An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques

Lionel C. Briand, Khaled El Emam
Dagmar Surmann, Isabella Wieczorek
Fraunhofer Institute for Experimental
Software Engineering IESE
Sauerwiesen 6
67661 Kaiserslautern
Germany
+49 6301 707251
{briand,elemam,surmann,wieczo}@iese.fhg.de

Katrina D. Maxwell
DATAMAX
14 Ave. F-Roosevelt
77210 Avon
France
+ 33-1 6074 0041
datamax@computer.org

ABSTRACT

This paper investigates two essential questions related to data-driven, software cost modeling: (1) What modeling techniques are likely to yield more accurate results when using typical software development cost data? and (2) What are the benefits and drawbacks of using organization-specific data as compared to multi-organization databases? The former question is important in guiding software cost analysts in their choice of the right type of modeling technique, if at all possible. In order to address this issue, we assess and compare a selection of common cost modeling techniques fulfilling a number of important criteria using a large multi-organizational database in the business application domain. Namely, these are: ordinary least squares regression, stepwise ANOVA, CART, and analogy. The latter question is important in order to assess the feasibility of using multi-organization cost databases to build cost models and the benefits gained from local, company-specific data collection and modeling. As a large subset of the data in the multi-company database came from one organization, we were able to investigate this issue by comparing organization-specific models with models based on multi-organization data. Results show that the performances of the modeling techniques considered were not significantly different, with the exception of the analogy-based models which appear to be less accurate. Surprisingly, when using standard cost factors (e.g., COCOMO-like factors, Function Points), organization specific models did not yield better results than generic, multi-organization models.

Keywords
cost estimation, classification and regression trees, analogy, analysis of variance, least-squares regression

1 INTRODUCTION

The estimation of software development cost has been the focus of much research over the past 20 years. Many methods have been applied to explain software development cost as a function of a large number of potentially relevant cost factors. A wealth of modeling techniques coming from statistics, machine learning, and knowledge acquisition have been used with various degrees of success and on a limited number of mostly small and medium-size data sets.

Despite the intense research activity, few generalizable conclusions can be drawn. For example, many of the studies that evaluate modeling techniques consider only a small subset of techniques. In addition, such studies are usually based on small data sets, making the results more vulnerable to the idiosyncrasies of the data at hand. Finally, initial evaluations are rarely replicated, hence one can only have limited confidence in the conclusions drawn.

One purpose of this paper is to address the first two issues noted above. First, we compare the prediction accuracy of most of the more commonly used modeling techniques in the context of cost estimation. Second, we do so with a large data set (by software engineering standards).

In addition, cost modeling can be based on multi-organization data (e.g., COCOMO) and local project data, which are specific to a given organization (e.g., SEL). When companies group together (e.g., European Space Agency database [18]), project cost data can be collected much faster, thus facilitating the construction of cost models and benchmarks. But how do the results compare with cost models based on local, company-specific data? In other words, assuming that data are collected somewhat consistently and projects come from comparable domains, what are the benefits of using local data over a multi-organization database?

These questions are of high practical importance since strategic decisions have to be made by software organizations in order to decide whether sanitized, multi-organization databases within industry sectors are worth developing and whether they can be used for their software cost estimation. Despite inherent data quality assurance and comparability issues, multi-organization databases can support the fast construction of cost models, helping organizations ensure that their cost estimation models keep up with the fast pace of technology evolution.

We can summarize the discussion above as two questions:

1. Which modeling technique performs best (a) if local, company-specific data is available and (b) using multi-organization data?

2. What are the advantages of company-specific data as compared to multi-organization data from the same domain?

These two issues are investigated using a unique database consisting of 206 business software projects from 26 companies in Finland. This database is the result of a rigorous data quality assurance process, contains projects from a similar application domain (i.e., business application systems), and is of considerable size by software standards. Furthermore, a company code identifies projects contributed by the same company, making this a very suitable database on which to assess and compare common software cost estimation models both within and across companies.

The remainder of the paper is organized as follows: Section 2 provides a brief literature review and places this paper in the context of existing research. Section 3 describes and justifies our research method. Section 4 presents the most salient data analysis results. Discussions of the results and conclusions are then presented in Section 5.

2 RELATED WORK

During the last twenty years, many different studies comparing modeling techniques for software cost estimation have been published. In the 1980's, widely used parametric models [3][20][1][2] were compared using data sets of various sizes and environments. Some of the main conclusions were that these models perform poorly when applied uncalibrated to other environments [15][17][10]. Kemerer, for example, used 15 projects from business applications and compared four models: SLIM [20], COCOMO [3], Estimacs [20], and Function Points (FP) [2]. He reported an estimation error in terms of the Mean Magnitude of Relative Error (MMRE) ranging from 85% to 772%. Conte et al. used six data sets from widely differing environments and reported an MMRE variation between 70% and 90% for their three tested models: SLIM [20], COCOMO [3], and Jensen's model [12]. As a result of their investigation, they proposed a new model COPMO [10] calibrated separately to the six data sets. This new model yielded a MMRE of 21%.

In the 1990's, comparative studies also included non-parametric modeling techniques based on machine learning algorithms and analogy. Shepperd et al. [25] compared an analogy-based technique with stepwise regression. They used nine different data sets from different domains and report that in all cases analogy outperforms stepwise regression models in terms of the MMRE. Mukhopadyay et al. [24] used Kemerer's project data set and found that their analogy-based model Estor, using case-based reasoning (CBR), outperformed the COCOMO model. Finnie et al. [11] compared CBR with different regression models using FP and artificial neural networks on a large database consisting of 299 projects from 17 different organizations. They report a better performance of CBR when compared with different regression models based on function points. In addition, artificial neural networks outperformed the CBR approach.

Srinivasan et al. [23] include in their comparison: regression trees, artificial neural networks, function points , the COCOMO model, and the SLIM model. They used the COCOMO data set (63 projects from different applications) as a training set and tested the results on the Kemerer data (15 projects, mainly business applications). The regression trees outperformed the COCOMO and the SLIM model. They also found that artificial neural networks and function point based prediction models outperformed regression trees.

Using a combination of the COCOMO and the Kemerer data sets, Briand et al. [6] compared the COCOMO model, stepwise regression, and Optimized Set Reduction (OSR), which is a non-parametric technique based on machine learning. OSR outperformed stepwise regression and the COCOMO model. Jørgensen [14] used 100 maintenance projects for testing several variations of regression, artificial neural networks, and combinations of OSR with regression. He found that two multiple regression models and a hybrid model combining OSR with regression worked best in terms of accuracy. In general, he recommended the use of more sophisticated prediction models like OSR together with expert estimates to justify the investments in those models.

Although parametric techniques are included in almost all of the studies comparing different cost estimation methods, the comparisons are partial in the sense that only certain techniques are evaluated. Moreover, replications of studies are rarely performed. Even when the same data set is used in different studies, the results are not always comparable because of different experimental designs. Briand et al. and Srinivasan et al., for example, both used the COCOMO and

Kemerer data; however, they used the data in different ways as training and test sets [6][23]. Furthermore, many studies use only small data sets coming from different environments. This makes it difficult to draw generalizable conclusions about the models' performance.

Our current study makes the contribution of evaluating and comparing many of the common cost modeling techniques that have been used in software engineering. In addition, we use both company specific and multi-organizational data, which allows the relative utility of multi-organizational databases to be evaluated.

3 RESEARCH METHOD

Data Set

The Experience Database started in close cooperation with 16 companies in Finland. Companies buy the Experience tool [30] and pay an annual maintenance fee. In return, they receive the tool incorporating the database, new versions of software and updated data sets. Companies can add their own data to the tool and are also given an incentive to donate their data to the shared database through the reduction of the maintenance fee for each project contributed. The validity and comparability of the data is assured as all companies collect data using the same tool and the value of every variable is precisely defined. Moreover, companies providing the data are individually contacted in order to verify and check their submission. At the time of writing, the database consisted of 206 software projects from 26 different companies. The projects are mainly business applications in banking, wholesale/retail, insurance, public administration and manufacturing sectors. This homogeneity in the data allows us to derive more generalizable conclusions for other projects in the business application domain. Six companies provided data from more than 10 projects. As one company submitted a big proportion (one third) of the whole database, we were able to address important issues regarding the usefulness of company specific data as compared to external data. The system size is measured in Experience Function Points, a variation of the Albrecht's Function Point measure [2]. The five functional categories used are the same as Albrecht's; however, complexity weights are measured on a five point scale instead of a three point scale. The variables considered in our analysis are presented in Table 1.

Modeling Techniques

As described in Section 2, many data-intensive modeling techniques have been proposed in the literature. Our comparative study considered only a subset of all of these proposed techniques. The criteria for inclusion were as follows:

Automatable: Since we use a computationally intensive cross-validation approach to calculate the accuracy values, we could only consider techniques that could be substantially automated. This excluded labor intensive approaches such as that suggested in [16].

Variable	Description	Scale	Values / Range / Unit
Effort	Total project effort	ratio	Person hours (ph)
EFP	System size measured in Function Points	ratio	Unadjusted Experience Function Points (EFP)
BRA	Organization Type	nominal	Banking, Wholesale/Retail, Insurance, Manufacturing, Public administration
APP	Application Type	nominal	Customer service, Management information systems, Office information systems, Process control and automation, Network management, Transaction processing, Production control and logistics, On-line and information service
HAR	Target Platform	nominal	Networked, Mainframe, PC, Mini computer, Combined (mf+pc, mini+pc, etc.)
F1–F15	15 Productivity Factors: Customer Participation, Development Environment, Staff Availability, Level and use of Standards, Level and use of Methods, Level and use of Tools, Logical Complexity of the Software, Requirements Volatility, Quality Requirements, Efficiency Requirements, Installation Requirements, Analysis Skills of Staff, Application Experience of Staff, Tool Skills of Staff, Project and Team Skills of Staff	ordinal	1 - 5 (very small – very large)

Table 1: Variables from the Laturi Data base

Applied in Software Engineering: There ought to be a precedent in software engineering where the technique has been used, especially in cost estimation. The rationale is that the technique would have an initial demonstrated utility.

Interpretable: The results of the modeling technique have to be interpretable. For instance, if we identify a modeling technique that produces difficult to interpret results as the best one, this would not be a useful recommendation because in practice project managers would be unlikely to apply a model that is not understandable. This excludes techniques such as Artificial Neural Networks.

Based on the above selection criteria, we considered the following modeling techniques: ordinary least-squares regression (OLS), a standard Analysis of Variance approach for unbalanced data sets, CART, and an analogy-based approach. We also considered combinations of these modeling techniques: CART and OLS regression, and CART and analogy-based approach. Below we describe the modeling techniques that we applied.

Ordinary Least Squares Regression

We used multivariate least squares regression analysis by fitting the data to a specified model that predicts effort [31]. The selected model specification is exponential, because linear models revealed marked heteroscedasticity, violating one assumption for applying regression analysis. A mixed stepwise process was performed to select variables having a significant influence on effort (α=0.05). Dummy variables were created to deal with categorical, nominal scaled variables. Ordinal-scaled variables were treated as if they were measured using an interval scale. This is reasonable, as shown in a study by Spector [22]. He showed that there is practically no difference in using scales having equal or unequal intervals. In addition, Bohrnstedt et al. [4] state that ordinal scales have been shown to be usable as interval when using parametric statistics. Thus, from a practical perspective, these variables are usable as interval covariates in the context of regression analysis.

Stepwise ANOVA

We applied an analysis of variance (ANOVA) procedure for constructing a model to predict effort. This procedure can analyze the variance of unbalanced data and fit regression estimates to models with categorical variables using the Stata tool [28]. It uses the method of least squares to fit linear models. The best models were built using a forward pass method similar to that of Kitchenham [16]. The main differences being that with this ANOVA method the productivity factors are treated as interval variables, the interactions among independent variables are not ignored, and the building of the best one variable, two variable, three variable etc. models is less labor intensive. The final model is of the form:

$$Effort = a \times EFP^{b} \times F_1^{c} \times F_2^{d} \times \ldots$$

where a is a constant which varies with the significant class variables and F is a significant productivity factor (α=0.05). Interaction effects of class variables were taken into consideration.

In order to avoid multicollinearity problems, any two variables with a Spearman rank correlation coefficient exceeding $\pm$0 .75 were considered to be highly correlated and were not used in the same model. Plots of residuals vs. fitted were examined to check for violations of least squares assumptions. In addition, the Ramsay RESET test was used to determine if there were omitted variables, and the Cook-Weisberg test was used to check for heteroscedasticity.

CART

We developed regression tree-based models based on the CART algorithm [3] using the CART (classification and regression trees) tool [29]. A regression tree is a collection of rules of the form: "if (condition 1 and condition 2 and ...) then Z", displayed in the form of a binary tree. The dependent variable for the trees was productivity.

Each node in a regression tree specifies a condition based on one of the project variables that have an influence on productivity. Each branch corresponds to possible values of this variable. Regression trees can deal with variables measured on different scale types.

Building a regression tree involves recursively splitting the data set until (binary recursive partitioning) a stopping criterion is satisfied. The splitting criterion used is the split which most successfully separates the projects' productivity values. The stopping criterion was set to a minimum of twenty observations for the one but terminal node. We selected optimal trees having an overall minimal absolute deviation between the actual and the predicted productivity, and having enough (ten) observations in each terminal node. The median productivity values (instead of the means) for all projects in a terminal node are used as predicted values to account for outliers.

A project can be classified by starting at the root node of the tree and selecting a branch to follow based on the project's specific variable values. One moves down the tree until a terminal node is reached. At the terminal node, effort is calculated by dividing the actual size in EFP by the median productivity within the terminal node.

Combination of CART with Regression Analysis

We combined CART with ordinary least-squares regression. This involves the development of a regression tree, and the application of regression analysis to projects belonging to each terminal node. Thus, for each terminal node in a tree, we developed regression equations for predicting effort, instead of just using median values for prediction.

We also used a simple univariate regression on the terminal nodes by fitting an exponential relationship between effort and system size on the observations in each terminal node. Each project's effort is determined by following down the tree to a terminal node and then applying the appropriate effort equation corresponding to this terminal node.

Analogy-Based Estimation

The basic idea of the analogy-based estimation is to identify the completed projects that are the most similar to a new project. Major issues are: the selection of appropriate similarity/distance functions, the selection of relevant project attributes (in our case cost-drivers), and the decision about the number of similar projects to retrieve (analogies).

We used an analogy-based approach similar to the one applied by Shepperd et al. [25][26] and implemented in the ANGEL tool [32]. We implemented analogy-based estimation using CBR-Works 4.0 beta, a case-based reasoning tool [9]. CBR-Works is flexible to the definition of similarity functions and extensions such as cross-validation. We applied a distance function identical to the one used in the ANGEL tool [32]. This function is based on the unweighted Euclidean distance using variables normalized between 0 and 1. The overall *distance(P_i, P_j)* between two projects P_i and P_j is defined as:

$$distance(P_i,P_j)=\sqrt{\frac{\sum_{k=1}^{n}\delta(P_{ik},P_{jk})}{n}}$$

where n is the number of variables. The distance regarding a given variable k between two projects P_i and P_j is $\delta(P_{ik}, P_{jk})$:

$$\delta(P_{ik},P_{jk})=\begin{cases}\left(\dfrac{|P_{ik}-P_{jk}|}{max_k-min_k}\right)^2, & if\ k\ is\ continuous\\ 0, & if\ k\ is\ categorical\ AND\ P_{ik}=P_{jk}\\ 1, & if\ k\ is\ categorical\ AND\ P_{ik}\neq P_{jk}\end{cases}$$

where value max_k/min_k is the maximum/minimum possible value of variable k.

The ANGEL tool [32] determines the optimal combination of variables by implementing a comprehensive search. This is, however, inefficient for a high number of variables and projects, as reported in [26][27]. In our case, with 19 variables and 119 projects, the computational time required for a comprehensive search to determine the optimal combination of variables would be prohibitive (5.24×10^5). Therefore, we used another strategy proposed by Finnie et al. [11]. We applied to all categorical variables a two tailed t-test to determine variables that show significant influence on productivity. We generated two levels for each variable by merging the variable's original levels.

For effort prediction, we used both the most similar project and the unweighted average of the two most similar projects. These choices were used for the sake of simplicity and are justified, since Shepperd et al. [26][27] report nearly equivalent accuracy when using more than two similar projects or using weighted averages.

Following Shepperd's et al. methodology, we used effort as a dependent variable. As an alternative, productivity was recently proposed as a dependent variable and results are reported in [8]. When using productivity, the predicted effort is calculated by dividing the actual size by the predicted productivity. This version of the analogy algorithm adjusts the effort prediction for system size [33]. Although we attempted to use productivity as a dependent variable and a noticeable improvement could be observed, we do not further discuss it in this paper since our overall conclusions are simply not affected by it.

Combination of CART with Analogy-Based Estimation

CART was combined with the analogy-based approach by developing a regression tree and applying analogy to projects that belong to one terminal node.

Each project's effort is determined by following down the tree to a terminal node and by selecting similar projects within the project subset that corresponds to this terminal node.

Evaluation Criteria

A common criterion for the evaluation of cost estimation models is the Magnitude of Relative Error (MRE) [10]. This is defined as:

$$MRE_i=\frac{|Actual\ Effort_i-Predicted\ Effort_i|}{Actual\ Effort_i}$$

The MRE value is calculated for each observation i whose effort is predicted. The aggregation of MRE over multiple observations, say N, can be achieved through the Mean MRE (MMRE):

$$MMRE=\frac{1}{N}\sum_i\frac{|Actual\ Effort_i-Predicted\ Effort_i|}{Actual\ Effort_i}$$

However, the MMRE is sensitive to individual predictions with excessively large MREs. Therefore, an aggregate measure less sensitive to extreme values should also be considered, namely the median of MRE values for the N observations (MdMRE).

An implicit assumption in using MRE as a measure of predictive accuracy is that the error is proportional to the size of the project. For example, a 10 man-month overestimate for a 10 man-month project is more serious than for a 100 man-month project.

A complementary criterion that is commonly used is the prediction at level l, $PRED(l)=\frac{k}{N}$, where k is the number of observations where MRE is less than or equal to l.

Thus the criteria we used to assess and compare cost estimation models are the relative values of MMRE, MdMRE, and PRED for the different techniques.

Cross Validation

If one constructs a cost estimation model using a particular data set, and then computes the accuracy of the model using the same data set, the accuracy evaluation will be optimistic (i.e., the error will be artificially low, and does not reflect the performance of the model on another unseen data set) [31]. A cross-validation approach gives more realistic accuracy measures. The cross-validation approach we use involves dividing the whole data set into multiple train and test sets, calculating the accuracy for each test set, and then aggregating the accuracy across all the test sets.

We used two different types of train/test splits that are congruent with the questions we posed in Section 1. To determine the accuracy and benefits or drawbacks of generic cost models, we selected subsets of projects that come from a single organization as test sets. We limited this to organizations for which there are 10 or more projects in our data set (in total 119 observations). In such a case, the training set is the whole data set minus that organization's projects. This resulted in six different test sets. Calculating accuracy in this manner indicates the accuracy of using an external multi-organizational data set for building a cost estimation model, and then testing it on an organization's projects.

To determine the accuracy and the benefits of deriving local cost estimation models we used a subset of 63 projects coming from a single organization. Again we used a six-fold cross-validation approach. However, in this case we randomly constructed six test sets, and for each test set we used the remaining projects as the training set. The overall accuracy is aggregated across all six test sets. Calculating accuracy in this manner indicates the accuracy to be expected if an organization builds a model using its own data set, and then uses that model to predict the cost of new projects.

4 DATA ANALYSIS RESULTS

We present in this section the main results of our analysis. Starting with descriptive statistics, we summarize the most important variable distributions. We continue with the results of the comparison of the modeling techniques for the whole database and for the single organization that provided 63 projects. Finally, we discuss the variables selected as important cost-drivers by the different techniques. The models that were generated are available in [8].

Descriptive Statistics

Table 2 summarizes the descriptive statistics for system size (EFP) and project effort (person-hours: ph). The table shows the results for the whole database and for the company that submitted 63 projects. On average, projects coming from this company have a higher effort than projects in the remainder of the database. The company operates in the banking domain and we refer to this part of the database as bank data. The breakdown of projects per organization type, for the whole data base is 38% banking, 27% insurance, 19% manufacturing, 9% wholesale, and 7% public administration. Figure 1 and Figure 2 illustrate the proportions of projects for different application types and target platforms. Each of the histograms compares the proportions of the whole database with the data coming from the bank. The proportions regarding application type and target platform for the whole database and the bank data are very similar. This will help our interpretation of the results in the remainder of the paper.

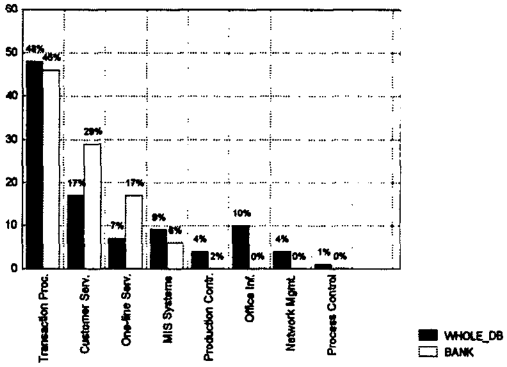

Figure 1: Distribution of projects by application type

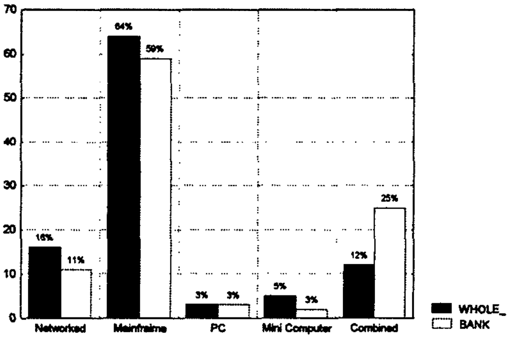

Figure 2: Distribution of projects by target platform

	Bank Data		Whole DB	
	Size (EFP)	Effort (ph)	Size (EFP)	Effort (ph)
min	48	583	48	480
mean	671.4	8109.54	718.7	8352.52
max	3634	63694	3634	63694
st.dev	777.3	10453.9	679.69	9707.23
obs.	63	63	119	119

Table 2: Descriptive Statistics for system size and effort

Comparison of Modeling Techniques

Table 3 and Table 4 show the cross-validation results of comparing the modeling techniques in terms of MMRE, MdMRE, and Pred(.25) for the whole data set and for the bank data, respectively. The values in the tables are averages across the hold-out samples. In Table 3 the hold-out samples (test sets) are the six companies with more than 10 projects. In Table 4 the hold-out samples are randomly selected samples. Some of the modeling techniques are abbreviated in the tables: "CART+Regr." means a combination of CART with univariate regression in the regression tree leaves. "Analogy-1s" is the analogy-based approach using the most similar project to predict effort. "Analogy-2s" is the analogy-based approach involving the two most similar projects for effort prediction. "CART+Analogy-1s" and "CART+Analogy-2s" are a combination of CART with Analogy-1s and Analogy-2s, respectively.

Considering the whole database (Table 3), we observe that, on average, the techniques not involving analogy (i.e., stepwise regression, stepwise ANOVA, CART, CART+Regr.) outperform the techniques involving analogy (i.e., Analogy-1s, Analogy-2s, CART+Analogy-1s, CART+Analogy-2s). Techniques involving the analogy-based approach showed up to a 22% higher MdMRE than CART, the technique yielding the best results. The statistical significance of these results are confirmed when using the matched-pair Wilcoxon signed rank test [12] (a non-parametric analog to the t-test) to compare the MRE's yielded by the various models. However, there is no statistically significant difference in the MREs among analogy-based techniques. The lowest accuracy of analogy-based models is in sharp contrast to what was found in other comparative studies using different data sets [11][24]. Since the specifics of our analogy procedure are very similar to Shepperd et al., such results trigger the need for further studies. One possible reason why techniques involving analogy are less accurate is the way similarity between projects is defined. All variables considered in the similarity function on which the analogy procedure is based have equal influence on the selection of the most similar project(s).

Looking at techniques not involving analogy, simple CART models seem to perform slightly better than regression, stepwise ANOVA, or CART in combination with regression. CART has the lowest MdMRE, MMRE and the highest Pred(.25). *But the differences among these* techniques are not practically significant. The difference in MdMRE remains below or equal to 5%. The MMRE values for these techniques are also quite similar. Moreover, the differences in the MRE values are not statistically significant.

The results above involved the whole database. When looking at the bank data set (Table 4), we can observe that, on average, all techniques show similar accuracy. There is little variation in the MdMRE values (up to 7%) and there is no statistically significant difference among the techniques' MRE's. Thus, in the bank data context, analogy-based techniques performed better than in the multi-organization case. This might be explained by a *higher homogeneity in the projects considered,* a factor to which analogy might be more sensitive than other models. Considering that the bank data set shows much lower variance in productivity, the selection of an inadequate most-similar project will very likely have lesser consequences in terms of prediction than in the context of the whole database.

Overall, the results show that simpler modeling techniques such as CART perform at least as well as more complex techniques. In addition, the results suggest that the key solution to achieve accurate cost predictions does not lie in the modeling technique itself, but more in the quality and adequacy of the data collection. The generic factors considered in our analysis, although several of them are shown to be useful predictors, do not explain a large part of the effort variation in the data set. This in turn suggests that companies may need to devise their own important cost

	Stepwise Regr.	Stepwise ANOVA	CART	CART+ Regr.	Analogy -1s	Analogy -2s	CART+ Analogy- 1s	CART+ Analogy -2s
MMRE	0.53	0.567	0.52	0.52	1.16	1.34	1.25	1.39
MdMRE	0.44	0.446	0.39	0.42	0.61	0.57	0.57	0.602
Pred(.25)	31%	34%	34%	34%	24%	17%	24%	16%

Table 3: Average results over all hold-out samples using the entire database

	Stepwise Regr.	Stepwise ANOVA	CART	CART+ Regr.	Analogy -1s	Analogy -2s	CART+ Analogy- 1s	CART+ Analogy -2s
MMRE	0.67	0.787	0.569	0.655	0.71	0.75	0.81	0.73
MdMRE	0.41	0.424	0.462	0.471	0.48	0.47	0.43	0.47
Pred(.25)	22%	22%	29%	25%	14%	21%	22%	19%

Table 3: Average results over all hold-out samples using the bank data

factors to achieve acceptable MRE levels [7]. Standard cost factors may not be enough. The MRE results shown here by all the models are, from a practical perspective, far from satisfactory for cost estimation purposes, despite the large number of factors collected and the rigor of data collection procedures.

Comparison of local Models versus multi-organization Models

The results in Table 5 are based on using the remaining project data in the Experience database and applying the models to the bank data. This emulates a situation where a company (bank) has only external, multi-organization data available to build a cost estimation model. However, it is important to point out that this data come from similar types of projects (MIS) and show similar distributions in terms of application domains and target platform (see Section 4). We compare the results in Table 5 regarding the bank data with the average results from Table 4, that are solely based on the company's local data. These latter results emulate the situation where a company has its own data available to build a cost estimation model. We can then observe that all techniques show similar MRE's across Table 4 and Table 5. The largest difference in MdMRE (13%) can be observed for Analogy-1s (MdMRE: 0.48 vs. MdMRE:0.61). But, none of the differences are statistically significant. Thus, from this analysis alone, it would appear that there is no advantage to developing company-specific effort estimation models using generic cost factors and sizing measures. One explanation is that the distribution of projects in terms of application domains (APP) and target platforms (HAR) is similar for the one organization data set has and the whole database. Furthermore, these variables (APP and HAR) were identified as important cost-drivers by almost all compared techniques. This implies that projects from the bank might be similar in nature to the remainder of the database.

	Stepwise Regr.	Stepwise ANOVA	CART	CART+ Regr.	Analogy -1s	Analogy -2s	CART+ Analogy -1s	CART+ Analogy -2s
MMRE	0.57	0.629	0.54	0.524	1.32	1.41	1.48	1.46
MdMRE	0.47	0.494	0.46	0.46	0.61	0.56	0.56	0.605
Pred(.25)	25%	28%	27%	29%	21%	14%	24%	16%

Table 4: Results on the bank data using the entire database

From a general perspective, if such results were to be confirmed by subsequent studies, it would have serious implications on the way we collect cost data and build data-driven cost models. To really benefit from collecting organization-specific cost data, one should not just automatically collect generic, COCOMO-like factors, but investigate the important factors in the organization to be considered and design a tailored, specific measurement program [7]. Another implication from our results is that it might very well be possible that multi-organization databases, within specific application domains and using rigorous quality assurance for the data collection, yield cost modeling results comparable to local cost models. In addition, multi-organization databases have the advantage of offering participating companies larger, more up-to-date project data sets.

Selected Model Variables

In Table 5, we summarize the frequency of selection of independent variables across model types. We consider here the models which are based on the whole database. This should give us some insight into the importance of the various independent variables with respect to cost prediction.

Since we generated six models (6-fold cross validation) for each modeling technique, we provide for each variable the number $n \leq 6$ of times a variable v was selected (indicated through pairs: "*v n*", see Table 5). Beside the most important cost-driver which is system size (EFP), organization type (BRA) and the target platform (HAR) are identified as the most important influential factors on cost by all modeling techniques. In case of stepwise ANOVA and when stepwise regression is applied, requirements volatility (F8) is selected as being another important cost-driver. Multivariate regression, ANOVA, and the analogy-based approach identified quality requirements (F9) as important in most of the constructed models.

Stepwise Regr.	EFP 6, BRA 6, HAR 6, APP 1, F5 1 F7 3, F8 6, F9 3, F11 1
Stepw. ANOVA	EFP 6, BRA 6, HAR 6, F7 1, F8 6, F9 5
CART[1]	BRA 6, HAR 4
CART+Regr	EFP 6, BRA 6, HAR 4
Analogy[2]	EFP 6, BRA 6, HAR 6, APP 6, F4 6, F9 6, F10 6, F11 6, F13 6, F14 6

Table 5: Selected variables in the models

[1] Note that the CART only models had productivity as the dependent variable. Therefore, size in EFP does not appear in the selected independent variable list.

[2] Any Analogy variants and all combinations of Analogy with CART.

5 CONCLUSIONS

In this study, we investigated two essential questions related to data-driven, software cost modeling. Firstly, what modeling techniques are likely to yield more accurate results when using typical software development cost data? And secondly, what are the benefits of using organization specific data as compared to multi-organization databases?

In addressing the first question, our results show that the considered modeling approaches do not show large differences according to the three standard evaluation criteria used in this study to estimate the prediction accuracy of cost models, (i.e., MMRE, MdMRE, and Pred(.25)) when applied to data coming from one company. Simple CART models perform a little better than other modeling approaches, which are in most cases more complex; however, the observed differences are not statistically significant. Therefore, the main interest of the CART models resides in their simplicity of use and interpretation.

Results for the multi-company database indicate that models using analogy-based procedures do not perform well when a cross-validation procedure using project data from individual companies as test samples is undertaken. The other models perform more or less equivalent as for the one-company data. Thus, analogy-based models do not seem as robust when using data external to the organization for which the model is built. One explanation is that the higher variability in the data makes the Euclidean similarity functions (used in the analogy approach) less suitable. However, the construction of specific similarity functions (involving weights on the project variables) based on expert opinion should help improve the applicability of analogy models and should be the focus of further research. Another reason is that the prediction of a project relies on one or two projects selected as the most similar ones (i.e., analogues). Thus, the results are likely to be more affected by higher variability in the data (i.e. outliers).

Overall, whether local or generic models are built, simple CART models seem to be a good alternative, both from an accuracy and interpretability point of view

In addressing the second question, we found that local models developed using the one-company database do not perform significantly better than the models developed using external multi-organization data, when applied to the one-company data set. (Not considering analogy-based models for the reasons discussed above). One would expect local models to perform better because of the higher homogeneity of the underlying data set and the fact that the projects considered for modeling are more specific to the organization. However, our results suggest that, when using data from projects belonging to similar application domains and when the data collection quality assurance is of high quality, homogeneity within the projects of one organization may not be higher than across organizations. One explanation is that the main source of heterogeneity may come from the project characteristics themselves rather than the organization where they take place. If such a result is confirmed, this should have important consequences on the strategies adopted by software development organizations to construct cost models and benchmarks. In this case, common project data repositories across companies and within homogeneous application domains should be considered as not only viable but also as a highly beneficial alternative. On the other hand, more specific measurement and data collection procedures which take into account the specificity of projects within an organization should help improve the accuracy of cost models based on local data. The standard cost factors used in this study might not be optimal for the one organization where we constructed a specific cost model.

In general the MRE results shown here are far from being satisfactory for cost estimation purposes. Despite the large number of cost factors collected and the rigourous data collection, a lot of uncertainty in the estimates can be observed. This supports the need for risk analysis to be an integrated part of software cost estimation procedures [7].

ACKNOWLEDGEMENTS

We would like to thank Risto Nevalainen and Pekka Forselius of Software Technology Transfer Finland (STTF) for defining, collecting and providing the data.

We would also like to thank Arthur Harutyunyan for his help on the data analysis reported in this paper.

REFERENCES

1. Albrecht, A.J. Measuring application development productivity. In: *SHARE/GUIDE: Proceedings of the IBM Applications Development Sysmosium,* (October 1979) 83-92.
2. Albrecht, A.J., Gaffney, J.E., Software function, source lines of code, and development effort prediction, *IEEE Transactions on Software Engineering,* vol. 9, no. 6 (1983) 639-648.
3. Boehm, B. Software Engineering Economics. *Englewood Cliffs, NJ* Prentice-Hall (1981).
4. Bohrnstedt G., Carter, T. Robustness in Regression Analysis. In: *Costner, H. (ed). Chapter 5, Sociological Methodology. Jossey-Bass* (1971).
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. Classification and Regression Trees. *Wadsworth & Books/Cole Advanced Books & Software* (1984).
6. Briand, L.C., Basili, V.R., Thomas, W.M. A pattern recognition approach for software engineering data analysis. *IEEE Transactions on Software Engineering, vol. 18, no. 11* (1992) 931-942.

7. Briand, L.C., El Emam K., Bomarius, F., A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. *Proceedings of the 20th International Conference on Software Engineering, ICSE-20* (April 1998) 390-399.

8. Briand, L.C. El Emam K., Maxwell, K., Surmann, D., Wieczorek, I. An Assessment and Comparison of Common Software Cost Estimation Models. *Technical Report, ISERN TR-98-27, International Software Engineering Research Network* (1998).

9. CBR-Works, 4.0 beta. Research group "Artificial Intelligence- Knowledge-Based Systems", *University of Kaiserslautern* <http://wwwagr.informatik.uni-kl.de/~lsa/CBRatUKL.html>

10. Conte, S.D., Dunsmore, H.E., Shen, V. Y. Software engineering metrics and models. *The Benjamin/Cummings Publishing Company, Inc.* (1986).

11. Finnie, G. R., Wittig, G. E. A comparison of software effort estimation techniques: using function points with neural networks, case based reasoning and regression models. *J. Systems Software vol. 39* (1997) 281-289.

12. Gibbons, J.D. S. Nonparametric Statistics. *Series: Quantitative Application in the Social Sciences* 90, SAGE University Paper (1993).

13. Jensen, R.W. A comparison of the Jensen and COCOMO schedule and cost estimation models. *Proceedings of International Society of Parametric Analysis* (1984), 96-106.

14. Jørgensen, M. Experience with the accuracy of Software Maintenance Task Effort Prediciton Models. *IEEE Transactions on Software Engineering*, vol. 21, no.8 (August, 1995) 674-681.

15. Kemerer, C.F. An empirical validation of software cost estimation models. *Communications of the ACM* vol. 30, no. 5 (May 1987) 416-429.

16. Kitchenham, B., A Procedure for Analyzing Unbalanced Datasets, *IEEE Transactions on Software Engineering, vol.24, no.4* (April 1998) 278-301.

17. Kitchenham, B.A., Taylor, N. R. Software project development cost estimation. *The Journal of Systems and Software vol.* 5 (1985) 267- 278.

18. Maxwell, K., Van Wassenhove, L. and Dutta, S. Software Development Productivity of European Space, Military and Industrial Applications. *IEEE Transactions on Software Engineering, vol. 22 no. 10* (1996).

19. Nevalainen, R., Maki, H. Laturi System Product Manual, Laturi 2.0. Finnland (January1996).

20. Putnam, L.H. A general empirical solution to the macro software sizing and estimation problem. *IEEE Transactions on Software Engineering, vol.4, no. 4* (July 1978) 345-381.

21. Rubin, H.A. A comparison of cost estimation tools (a panel discussion). *Proceedings of the 8th International Conference on Software engineering*, IEEE Computer Society Press (1985) 174-180.

22. Spector, P. Ratings of Equal and unequal Response Choice Intervals. *The Journal of Social Psychology, vol. 112* (1980) 115-119.

23. Srinivasan, K., Fisher, D. Machine learning approaches to estimating software development effort. *IEEE Transactions on Software Engineering, vol. 21, no. 2* (February 1995) 126-137.

24. Mukhopadhyay, T., Vicinanza, S.S., Prietula, M.J. Examining the feasibility of a case-based reasoning model for software effort estimation. *MIS Quarterly* (June 1992) 155-171.

25. Shepperd, M., Schofield, C. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering,* vol. 23, no. 12 (November 1997) 736-743.

26. Shepperd, M., Schofield, C., Kitchenham, B .A. Effort estimation using analogy. *Proceedings of the 18th International Conference on Software Engineering, ICSE-18* (1996) 170-175.

27. Shepperd, M., Schofield, C. Effort Estimation by Analogy: A Case Study. *Presented at ESCOM 7,* (Wilmslow 1996).

28. StataCorp, Stata Statistical Software: Release 5.0. *Stata Corporation, College Station,* (Texas 1997).

29. Steinberg, D., Colla, P. CART, Classification and Regression Trees, Tree Structured Nonparametric Data Analysis, Interface Documentation. *Saflord Systems* (1995) <http://www.salford-systems.com/index.html>

30. <http://www.sttf.fi/index.html>

31. Hayes W. Statistics. *Fifth Edition, Hartcourt Brace College Publishers* (1994).

32. <http://dec.bournemouth.ac.uk/dec_ind/decind22/web/Angel.html>

33. Walkerden F. , Jeffery R., An Empirical Study of Analogy-based Software Effort Estimation, Technical Report 98/8, The University of New South Wales, Center for Advanced Empirical Research, Australia,