares the calibrated parameters of a general estimation model on the basis of a cross-company dataset. The results of this comparison can be used to calculate the heterogeneity of this cross-company dataset. We propose that the proportion of calibrated parameters with small p-values should be used as a positive indicator of the degree of heterogeneity. The larger the proportion is, the greater the heterogeneity.

The ability to measure the heterogeneity of a dataset has important implications for both the maintainers of the general model and for the organizations that typically employ locally-biased models. By using more homogenous datasets, general models can be frequently re-calibrated with some expectation of reliability. This would ease the difficulty of maintaining such models and would help ensure that general models remain relevant to the frequently changing state of the software engineering field. A more homogenous general model, or even a method of determining the heterogeneity of the data that was used to calibrate the general model, could then create a feedback loop where an organization uses the more homogeneous general model to evolve their practices into proximity to the mainstream, which in turn would help further homogenize the general model. Furthermore, such a measure can be used to steer model users away from overtly heterogeneous datasets. Model users restricted to heterogeneous datasets can make more informed decisions about how to use their data. Rather than using it “as is,’ they may elect to use a data preprocessing technique to filter out some of the heterogeneity.

6. ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant Nos. 90718042, 60873072, and 60803023; the National Hi-Tech R&D Plan of China under Grant No. 2007AA010303; the National Basic Research Program under Grant No. 2007CB310802.

7. REFERENCES

[1] B. A. Kitchenham, E. Mendes, and G. H. Travassos, “Cross versus within-company cost estimation studies: A systematic review,” *IEEE Transactions on Software Engineering*, vol. 33, no. 5, pp. 316-329, May, 2007.

[2] R. Jeffery, M. Ruhe, and I. Wieczorek, “A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data,” *Information and Software Technology,* vol. 42, no. 14, pp. 1009-1016, Nov, 2000.

[3] B. W. Boehm, *Software Engineering Economics*: Prentice Hall PTR, 1981.

[4] B. W. Boehm, Clark, Horowitz *et al.*, *Software Cost Estimation with Cocomo II with Cdrom*: Prentice Hall PTR, 2000.

[5] B. Clark, S. Devnani-Chulani, B. Boehm *et al.*, "Calibrating the COCOMO II Post-Architecture model," *Proceedings of the 1998 International Conference on Software Engineering*, International Conference on Software Engineering, pp. 477-480, Los Alamitos: IEEE Computer Soc, 1998.

[6] V. Nguyen, B. Steece, B. Boehm *et al.*, *A Constrained Regression Technique for COCOMO Calibration,* New York: Assoc Computing Machinery, 2008.

[7] S. Chulani, B. Boehm, and B. Steece, “Bayesian analysis of empirical software engineering cost models,” *IEEE Transactions on Software Engineering,* vol. 25, no. 4, pp. 573-583, Jul-Aug, 1999.

[8] E. Mendes, B. Kitchenham, and s. IEEE computer, *Further comparison of cross-company and within-company effort estimation models for web applications*, Los Alamitos: IEEE Computer Soc, 2004.

[9] K. Maxwell, L. Van Wassenhove, and S. Dutta, “Performance evaluation of general and company specific models in software development effort estimation,” *Management Science,* vol. 45, no. 6, pp. 787-803, Jun, 1999.

[10] Q. Liu, and R. Mintram, “Preliminary data analysis methods in software estimation,” *Software Quality Journal,* vol. 13, no. 1, pp. 91-115, Mar, 2005.

[11] B. Kitchenham, “A procedure for analyzing unbalanced datasets,” *IEEE Transactions on Software Engineering,* vol. 24, no. 4, pp. 278-301, Apr, 1998.

[12] J. J. Cuadrado-Gallego, and M. A. Sicilia, “An algorithm for the generation of segmented parametric software estimation models and its empirical evaluation,” *Computing and Informatics*, vol. 26, no. 1, pp. 1-15, 2007.