Calibrating Use Case Points Using Bayesian Analysis

Kan Qi, Anandi Hira, Elaine Venson, Dr. Barry Boehm
{kqi, a.hira, venson, boehm}@usc.edu
University of Southern California
Outline

Motivations
The Bayesian method for parameter estimation
Data collection for prior and sample information
Model calibration and evaluation
Conclusions
Future directions
Effort and Size Estimation for Use Case Driven Projects

Early and accurate effort estimation is essential for various software project management decisions[1].

• Risk management/project scope control.
• Planning based on estimated schedule.
• Resource allocation.

To achieve the purpose of early estimation of project effort, Karner proposed a use case centric method of measuring software size, which is further used to estimate project effort [2].
Software Sizing Model Based on Use Cases

Use Cases

Use Case Points (UCPs) relies on the software sizing model that use Use Cases to describe software functional size. Use cases model the interactions between actors and a system. Use Cases are weighted by the number of transactions. The sum of the weighted use cases is used to estimate the software size.

\[ \text{Software Size} = \sum_{c \in C} w_c \]

The definition of UCPs also considers the contribution from the actors to software size. Practically use cases contribute most of the software size (\(\geq 90\%\) according to our dataset).
Use Case Points

The counting method of Use Case Points:

1. Identify the use cases and the number of internal transactions for each use case.


3. Calculate the Unadjusted Use Case Weight \((UUCW)\) by:

\[
UUCW = \sum_{c \in C} w(c)
\]

4. Similarly calculate Unadjusted Actor Weight \((UAW)\).

5. Evaluate the 13 technical complexity factors \((TCF)\) and the 8 environmental factors \((EF)\).

6. Calculate Use Case Points \((UCPs)\) by:

\[
UCP = (UUCW + UAW) \times TCF \times EF
\]
The Issues of Use Case Points

The complexity levels of the original UCPs are abrupt [1].

The weighting schema was based on Karner’s working experience at Objectory Systems in 1990’s [2].

Divergence on the experiences of how use cases should be weighted. Different weighting schemas were proposed.

• Extra levels were added to the original use case weighting schema to represent wider ranges.
• Original complexity levels were discretized into more levels.
The Existing Approaches

The existing methods that aim to improve prediction accuracy by adjusting the original weighting schema are in the following three directions:

• Adding extra levels of complexity
  - Re-UCP by Kirmani and Wahid [4]
  - Soft-UCP by Nassif [5]

• Discretizing the complexity levels
  - EUCP by Wang [3]
  - Enhanced UCP by Nassif [7]

• Empirically calibrate the use case complexity weights
  - Neural Network by Nassif [8]
  - Our method
Expert-based vs. Data-based Estimation

The weights are assigned to different use case complexity levels by expert’s domain experience.

• Biased by only taking into consideration specific types of projects. Experts’ estimates tend to be optimistically biased (Hofstadter's Law).
• Outdated by only taking into consideration the projects developed at certain times (productivity was constrained by software development methods at that time).

The weights can also be calibrated empirically with a dataset.

• Not applicable for general use if it is calibrated to a small dataset.
• Usually only small datasets are available for calibration.
  - Data collection is expensive.
  - Project materials collected are heterogeneous: different methods to measure software size and effort.
Expert-based vs. Data-based Estimation Cont’d

The better way is to synthesize the two pieces of information.

- The Bayesian averaging approach provides a framework to update the domain experience with empirically collected information.
- A Bayesian estimator minimize the posterior expected loss.
Bayesian Method of Calibration

The Bayesian method of estimation relies on two pieces of information:

The prior information - the estimates by experts.

- The weights proposed by the domain experts, for which the mean ($w_{a-pri}$) and variance ($\delta_{a-pri}^2$) are calculated.

The sample information - the estimates by empirical analysis.

- The parameters ($w_{regression}$) and variance ($\delta_{regression}^2$) estimated by applying a statistical model to the empirical data set.

The Bayesian estimate is the posterior mean by taking the weighted average of the prior and the sample information.

$$W_{bayes} = \left((\delta_{a-pri}^2 + \delta_{regression}^2)^{-1} \ast (\delta_{regression}^2 \ast w_{a-pri} + \delta_{a-pri}^2 \ast w_{regression})\right)$$

$$\delta_{bayes}^2 = (\delta_{a-pri}^2 \ast \delta_{regression}^2)/(\delta_{a-pri}^2 + \delta_{regression}^2)$$
Bayesian Method of Calibration Example

The Bayesian calibration approach has been adopted in COCOMO II calibration[1].

- to resolve the conflict between the sample estimates and the expert estimates for cost drivers: AEXP, LTEX, FLEX, RUSE, and TEAM.
- prior information: Wideband Delphi; sample information: empirically calibrated parameters.
- succeeded in improving estimation accuracy (more than 16% for PRED(.25)).

We follow the same general structure of using Bayesian analysis to calibrate the weights for use cases.

- update the expert estimates of use case weights with the sample information.
The Proposed Bayesian Approach

1. Calculate a priori means and variances based on domain experts’ proposals as the prior information.
2. Calculate UC complexity weights and their variances using MLR based on the empirical dataset as the sample information.
3. Calculate Bayesian weighted average using the two pieces of information.
The Prior Information - Weights Proposed by Experts

Different weights have been proposed by experts to improve prediction accuracy.

- Extra levels of complexity are added and weighted.
- Original complexity levels are discretized and weighted.

6 existing studies have been identified (including the original Use Case Points).

<table>
<thead>
<tr>
<th>Wght. Schm.</th>
<th>Study</th>
<th>Year</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Wang et al. [3]</td>
<td>2009</td>
<td>EUCP</td>
</tr>
<tr>
<td>6</td>
<td>Nassif et al. [7]</td>
<td>2016</td>
<td>Enhanced UCP</td>
</tr>
</tbody>
</table>
Summary of the Existing Weighting Schemes

The use case complexity weights are plotted in the diagram below as the weights assigned to different numbers of transactions.
The Weights Proposed by Experts & Their Variances

The weights for different levels of use case complexity.

- Simple: 5.29
- Average: 9.00
- Complex: 16.14

<table>
<thead>
<tr>
<th>Level</th>
<th>A-Priori</th>
<th>Regression</th>
<th>Bayesian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>5.29</td>
<td>0.20</td>
<td>4.84</td>
</tr>
<tr>
<td>Average</td>
<td>9.00</td>
<td>1.48</td>
<td>6.41</td>
</tr>
<tr>
<td>Complex</td>
<td>16.14</td>
<td>3.06</td>
<td>19.15</td>
</tr>
</tbody>
</table>
The Sample Information - Weight Calibration by Empirical Study

Data Source:

• 34 student projects from USC’s software and systems engineering center during 2014-2016.

• The projects are of different types: web applications, mobile applications, mobile games, scientific tools, etc.

• The projects: 6-8 people, 4-12 months, 1-20 KSLOC, 1-40 Use Cases.
Weight Calibration by Empirical Study

The calibration process:

- Empirical software size: \( UUCW_{emp} = \frac{Effort_{real}}{EF*TCF*\alpha} - UAW \)

- Multiple linear regression is applied to estimate the effects that different levels of complexity have on software size.

\[
UUCW_{emp} = w_1 \times UC_{simple} + w_2 \times UC_{average} + w_3 \times UC_{complex}
\]

- The variances of the estimated parameters are also estimated.

\[
\theta^2 = (X^TX)^{-1} \times s^2
\]
The Calibration Results

The weights for different levels of complexity.

- Simple: 4.84
- Average: 6.41
- Complex: 19.15

<table>
<thead>
<tr>
<th>Level</th>
<th>A-Priori</th>
<th>Regression</th>
<th>Bayesian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>5.29</td>
<td>0.20</td>
<td>4.84</td>
</tr>
<tr>
<td>Average</td>
<td>9.00</td>
<td>1.48</td>
<td>6.41</td>
</tr>
<tr>
<td>Complex</td>
<td>16.14</td>
<td>3.06</td>
<td>19.15</td>
</tr>
</tbody>
</table>
Estimation with Bayesian Estimation Method

The Bayesian weighted average is taken over the two pieces of information.

• Simple: 5.26
• Average: 8.41
• Complex: 17.32

<table>
<thead>
<tr>
<th>Level</th>
<th>A-Priori</th>
<th>Regression</th>
<th>Bayesian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>5.29</td>
<td>0.20</td>
<td>4.84</td>
</tr>
<tr>
<td>Average</td>
<td>9.00</td>
<td>1.48</td>
<td>6.41</td>
</tr>
<tr>
<td>Complex</td>
<td>16.14</td>
<td>3.06</td>
<td>19.15</td>
</tr>
</tbody>
</table>
Estimation with Bayesian Estimation Method

The properties of Bayesian estimates:

- The weighted averages of the weights proposed by experts and empirically calibrated weights.
- The variances of the Bayesian estimates are smaller than both empirically calibrated weights and the weights estimated by experts.
- The influences from the different use case complexity levels tend to be non-linearly increasing (1:1.6:3.3), instead of the linear relationship (1:2:3) proposed by original UCPs.
Model Evaluation

The performance of Bayesian approach of estimating use case complexity weights is evaluated by comparing it with other typical methods.

- The a priori method (A-Priori).
- The regression method (Regression).
- The original use case point method (Original).
- The Bayesian-based method (Bayesian).
Evaluation Measures

We evaluated the out-of-sample estimation accuracy by 10-fold cross validation in terms of MMRE, PRED(.15), PRED(.25), PRED(.50).

- **MRE:**
  
  \[ MRE = \frac{|y - \hat{y}|}{y} \]

- **MMRE:**
  
  \[ MMRE = \frac{1}{N} \sum_{i=1}^{N} MRE_i \]

- **PRED:**
  
  \[ \text{PRED}(x) = \frac{1}{N} \sum_{i=1}^{N} \left\{ \begin{array}{ll} 1, & \text{if } MRE_i \leq x \\ 0, & \text{otherwise} \end{array} \right. \]

*The low values for MMRE and high values for PRED are desirable.*
The Evaluation Results

The Bayesian based estimator outperforms the other estimators by around 17% in terms of MMRE.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>MMRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian</td>
<td>0.714</td>
</tr>
<tr>
<td>A-Priori</td>
<td>0.882</td>
</tr>
<tr>
<td>Original</td>
<td>0.883</td>
</tr>
<tr>
<td>Regression</td>
<td>0.892</td>
</tr>
</tbody>
</table>
The Evaluation Results - Cont’d

The Bayesian based estimator outperforms the other estimators by more than 13% for PRED(.15), 13% for PRED(.25), and 20% for PRED(.50).

<table>
<thead>
<tr>
<th>Estimator</th>
<th>PRED(.15)</th>
<th>PRED(.25)</th>
<th>PRED(.50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian</td>
<td>0.442</td>
<td>0.492</td>
<td>0.758</td>
</tr>
<tr>
<td>A-Priori</td>
<td>0.308</td>
<td>0.358</td>
<td>0.475</td>
</tr>
<tr>
<td>Original</td>
<td>0.308</td>
<td>0.333</td>
<td>0.558</td>
</tr>
<tr>
<td>Regression</td>
<td>0.217</td>
<td>0.333</td>
<td>0.475</td>
</tr>
</tbody>
</table>
The Evaluation Results - Cont’d

A more comprehensive evaluation of the out-of-sample accuracy (from PRED (0.01) - PRED (0.99)). The Bayesian based estimator consistently outperforms other estimators.

![Graph showing the comparison of different methods]

- Method: Bayesian, A.Priori, Original, Regression
Conclusions

• The framework of synthesizing domain experts’ estimates and empirically calibrated weights is proposed.
• A review of the divergence on use case weights proposed by domain experts is conducted.
• An empirical study on the use case weights have been conducted to calibrate the empirical weights.
• The estimation accuracy using Bayesian analysis is evaluated and the improvement in estimation accuracy is validated by comparing the Bayesian based estimator with other size estimators.
Future Directions

- More data points need to be collected to calibrate the size metric, which can be applied to more general software development environments.
- As more data points collected, the weights assigned to the actors of different complexity levels can also be calibrated.
Thank You!

If you would like to support the research with any software project data (use case driven projects, UML-based projects, or others), please visit our survey page: http://umlx.kanqi.org:8081/surveyproject or contact me at: kqi@usc.edu. Thank you very much!
Q&A
References