Challenges of Data-Driven Cost Estimation in an Industrial Environment

Authors

Name	Affiliation	Email
Jens Heidrich	Fraunhofer IESE Fraunhofer-Platz 1 67663 Kaiserslautern, Germany	jens.heidrich@iese.fraunhofer.de
Adam Trendowicz	Fraunhofer IESE Fraunhofer-Platz 1 67663 Kaiserslautern, Germany	adam.trendowicz@iese.fraunhofer.de
Jürgen Münch	Fraunhofer IESE Fraunhofer-Platz 1 67663 Kaiserslautern, Germany	juergen.muench@iese.fraunhofer.de
Yasushi Ishigai	IPA-SEC 2-28-8 Honkomagome Bunkyo-Ku, Tokyo, 113-6591, Japan	ishigai@ipa.go.jp
Kenyi Yokoyama	IPA-SEC 2-28-8 Honkomagome Bunkyo-Ku, Tokyo, 113-6591, Japan	k-yokoya@ipa.go.jp
Nahomi Kikuchi	IPA-SEC 2-28-8 Honkomagome Bunkyo-Ku, Tokyo, 113-6591, Japan	n-kiku@ipa.go.jp
T. Kawaguchi	Toshiba Information Systems (Japan) Corporation 7-1 Nissin-Cho Kawasaki-City 210-8540, Japan	kawa@tjsys.co.jp

Contact Author

Adam Trendowicz
Fraunhofer Institute for Experimental Software Engineering (IESE)
Fraunhofer-Platz 1
67663 Kaiserslautern, Germany
Tel.: +49 (631) 6800 2137
Fax: +49 (631) 6800 9 2137
Email: adam.trendowicz@iese.fraunhofer.de

Acknowledgement

We would like to thank Toshiba Information Systems (Japan) Corporation, where we conducted the study, as well as all involved experts and local organizers, who greatly contributed to the successful performance of the project. We would also like to thank the Japanese Information-technology Promotion Agency (IPA) for their personnel and financial support in conducting the case study. Finally, we would like to thank Sonnhild Namingha and

Michael Klaes from Fraunhofer IESE for reviewing a first version of the article.

Abstract

The increasing availability of cost-relevant data in industry allows companies to apply data-intensive estimation methods. However, available data are often inconsistent, invalid, or incomplete, so that most of the existing data-intensive estimation methods cannot be applied. Only few estimation methods can deal with imperfect data to a certain extent (e.g., Optimized Set Reduction, OSR®). Results from evaluating these methods in practical environments are rare. This article describes a case study on the application of OSR® at Toshiba Information Systems (Japan) Corporation. An important result of the case study is that estimation accuracy significantly varies with the data sets used and the way of preprocessing these data. The study supports current results in the area of quantitative cost estimation and clearly illustrates typical problems. Experiences, lessons learned, and recommendations with respect to data preprocessing and data-intensive cost estimation in general are presented.

Keywords: Case Study, Cost Estimation, Experience Report, Lessons Learned, Optimized Set Reduction, OSR®

1. Introduction

Reliable software cost estimation is a crucial factor impacting project success. However, many software and system organizations still have significant problems in proposing realistic software costs, work within tight schedules, and finish their projects on schedule and within budget (Standish Group 2003). Considerable research has been directed at gaining a better understanding of the software development processes, and at building and evaluating cost estimation techniques, methods, and tools (Beitz and Wieczorek 2000).

Recently, data-intensive estimation methods (that make intensive use of data to compute estimates) have been gaining more and more interest from both research and industry communities (Trendowicz 2008). One reason is the increasing availability of data in industry that have been collected in a systematic way (e.g., motivated by companies' efforts to reach higher maturity levels). Organizations often want to gain more benefits from that measurement data. In addition, the cost-intensive involvement of experts in the cost estimation process could be reduced if data-intensive methods would reliably support the estimation process. Applying data-intensive estimation models also helps to get more insight on which factors (project characteristics) are cost-related. This could be used for initiating improvement programs that address those factors in future projects.

A major challenge in using data-intensive estimation methods is that the available data sets are typically not suitable for automated data analysis without initial analysis and preprocessing (because data are incomplete, partially invalid, or inconsistent) (Mair et al. 2005). In practice, a common preprocessing strategy is to remove incomplete and inconsistent data items (e.g., the whole case is removed if one attribute value is missing). Alternatively, experts are involved to complete missing data. In consequence, significant parts of measurement data are either not considered at all or (in the best case) completed with subjective estimates. A few estimation methods exist that are able to deal with such imperfect data sets. Such methods usually do not make assumptions regarding data distribution and contain built-in mechanisms to handle missing data and data inconsistencies. However, evaluation results of such methods applied to recent data from industry (e.g., Briand et al. 2000) are quite rare.

The objective of this article is to present empirically-based lessons learned and recommendations for dealing with imperfect industrial data sets for the purpose of cost estimation. The lessons learned and recommendations are derived from a case study that was conducted with Toshiba Information Systems (Japan) Corporation (TJ) in the context of a cooperation project between the Software Engineering Center of the Japanese Information-technology Promotion Agency (IPA-SEC) and the Fraunhofer Institute for Experimental Software Engineering (IESE). The estimation technique Optimized Set Reduction (OSR®, registered trademark of the Fraunhofer Institute for Experimental Software Engineering) was selected for the study, because it copes with numerous practical problems of industrial data sets (Briand et al. 1992). For example, it does not make any assumptions about the distribution of the underlying data, copes with missing data, and can operate on nominal- and continuous-scale project characteristics such as application type and effort, respectively. Moreover, the OSR® algorithm itself is completely automated. Those requirements were explicitly stated by TJ when selecting an appropriate algorithm.

The article is structured as follows: Section 2 introduces the relevant principles of the OSR® method. Section 3 presents the goals and the context of the industrial case study, its execution, analysis, and re-

sults, as well as a discussion of the validity. Section 4 describes lessons learned from the case study with respect to applying OSR® to an industrial data set. Section 5 discusses related work. Section 6 concludes with a list of recommendations for performing data-intensive cost estimation.

2. The OSR® Method

The Optimized Set Reduction (OSR®) method is a pattern recognition method that analyzes trends in software engineering data sets based on machine-learning algorithms. An overview of the basics can be found in Briand et al. (1992). A detailed description of the OSR method can be found in Wieczorek (2001). The idea is to select a subset of similar projects from a project data set as the basis for estimating a certain project attribute such as the overall effort of the project. The original set of project data is iteratively sub-divided into smaller sets, until a certain stop criterion is reached. Each project is described by a set of project characteristics, the so-called independent variables. Based on these independent variables, the subdivision is performed. The final sub-set is described by a Boolean expression, the so-called OSR® model, which is composed of independent variables that were identified as having a great influence on the variable to be estimated. The latter directly depends upon certain characteristics of the project data set and is therefore called dependent variable. The projects included in the set identified by the OSR® model are used to compute an estimate for the dependent variable.

The OSR® algorithm can be used to estimate two different types of dependent variables, namely variables on a continuous scale (e.g., the project's effort or productivity) and on a nominal scale (e.g., defect classes). In the first case, OSR® is used in "regression mode" and in the second case, it is used in "classification mode". The estimation model is dynamically generated for each project that is estimated (and is meant to be optimal for this project) based on the specific characteristics of this project. The algorithm may be used with different parameter settings that influence estimation accuracy. So, different combinations have to be evaluated in order to optimize estimation results for the data set of a specific organization. The basic algorithm is performed using the following steps:

(1) Read parameters: A historical data set that is used as a prediction basis is determined (called Pattern Vector Set, PVS). The project for being estimated is characterized using all independent variables, whereas the values all independent variables are called Measurement Vector MV. After that, the parameter setting is determined depending on whether OSR® is used in classification or regression mode.

(2) Initialize: The set PSS containing historical projects, which are later on used for estimating the current project, is initialized as the whole historical data set PVS. The predicate $Pred_{PSS}$ that describes the current subset PSS as a Boolean expression of independent variables is initialized as well; initially "true", because the whole data set is included in PSS.

(3) Generate singletons: A set of singleton predicates SP is generated. Each singleton in SP consists of an independent variable (a cost factor) and its corresponding value from the measurement vector MV. For each iteration, only those variables of MV are considered that were not already used in previous iterations. Independent variables have to be made discrete beforehand.

(4a) Generate valid predicates: A set PRED of all possible conjunctive predicates is generated from the set SP of available singletons. A predicate is valid if and only if:

- It consists of at most *s* singletons (configuration option).
- It characterizes a subset of PSS that contains more than *o* observations, that is, historical projects.
- It characterizes a subset of PSS that has a distribution of the independent variable that is significantly different from the one implied by PSS. The difference is computed using a binominal test, if OSR® is used in classification mode and a Bootstrap sampling algorithm, if OSR® is used in regression mode.
- It characterizes a subset of PSS that has an improved predictive power compared to the power implied by PSS. The predictive power of a distribution is high if a large number of observations are concentrated on a small number of categories (classification mode) or a small range (regression mode). For classification mode, Shannon's entropy is used as a measure of uncertainty. For regression mode, the mean magnitude of relative error, the mean squared deviance, or the mean absolute deviation may be used depending on the configuration.

(5) Test exit criterion: If PRED is empty, the dependent variable is predicted based on the current subset PSS. PRED is empty if no valid predicate was found.

(6) Extract valid subsets: All subsets of observations characterized by the predicates contained in PRED are extracted from the current set PSS.

(7) Select optimal subset: All extracted subsets are

analyzed with respect to their predictive power and the one having the best power is selected (PSS_{opt}). After that, the corresponding predicate $Pred_{opt}$ is identified that characterizes PSS_{opt}.

(8) Set up new base data set: For initiating the next iteration, PSS is set to current optimal set PSS_{opt} in order to continue searching for optimal subsets in a recursive manner. For each iteration the predictive power of the current subset should improve significantly according to a previously defined significance level α.

(9) Update the final predicate: The current predicate $Pred_{PSS}$ is extended by the current optimal predicate $Pred_{opt}$ using a logical "and". The algorithm proceeds with Step 3, whereas only those variables of MV are considered that were not already used in previous iterations, i.e., that were not collected by $Pred_{PSS}$. So, the final predicate is built in a stepwise manner, concurrent with the generation of the optimal subset.

(10) Predict dependent variable: Based on the final optimal subset, the value of the dependent variable is predicted. In classification mode, the most frequented class is used for computing an estimate. In regression mode, the mean, the trimmed mean, or the median may be used for computing an estimate. The final predicate describes the optimal set, that is, past projects with the best predictive power, and contains all independent variables identifying the most relevant factors for predicting the independent variable.

("Place Figure 1 about here.")

An overview of the overall approach is presented in Figure 1. The steps described above are shown and corresponding parameter settings are assigned. In some cases, a certain parameter implies another one, that is, only a certain combination of parameters makes sense. This is indicated by arrows combining different parameter settings.

OSR® applications on real industrial data sets have shown that estimation accuracy is comparable to other data-intensive techniques, such as Analogy-based, with one of the best standard deviations (Beitz and Wieczorek 2000, Briand and Wieczorek 2002). But, in contrast to other techniques, OSR® was able to produce over 50% more predictions for projects having missing data. The analyzed data sets were provided by the European Space Agency (ESA) and Laturi. The ESA data set was multi-organizational and included 160 projects (90 having complete data) from 4 companies from 1986 to 1998, with 17 characteristics per project. The Laturi data set included 206 software projects from 26 companies from 1986 to 1994, with 8 characteristics. OSR® was able to achieve an estimation accuracy of about 30% Mean Magnitude of Relative Error, including estimates for projects having incomplete data.

Benefits of OSR® include that it is able to work with missing data, provides means for uncertainty evaluation, is able to process continuous and discrete data, is automated, and produces well-interpretable outputs (OSR® models). These models contain several project characteristics and are not necessarily based mainly on a size measure, as is the case for COCOMO, for example. Recently, OSR® was applied to data sets of two Japanese companies, ~80 and ~550 projects from different business units, respectively. The TJ case study applying OSR® and analyzing the outcomes will be described in more detail in the next section.

3. The TJ Case Study

The case study was conducted by Fraunhofer IESE in collaboration with IPA-SEC and Toshiba Information Systems Corporation, Japan (TJ). TJ provided a data base containing project data from different application domains within their organization. IPA-SEC and IESE analyzed the data set in terms of whether it is suited for data-intensive cost estimation. For this purpose, the OSR® algorithm was chosen to compute estimates for all projects and evaluate their quality using a cross-validation approach. The case study was divided into a pre-study and the application phase. The pre-study focused on evaluating the "technical" applicability of OSR® on industrial data provided; e.g., whether data quality allows applying the algorithm. The application phase evaluated the quality of OSR® estimates with respect to the estimation accuracy measured in terms of the Mean Magnitude of Relative Error (MMRE), Mean Squared Deviation (MSD), and Mean Absolute Deviation (MAD) as commonly used measures to evaluate the precision of estimation methods (Conte et al. 1986).

3.1. The OSR® Application Phase

The goal of the pre-study was to define a list of transformation steps that were needed in order to apply OSR® and allow automated data analysis in general. During the pre-study, the OSR® method was successfully applied to an initial data set and produced estimates with an MMRE of 37.01%. The goal of the OSR® application phase was to use the results (i.e., the

list of transformation steps) from the pre-study and to apply the algorithm to an updated data set trying to achieve the best estimation accuracy possible and coming up with a list of lessons learned that will have to be considered when doing data-intensive cost estimation in general and estimation with OSR® in particular. The following issues were especially addressed: (A) Improved data set preparation (syntactically and semantically). (B) Clustering of data. (C) Comparison of OSR® results with standard regression analysis.

Compared to the data set used in the pre-study, TJ updated the data set in order to be able to apply the OSR® tool suite and create OSR® estimates. After that, the independent variables and the dependent variable (the one that will be estimated) were determined. The "normalized performance index", which is a measure for determining a project's productivity, was chosen as the dependent variable, because it was seen as the main project planning criterion by the TJ people. The performance index was measured as function points per person hours. For privacy reasons, the performance index was normalized. This was done by dividing it by the average performance index over all projects analyzed.

Then, the OSR® tool suite was invoked using a direct cross-validation strategy; this means that certain subsets of a project data set were used as test set. For each project in the test set, an effort estimate was calculated using the projects that were not included in the test set. After that, the estimated value was compared with the actual one provided in the project data set. This approach is used for computing the OSR® estimation accuracy by computing the Mean Magnitude of Relative Error (MMRE), the Mean Squared Deviation (MSD), and the Mean Absolute Deviation (MAD). OSR® can be used with different parameters and options (see Briand et al. 2000). In order to find the most suitable ones, the prediction accuracy was computed for different combinations of parameters and options. In order to get an impression of the quality of the OSR® results, we applied a linear regression approach (LRA) to the same test sets based on the "adjusted function points count" (IFPUG 2004). Size was chosen for the regression because it is seen as one of the most popular cost drivers in data-intensive cost estimation. LRA was chosen because of its popularity and simplicity.

3.2. Performing OSR® Analyses

When analyzing a data set with OSR®, the following steps have to be performed: Step 1 (data preparation): In this step, data are pre-processed, so that an OSR® analysis can be conducted. For instance, special characters are replaced, unique column identifiers for project characteristics are introduced, categories for each project characteristic on a nominal scale are analyzed, and unique categories are introduced and mapped to the original categories, if necessary.

Step 2 (data selection): In this step, the projects that will be included in the OSR® analysis as well as independent and dependent variables are determined. For instance, the projects that were outliers in terms of functional size and productivity are excluded from the analysis. Moreover, the projects are randomly assigned to test-sets for cross-validation. After that, some basic statistics are computed, characterizing the data set, such as number of independent variables, number of projects, and ratio of missing data.

Step 3 (OSR® analysis): This step determines the different parameter combinations that are used for the OSR® analysis. After that, the OSR® analysis is conducted accordingly. As mentioned before, OSR® can be invoked with different parameters and options that lead to (slightly) different estimation results. The dependent variable for the TJ case study is on a continuous scale. This implies that several OSR® parameters and options are already fixed. This includes the Classification and Regression Trees (CART, see Breiman et al. 1984) algorithm for discretizing continuous variables and Bootstrap (see Efron and Tibshirani 1993) for computing the difference between distributions. For some parameters, commonly used values were chosen for fine-tuning the algorithm. The significance level was set to 5% and the number of Bootstrap draws was set to 1000. For the prediction function, the objective function, the set size, and the predicate size, 36 parameter combinations were applied for each data subset analyzed (see Figure 2). Four different set sizes were evaluated. For the TJ data set, we set the maximal set size that is evaluated to 20, because it seemed not to be reasonable to include more than a quarter of all projects in one set. With respect to the predicate size, we evaluated three settings. The pre-study had shown that the OSR® models created did not change significantly if more than 4 predicates were added per iteration of the OSR® algorithm.

Step 4 (evaluate results): In this step, the results of the OSR® analysis are evaluated. For each parameter combination, the estimation accuracy (MMRE, MSD, and MAD) is computed. The best parameter combination is identified and compared to linear regression results.

("Place Figure 2 about here.")

3.3. Project Data Set

The TJ data set consists of 78 projects and 82 characteristics per project, excluding the project identifier. As mentioned above, the "normalized performance index" was determined as the dependent variable, which will be estimated by the OSR® algorithm. Furthermore, we reduced the project characteristics that will be considered in the prediction algorithm to 30 selected independent variables. The variables were selected based on the following criteria: (a) Ratio of missing data across all projects. If 90% or more of the project data were missing, the characteristic was not selected. (b) Redundancy of characteristics, e.g., "actual effort", "function point count", and "performance index". (c) Explanatory power regarding the dependent variable. In addition, 14 independent variables were revised and adapted in order to reduce the number of categories and to get unique categories for all variables. The final data set had a missing data ratio of 7.6%.

For the TJ case study, five different subsets were analyzed. Data set "A" removes obvious outliers with respect to functional size and projects that had more than 60% missing characteristics. As presented in Figure 3, four projects were marked as outliers and/or extreme values. Two projects had more than 60% missing characteristics. Those projects were removed from the initial data set in order to get a cleaned starting point for the first OSR® computations. All other data sets analyzed were based on data set "A".

("Place Figure 3 about here.")

Data sets "B" and "C" try to reduce the range of the normalized performance index in order to see whether this would affect estimation accuracy. As shown in Figure 4, no outliers or extreme values could be detected by the analysis, but the overall range of the normalized performance index is quite large (0.2868 to 2.0581). However, 50% of all projects stay within a fairly small interval (0.6743 to 1.253). Data sets "D1" and "D2" were obtained by clustering the remaining data (data set "C") into new and enhancement projects, in order to see whether the development type has an effect on estimation accuracy. Figure 5 presents an overview of the data sets, including the number of projects (#P), the number of characteristics (#C), and the ratio of missing data (MD).

("Place Figure 4 about here.")

("Place Figure 5 about here.")

3.4. Analysis Results

For all analyzed data sets, the OSR® estimation accuracy was compared to a simple linear regression approach (LRA). We applied LRA using exactly the same projects for prediction as OSR®. The LRA estimates were computed based on the adjusted function point value for each project. As already mentioned, OSR® can be used with different parameters and options. Depending on the chosen options, the estimation accuracy can be quite different. This has something to do with the characteristics of the data set. Figure 6 shows the estimation accuracy achieved with OSR® and linear regression. OSR® parameters that produced the best results are listed by referring to the prediction function, the objective function, the minimal set size, and the maximal predicate size. The estimation accuracy was determined using MMRE, MSD, and MAD measures over all estimated projects of the corresponding data subset.

("Place Figure 6 about here.")

In general, OSR® produces much better results than regression for MMRE in any case. However, with respect to the MSD and the MAD, OSR® produced less accurate results for data subset "B". The more outliers and extreme values are removed, the higher the estimation accuracy from data set "A" to "C". For different clusters of projects (new development and enhancement in data sets "D1" and "D2"), quite different estimation accuracies were achieved. For heterogeneous data sets, OSR® produces much better estimates, depending on the variation, e.g., in productivity. The enhancement projects seem to be homogeneous with respect to the remaining characteristics considered in the analysis and thus, regression analysis produced nearly the same results as OSR®. For new development projects, the difference in estimation accuracy was huge. This data set seems to profit the most from OSR®.

Thus, OSR® accuracy (MMRE) improved significantly from data sets "A" to "C" and seems to mainly depend on the project type (see data sets "D1" and "D2"). The chosen OSR® parameters have something to do with the characteristics of the data set. When taking into account all data, including new and enhancement projects, nearly the same parameter combination was used for runs "A", "B", and "C". For new and enhancement projects, different parameter combinations produced the best results. For a certain type of data set, individual cross-validation has to be performed in order to identify the best combination.

3.5. Threats to Validity

The Magnitude of Relative Error (MRE) is one of the most common measures used to evaluate the precision of an estimation method. However, as stated by several researchers, it has several significant limitations when applied to compare estimation methods (Shepperd et al. 2000; Kitchenham et al. 2001; Foss et al. 2003). In the case study presented here, it is mainly used to illustrate the improvement of OSR® when using different data sets. Its expressiveness with respect to comparing LRA and OSR® is limited.

Removing outlier projects and projects having extreme values on productivity makes the data set more homogeneous and naturally produces better estimates. In our case study, linear regression improved, as did OSR®. For other data-driven estimation techniques and other environments, results may be different. However, the identified results give a first indication, even for problems that have to be considered using other data-driven estimation techniques.

4. Lessons Learned

The following section describes basic lessons learned from the case study regarding what has to be considered when doing data-intensive estimation with OSR® using industrial data. Some of them may seem obvious and may hold for many data-driven estimation methods. However, the following lessons learned illustrate those findings from a practical viewpoint.

Size-based Cost Estimation: Considering other project characteristics than size helps to improve estimates and produce promising results, especially when dealing with inhomogeneous data. In our case study, the difference in estimation accuracy was quite large between OSR® and linear regression, which is based solely on size.

Data Collection Process: The quality of the data collection process is essential if data-intensive estimation techniques are to be applied. Consistently specified scales of non-continuous project characteristics, for instance, facilitate reliable automatic analyses. Mapping and preprocessing project characteristics that are used for estimation is required for automated data analysis. If a column contains nominal values, the possible categories are checked and their names corrected. If necessary, the complete column has to be recorded in order to reduce the number of different categories for a certain project characteristic.

OSR® Parameter Selection: Finding the right OSR® parameters is a difficult task, but crucial for achieving good estimates. Currently, this is done in an exploratory approach, where the parameter combination leading to the best estimates is selected. This process is time-consuming, does not consider all parameter combinations and, therefore, does not guarantee optimal results. In the future, more guidance should be provided on how to determine the right parameter settings.

Maintaining Data Sets: When using OSR® for predicting actual projects, it is important to update the estimation data base used regularly, so that estimates do not (solely) depend on old projects. It is also important to detect outliers in the data set in order to improve estimation accuracy. This can be done using conventional mechanisms such as having a look at distributions and identifying outliers and extreme values.

Missing Data: As other studies have shown (Cartwright et al. 2003), the ratio of missing data has to be reduced in order to get good results.

Reducing Scope: Clustering may also improve the accuracy of estimates. For different clusters of projects ("D1" and "D2"), quite different estimation accuracies were achieved in our case study.

Acceptance of Estimation Method: Cost estimation methods shall support a project planner in coming up with a reliable estimate. Therefore, it is important that people applying a cost estimation method can trust the results and interpret them accordingly. In the TJ case, a number of practical problems had to be addressed before the method could be accepted. This included data storage issues, automated tool support, localization issues, proper identification of outliers, and the computation of confidence intervals for estimates.

5. Related Work

Numerous types of estimation methods have been developed over the last decade (Briand and Wieczorek 2002; Trendowicz 2008). They basically differ with respect to the type of data they require and the form of the estimation model they do provide (see Figure 7).

("Place Figure 7 about here.")

With respect to input data, we differentiate between three major groups: data-intensive, expert-based, and hybrid methods (combining quantitative data and expert knowledge). The current trend among software organizations to increase the maturity of their software processes pushes software industry toward quantitative collection of measurement data. The relatively high accuracy and the low application cost of data-intensive

methods presented in related literature (as compared to traditionally applied expert-based methods) are also tempting for commercial software organizations that plan to collect or already do collect quantitative project data. Yet, besides potential benefits, the applicability of a certain method in a specific context has to be considered. In case of *data-intensive methods*, this includes the required quantity and quality of project data. In practice, even if data-collection processes are in place they are typically not based on defined measurement goals and do not follow disciplined measurement process. In consequence even if sufficient amounts of data are collected they often suffer from significant incompleteness and inconsistency. In order to be applicable estimation methods should cope with all those problems. Moreover, in order to be acceptable, they should provide appropriate decision support facilities, e.g., support project risk management.

Memory-based methods (also known as *analogy-based*) such as Case-based reasoning (CBR) (Shepperd and Schofield 1997; Walkerden & Jeffery 1999; Li et al. 2006) implement an estimation mechanism similar to that used by human estimators. Each time a new project needs to be estimation analogy methods take already finished projects that are similar to the estimated one in order to come up with an estimate. Although the estimation process is intuitive of human experts, the rationale behind estimates provided is typically not clear due to lack of explicit model of effort dependencies. Moreover, similarity measures are intolerant of noise and of irrelevant cost factors. This might be partially dealt with by applying additional factor selection techniques (Auer et al. 2006). The Optimized Set Reduction method (OSR®) (Briand et al. 1992) solves those problems by selecting the most relevant factors and building transparent model specific for a new estimated project.

Model-based approaches use past project data to build an explicit estimation model that may be reused without requiring historical data each time estimates are needed. *Parametric* methods require, however, specifying a priori parameters of the estimation model. Machine learning methods such as ANN (Boetticher 2001) require specifying the structure of the network and form of the activation function. Statistical methods such as those based on regression (Briand et al. 1999) or analysis of variance (ANOVA) (Kitchenham 1992) besides functional form of the output model require certain characteristics of the input data being met (e.g., normal distribution). Moreover, statistical methods do not perform well with discontinuous variables, and are very susceptible to the effect of outliers (Shepperd and Kadoda 2001). *Non-parametric* methods such Classification and Regression Trees (CART) (Breiman et al. 1984), or Rule Induction (RI) (Mair and Shepperd 1999), make practically no assumptions about the data and can deal with messy data (Srinivasan and Fisher 1995). Yet, like most of the machine learning methods they require large input data sets and are quite sensitive to their parameter configuration (Shepperd and Schofield 1997). As there is usually little universal guidance regarding how to set their parameters, finding appropriate values requires typically some preliminary experimentation.

Unlike thee *define-your-own-model* approaches which require past project data in order to build customized models, the *fixed-model* approaches provide an already defined model, where factors and their relationships are set. The major advantage of fixed-mode approaches such as COCOMO (Boehm 1981; Boehm et al. 2000) or SLIM (Putnam and Myers 1992) is that they, theoretically, do not require any data from already completed projects. Yet, in practice, fixed models are developed for a specific context and are, by definition, only suited for estimating the types of projects for which the fixed model was built. The applicability of such models for different contexts is usually quite limited and organization-specific project data are required for calibrating the generic model in a specific application context. Moreover, a fixed model may include factors that are irrelevant in a certain context, while excluding others having a significant impact on project cost.

Finally, there are several data-related problems that affect nearly all data-intensive cost estimation methods. Missing data, for instance, is a very common weakness of industrial data sets that has a significant impact on the applicability of the method. Another problem is handling mixed continuous/non-continuous data. The OSR® application as described in this paper explicitly addresses data preprocessing steps in order to deal with missing and messy data.

Another strategy to deal with common problems of data-intensive methods is to complement analytical effort estimation with expert judgment such as, for instance, proposed in the CoBRA® method (Briand et al., 1998).

In summary, software decision makers who decide to use data-intensive estimation methods must face numerous practical problems. Software estimators need support to select and apply data preparation and cost estimation methods. This calls for practical guidelines and reliable field studies regarding the application of such methods in industrial environments, on

up-to-date project data.

6. Conclusions

Data-intensive methods have numerous advantages. Yet, in order to fully benefit from the application of such methods, the data have to have an appropriate quality and quantity. In our study we have applied a data-intensive method, OSR®, which was designed to cope with most of the common problems of industrial data, such as missing data or mixed continuous and discontinuous data. We found that there are still a number of issues to be solved that might affect the quality of predictions when applying data-intensive estimation methods. These issues include:

- Data quality has to be addressed. Before employing automated estimation of project cost, the quality of input measurement data has to be carefully analyzed. Missing data must be resolved and outliers must be detected and removed before acceptable estimation results can be obtained. In addition, consistently defined unified measurement scales are needed for non-continuous factors and the right project characteristics have to be measured, i.e., only those having a significant influence on observed project characteristics such as productivity or cost.
- Data quantity has to be addressed. Data-intensive estimates can support a planner in coming up with a prediction. However, it depends on the number of similar projects how reliable such estimates are. There is also a technical issue related to data quantity. Advanced data-intensive estimation techniques use computationally intensive algorithms, in particular, machine learning approaches. The more data are analyzed, the longer it takes to come up with an estimate.

The results of the case study support current knowledge in the area of quantitative cost estimation and lead to the following recommendations:

- An organization may profit by collecting more than "just" size when doing data-driven estimation. However, it is therefore important to know which factors are important for an organization or a certain part of an organization. Goal-oriented measurement (Basili et al. 2001) can support this process by systematically taking into account influencing factors and coming up with reliable measures.
- As the study indicates, it can help to improve estimation accuracy if estimates are computed for a smaller scope of projects with more homogeneous characteristics, e.g., through data clustering.
- Hybrid data- and expert-based identification of the most significant influencing factors and relationships between them can help to focus the data collection process, improve prediction accuracy, and reduce costs. It would also be possible to develop a kind of causal model as it is done for the CoBRA® method (Briand et al., 1998; Trendowicz et al. 2006) that explicitly describes the interaction between project cost and influencing factors by making use of experts.
- A systematic and restrictive measurement process is needed for doing data-intensive cost estimation. Otherwise, intensive rework and preprocessing is needed before being able to do cost estimation. Moreover, the estimation model needs to be maintained over time. So, model maintenance and validation processes have to be in place on how to update the estimation base with new projects and how to guarantee a certain quality of the estimation model, respectively.
- Data-intensive estimation methods, such as OSR®, usually make use of quite complex algorithms and therefore need tool support. This support should, however, not be restricted to the algorithm itself. When introducing such a method to an organization, it is also important to support the organizational processes around the pure application of the algorithm. This includes aspects like maintaining the estimation base, detecting outliers, and controlling the quality of incoming data syntactically as well as semantically.

Project planning is a human-based process and estimation methods should support project planners and decision makers and not replace them. In that sense, any additional information provided by a method such as estimation uncertainty or information about the estimation model (i.e., influencing factors and their dependencies) can help to understand the estimates obtained and the related software processes. This can be a fundamental factor for the practical usefulness of an estimation method.

References

M. Auer, A. Trendowicz, B. Graser, E. Haunschmid, and S. Biffl, "Optimal Project Feature Weights in Analogy-based Cost Estimation: Improvement and Limitations", IEEE

Transactions on Software Engineering, Vol. 32, No. 2, 2006, pp. 83-92.

V.R. Basili, G. Caldiera, and H.D. Rombach, "Experience Factory", Encyclopedia of Software Engineering, Vol. 1, John Wiley & Sons, 2001, pp. 511-519.

B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Refer, B. Steece, Software Cost Estimation with COCOMO II. Prentice Hall, 2000.

B.W. Boehm, Software Engineering Economics, Prentice-Hall, 1981.

G. Boetticher, "An Assessment of Metric Contribution in the Construction of a Neural Network-Based Effort Estimator", Proceedings of the 2nd International Workshop on Soft Computing Applied to Software Engineering, 2001, pp. 59-65.

International Function Point Users Group (IFPUG), Function Point Counting Practices Manual, Release 4.2, IFPUG, 2004.

L. Breiman, J. Friedman, R. Ohlsen, C. Stone, Classification and Regression Trees, Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

L.C. Briand, T. Langley, and I. Wieczorek, "A Replicated Assessment and Comparison of Common Software Cost Modeling Techniques", Proceedings of the 22nd International Conference on Software Engineering, Limerick, Ireland, 2000, pp. 377-386.

L.C. Briand, K.E. Emam, and I. Wieczorek, "Explaining the Cost of European Space and Military Projects", Proceedings of the 21st International Conference on Software Engineering, Los Angeles, California, USA, 1999, pp. 303-312.

L.C. Briand, K.E. Emam, and F. Bomarius, "COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment," Proceedings of the 20th International Conference on Software Engineering, Kyoto, Japan, 1998, pp. 390-399.

L.C. Briand, V.R. Basili, and W.M. Thomas, "A Pattern Recognition Approach for Software Engineering Data Analysis", IEEE Transactions on Software Engineering, Vol. 18, No. 11, 1992, pp. 931-942.

A. Beitz and I. Wieczorek, Applying Benchmarking to Learn from Best Practice, Fraunhofer Institute for Experimental Software Engineering, Report No. 007.00/E, 2000.

L.C. Briand and I. Wieczorek, "Software Resource Estimation", Encyclopedia of Software Engineering, Vol. 2, John Wiley & Sons, 2002, pp. 1160-1196.

M.H. Cartwright, M.J. Shepperd, and Q. Song, "Dealing with Missing Software Project Data", Proceeding of the 9th International Symposium on Software Metrics, Sydney, Australia, 2003, p. 154.

S.D. Conte, H.E. Dunsmore, V.Y. Shen, Software Engineering Metrics and Models, The Benjamin/Cummings Publishing Company, Inc., 1986.

E. Efron, R. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, 1993.

T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit, "A simulation study of the model evaluation criterion MMRE," IEEE Transactions on Software Engineering, vol. 29, no. 11, November 2003, pp. 985-995.

B.A. Kitchenham, L.M. Pickard, S.G. MacDonell, M.J. Shepperd, "What accuracy statistics really measure." IEEE Software, Vol. 148, No. 3, June 2001, pp. 81–85.

B. Kitchenham, "Empirical Studies of the Assumptions That Underline Software Cost-Estimation Models", Information and Software Technology, Vol. 34, No. 4, 1992, pp. 211-218.

J. Li, G. Ruhe, A. Al-Emran, M.M. Richter, "A Flexible Method for Software Effort Estimation by Analogy," Empirical Software Engineering, vol. 12, no. 1, February 2007, pp. 65-106.

C. Mair, M. Shepperd, and M. Jørgensen, "An analysis of data sets used to train and validate cost prediction systems", Proceedings of the 2005 Workshop on Predictor Models in Software Engineering, St. Louis, Missouri, USA, 2005, pp. 1-6.

C. Mair, M.J. Shepperd, "An Investigation of Rule Induction Based Prediction Systems", Proceeding of the IEEE ICSE Workshop on Empirical Studies of Software Development and Evolution, May 18th, 1999.

L.H. Putnam and W. Myers, Measures for Excellence, Reliable Software on Time, Within Budget, Yourdon Press, Englewood Cliffs, N.J., USA, 1992.

M. Shepperd, G. Kadoda, M. Cartwright, "On Building Prediction Systems for Software Engineers." Empirical Software Engineering, Vol. 5, No. 3, November 2000, pp. 175-182.

M. Shepperd and G. Kadoda, "Comparing Software Prediction Techniques Using Simulation." IEEE Transactions on Software Engineering, Vol. 27, No. 11, 2001, pp. 1014-1022.

M. Shepperd and C. Schofield, "Estimating Software Project Effort Using Analogies." IEEE Transactions on Software Engineering, Vol. 23, No. 12, 1997, pp. 736–43.

K. Srinivasan and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort", IEEE Transactions on Software Engineering, Vol. 21, No. 2, 1995, pp. 126-137.

The Standish Group. CHAOS Chronicles, West Yarmouth, MA, Standish Group International, Inc., 2003.

A. Trendowicz, "Software Effort Estimation – An Overview of Current Industrial Practices and Existing Methods," Tech. Rep. 06.08/E, Fraunhofer IESE, Kaiserslautern, Germany, 2008.

A. Trendowicz, J. Heidrich, J. Münch A, Y. Ishigai, K. Yokoyama, and N. Kikuchi, "Development of a Hybrid Cost

Estimation Model in an Iterative Manner", Proceedings of the 28th International Conference on Software Engineering, Shanghai, China, 2006, pp. 331-340.

F. Walkerden, R. Jeffery, "An Empirical Study on Analogy-based Software Effort Estimation," Empirical Software Engineering, vol. 4, 1999, pp. 135-158.

I. Wieczorek, Improved Software Cost Estimation. A Robust and Interpretable Modeling Method and a Comprehensive Empirical Investigation. PhD Theses in Experimental Software Engineering; Vol. 7, Fraunhofer IRB Verlag, Stuttgart, 2001.

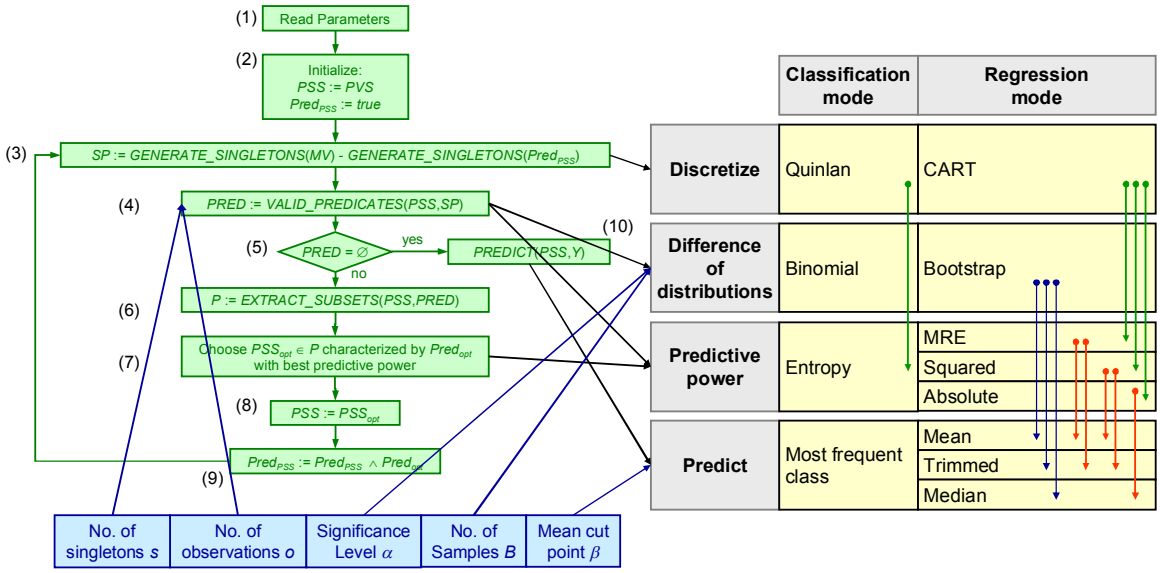

Figure 1: Overview of the OSR® method

Figure 2: OSR® parameter combinations used

Prediction Function	Mean, Median (with MAD only)
Objective Function	MMRE, MSD, MAD
Minimal Set Size	5, 10, 15, 20
Max. Predicate Size	2, 3, 4

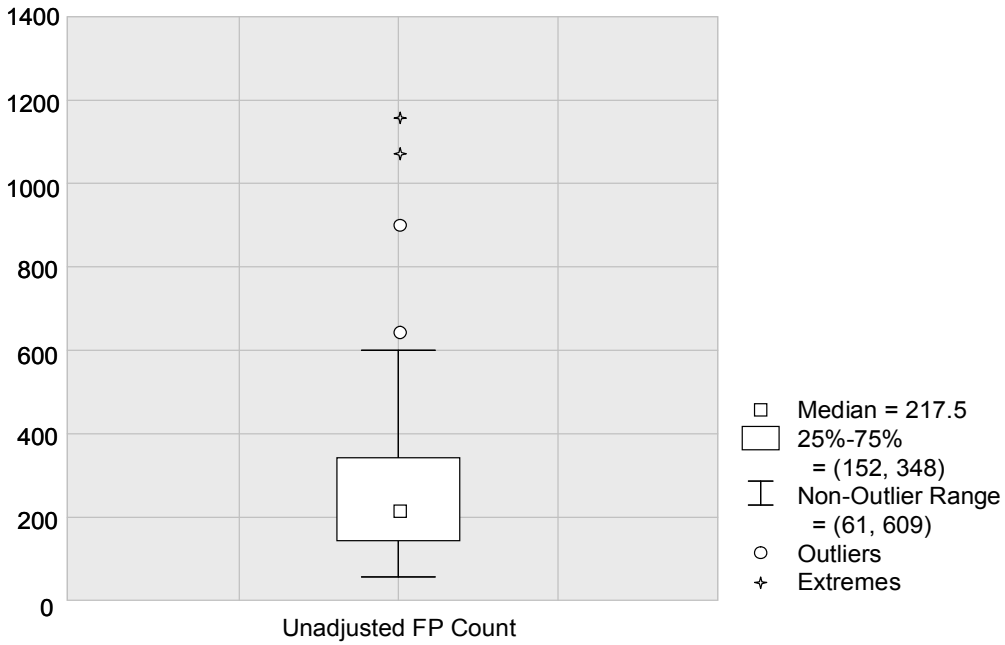

Figure 3: Box plot of the unadjusted function point count

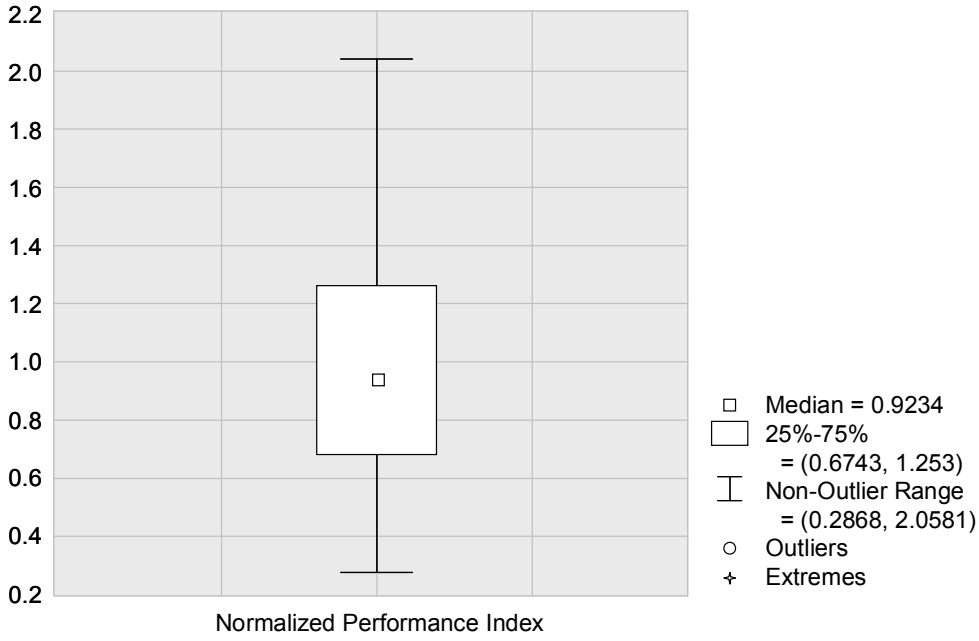

Figure 4: Box plot of the normalized performance index

Figure 5: Data set characteristics

	Data Set Description	*#P*	*#C*	*MD*
A	No functional outliers	72	30	5.56%
B	Reduced productivity extreme values	61	30	5.02%
C	Even more reduced productivity extreme values	58	30	5.23%
D1	New development projects only	36	30	5.64%
D2	Enhancement projects only	22	30	4.55%

Figure 6: Case study results

	OSR® Parameters				*LRA*		*OSR®*
Mean Magnitude of Relative Error							
A	Mean	MSD	10	3	43.12%	>	37.35%
B	Mean	MSD	10	3	30.79%	>	27.19%
C	Mean	MSD	10	2	30.50%	>	24.73%
D1	Median	MAD	5	2	37.04%	>	21.75%
D2	Median	MAD	10	2	31.98%	>	31.15%
Mean Squared Deviation							
A	Mean	MMRE	10	2	0.209	>	0.191
B	Mean	MMRE	20	3	0.120	<	0.122
C	Mean	MMRE	10	2	0.119	>	0.115
D1	Median	MAD	5	2	0.181	>	0.091
D2	Median	MAD	10	2	0.156	>	0.147
Mean Absolute Deviation							
A	Mean	MMRE	10	2	0.367	>	0.358
B	Mean	MMRE	15	2	0.284	<	0.287
C	Mean	MMRE	10	2	0.284	>	0.272
D1	Median	MAD	5	2	0.331	>	0.231
D2	Median	MAD	10	2	0.315	>	0.311

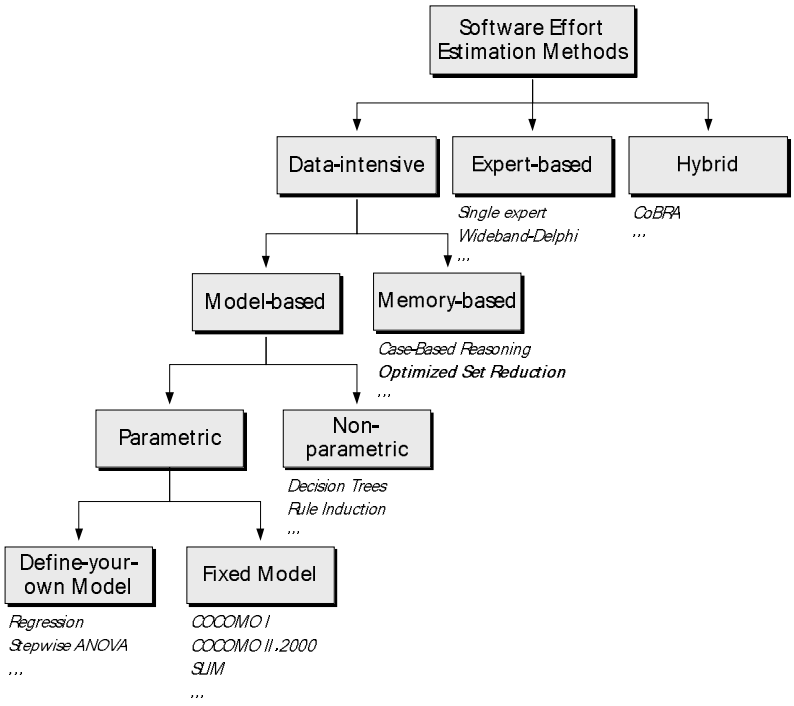

Figure 7: Classification of Software Effort Estimation Methods