An Approach to Global Sensitivity Analysis: FAST on COCOMO

Stefan Wagner
Institut für Informatik, Technische Universität München
Boltzmannstr. 3, 85748 Garching b. München, Germany
wagnerst@in.tum.de

Abstract

There are various models in software engineering that are used to predict quality-related aspects of the process or artefacts. The use of these models involves elaborate data collection in order to estimate the input parameters. Hence, an interesting question is which of these input factors are most important. More specifically, which factors need to be estimated best and which might be removed from the model? This paper describes an approach based on global sensitivity analysis to answer these questions and shows its applicability in a case study on the COCOMO application at NASA.

1 Introduction

In software engineering, there is a variety of (mathematical) models that aim to predict certain aspects of the development process and artefacts. Those so-called *predictor* or *prediction models* are a valuable tool in improving processes and making software projects more manageable. Analytical and empirical models are the only way to find well-founded answers to questions like "How long will the project take?" or "How much effort do we need to spend?"

However, those models are complex to build and even more complex to validate. Often, there is simply not enough empirical data available. Moreover, analytical models tend to be over-parameterised because all input factors that *might* have an influence are included. The reason for this is that it is often very hard to find out which are the "important" factors. Hence, we need simple but well-founded ways to analyse such models w.r.t the importance of the input factors.

This paper describes an approach to global sensitivity analysis for predictor models in software engineering. The proposed method includes using scatterplots and the *Fourier Amplitude Sensitivity Test* (FAST). We specifically look at the COCOMO effort prediction model [1] in a case study to show the applicability of the approach.

2 Global sensitivity analysis

Mathematical models are applied to make predictions and estimates. They use input factors and equations to characterise the process under investigation. Sensitivity analysis allows to determine the uncertainty associated with such a model. This helps to answer questions like "what input factor needs to be investigated in more detail?" or "what input factors can be removed?".

A recent approach is global sensitivity analysis. It allows to apportion the output uncertainty to the uncertainty of the input factors. The input factors are described by probability distribution functions that represent our knowledge of the factors. This leads to two advantages that we cite directly from Saltelli [5]: (1) *The inclusion of influence of scale and shape*: The sensitivity estimates of individual factors incorporate the effect of the range and the shape of their probability density functions. (2) *Multidimensional averaging*: The sensitivity estimates of individual factors are evaluated varying all other factors as well. Another important aspect is that global methods have the *model independence property*, i.e. the actual linearity or additivity of the model does not influence the functioning of the method.

For this analysis, we have the choice between several methods such as scatterplots or correlation methods. Most useful are *importance measures* that quantify the sensitivity of the input factors using so-called *sensitivity indices* [5]. The indices are divided into *first-order* and *higher-order indices*. The former describe the direct effect a factor has. The latter quantifies the interactions between the different factors. A common method to compute such measures of importance is the *Fourier Amplitude Sensitivity Test (FAST)* [4].

There are re-occurring questions and solutions that are summarised in so-called *settings*. The factors prioritisation (FP) setting ranks the factors in terms of their contribution to the variance and the factors fixing (FF) setting is concerned with model simplification, i.e. which factors can be fixed without influencing the output. The FP setting can be answered using the first-order indices. The factors with

the highest first-order values are most important for further investigation. The answer to the FF setting can be given using the total-order indices. Only input factors with low total-order index can be safely removed from the model.

3 Approach

We propose in the following a way to use global sensitivity analysis for models in software engineering. In particular, we describe three basic steps to analyse models by determining the distributions of the input factors, detecting errors using scatterplots, and quantifying the influence on the output by global sensitivity analysis.

Determining distributions The first step is to determine the distributions of the input factors. This is needed to generate samples that are later used for scatterplots and global sensitivity analysis. In our context, there are mainly four ways to determine the distributions: Scientific literature, expert opinion, empirical project data, and controlled experiments. What is the best way depends largely on the input factor itself.

Visualising using scatterplots In the second step, sample data is generated and visualised by scatterplots. The needed sample data can be generated based on the distributions we determined above. For example, the Simlab[1] tool is able to generate sample data.

The use of the scatterplots is two-fold: (1) detection of errors and (2) first indications of influence. If there are errors in the model implementation or the distribution specification, they will most likely be visible in the scatterplots. Strange curves that suddenly change direction are good indicators for that. Typical scatterplots either look strongly chaotic or follow some kind of curve. A clear curve suggest a high correlation between the factor and the output and hence a probable high influence.

Applying global sensitivity analysis There are various possibilities for global sensitivity analysis but we suggest to use FAST (cf. Sec. 2). Its results are first-order and total-order indices that describe the quantitative difference between the input factors. Using the input distributions, the sampled inputs and the corresponding outputs, we can calculate the sensitivity indices. This is also supported by the Simlab tool. The first-order indices give the share of the output variation that is directly related to each input factor. For example, a first-order index of 0.2 means that the input factor causes directly 20% of the variance of the output factor. Hence, the interpretation is that by reducing the variance in an input factor by determining it more precisely, we can reduce the amount given by the first-order index in the variation of the output.

The total-order indices describe the share of the output variation that is *related* to each input factor. This includes all interactions. A total-order index of 0.12 means that 12% of the output variation is caused by this input factor. This includes the direct effect as well as interactions with other factors. The interpretation is that by removing this factor we remove the amount of the total-order index from the output variation. Hence, we can only remove factors with very small total-order indices in order to not change the output significantly.

4 Case study

We demonstrate the proposed approach in a case study based on published data from NASA [6]. It contains the values used for COCOMO estimations of 60 projects. Hence, we have values for the lines of code and the cost multipliers from projects of a similar domain. This allows us to determine which factors are the most important ones in this domain.

COCOMO is a well-known effort prediction model developed by Boehm [1]. There are two reasons why we investigate COCOMO: (1) It is well-known in research and practice. (2) There is public data available of COCOMO applications. COCOMO uses several input factors. Firstly, the size of the software in KLOC needs to be estimated. Secondly, there are two parameters, a and b, that are determined by the development mode of the project. Thirdly, there are *cost drivers* or *multipliers*. They describe additional project conditions that influence the needed effort. These are for example the level of required reliability (RELY) or execution time constraints (TIME).

We model the empirical data using discrete distributions. Factors a and b depend on the discrete classification of the software and the cost multipliers have one of the values between *very low* and *extra high*. All these correspond to specific numeric values. Hence, we use the frequency of occurrence to determine the probability of each value for each factor. The results are shown in Tab. 1. The situation for the size is different. There is no certain set of discrete values. A Kolmogorov-Smirnov test approved that the sample data for *size* follows an exponential distribution with a significance value $\alpha = .05$ and an estimated parameter λ of .013.

We use the Simlab tool for the generation of 100,000 samples. These samples are used to create scatterplots which are omitted because of space limitations. All of them show an expected behaviour. There are no strange angles or outliers in the plots which would indicate errors.

The factors priorisation setting answers the question which factors are most beneficial to determine with more precision. In Tab. 2 the corresponding indices are shown on

[1] `http://simlab.jrc.cec.eu.int/`

	RELY	DATA	CPLX	TIME	STOR	VIRT	TURN	ACAP
Very low	0	-	0	-	-	-	-	0
Low	0.0167	0.4667	0.0333	-	-	0.7333	0.5	0
Nominal	0.5167	0.2667	0.0833	0.6667	0.7	0.2333	0.2167	0.4833
High	0.4333	0.15	0.8333	0.1167	0.1167	0.0333	0.2833	0.3833
Very high	0.0333	0.1167	0.0333	0.2	0.1333	0	0	0.1333
Extra high	-	-	0.0167	0.0167	0.05	-	-	-
	AEXP	PCAP	VEXP	LEXP	MODP	TOOL	SCED	
Very low	0	0	0	0.0333	0	0.0167	0	
Low	0	0	0.1833	0.0167	0.15	0.0667	0.4167	
Nominal	0.4167	0.6167	0.8	0.25	0.3167	0.65	0.4833	
High	0.3833	0.2667	0.0167	0.7	0.4167	0.0833	0.1	
Very high	0.1333	0.1167	0	-	0.1167	0.1833	0	
Extra high	-	-	-	-	-	-	-	

Table 1. The discrete distributions of the cost multipliers

First Order		Total Order	
size	0.5926	size	0.86904
b	0.0454	b	0.261585
TIME	0.0096	TIME	0.163358
STOR	0.0086	TURN	0.159402
ACAP	0.0061	STOR	0.157869
TURN	0.005	LEXP	0.147762
PCAP	0.0047	AEXP	0.146267
AEXP	0.0034	ACAP	0.143132
TOOL	0.0034	SCED	0.141704
MODP	0.003	CPLX	0.140605
RELY	0.0024	VIRT	0.138147
a	0.0024	MODP	0.137153
DATA	0.0022	PCAP	0.133477
VIRT	0.0022	VEXP	0.132458
CPLX	0.0016	a	0.13193
SCED	0.000676	DATA	0.129065
VEXP	0.000668	TOOL	0.126634
LEXP	0.000296	RELY	0.11941

Table 2. The first order and total order indices

the left-hand side. They indicate that the input factor *size* has by far the highest value and therefore the most influence. It causes nearly 60% of the variation in the output. Nearly all of the other factors are below 1% and hence negligible for very detailed estimations. Only factor b has an influence of nearly 5%.

The factors fixing setting is used for model simplifications. The necessary total-order indices are shown on the right-hand side of Tab. 2. We see that the factors *size* and b are still on the top of the list. The order of the remaining factors has changed slightly showing that the higher-order effects of those factors vary. However, the total-order indices of those factors are all very close in the range .16 to .12. Hence, we do not identify any factors that contribute insignificantly to the total variance. What is insignificant depends on the concrete context, i.e., how much error in the estimation is acceptable. Thus, introducing an error of more than 10% would be significant for software cost estimations. The high total-order indices indicate a strong interaction of the effects which can be explained by the fact that they are all multiplied in the COCOMO equation.

5 Related work and conclusions

This approach of using global sensitivity analysis for predictor models proved to be useful in the case of COCOMO. Although we are not able to remove factors in order to simplify the model, we still show the factors that need to be estimated best at NASA. In practice this is a valuable information because data collection and analysis is an elaborate process. Hence, we believe such an approach to be useful in real project environments.

We have already applied the approach for similar predictor models. A sensitivity analysis of a cost/benefit model of analytical quality assurance was performed in [7]. We also analysed a reliability growth model [8] to remove factors.

A variety of analyses of COCOMO can be found in the literature [1–3]. However, all these analyses used *local* methods for determining sensitivity that do not have the described useful properties of global sensitivity analysis. Most importantly, the input distributions are not considered as a whole. Hence, these studies also find the factor *size* to be most important but the scale factors are not always identified as important.

References

[1] B. W. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.

[2] Z. Chen, T. Menzies, D. Port, and B. Boehm. Feature subset selection can improve software cost estimation accuracy. In *Proc. 2005 Workshop on Predictor Models in Software Engineering (PROMISE '05)*, pages 1–6. ACM Press, 2005.

[3] P. Musilek, W. Pedrycz, N. Sun, and G. Succi. On the sensitivity of COCOMO II software cost estimation model. In *Proc. Eighth IEEE Symposium on Software Metrics (METRICS'02)*, pages 13–20. IEEE CS Press, 2002.

[4] A. Saltelli and R. Bolado. An alternative way to compute Fourier amplitude sensitivity test (FAST). *Computational Statistics & Data Analysis*, 26(4):445–460, 1998.

[5] A. Saltelli, K. Chan, and E. M. Scott, editors. *Sensitivity Analysis*. John Wiley & Sons, 2000.

[6] J. Sayyad Shirabad and T. Menzies. The PROMISE Repository of Software Engineering Databases. University of Ottawa, 2005.

[7] S. Wagner. A model and sensitivity analysis of the quality economics of defect-detection techniques. In *Proc. International Symposium on Software Testing and Analysis (ISSTA '06)*, pages 73–83. ACM Press, 2006.

[8] S. Wagner. Global sensitivity analysis of predictor models in software engineering. In *Proc. International Workshop on Predictor Models in Software Engineering (PROMISE '07)*. IEEE CS Press, 2007.