Analogy Based Cost Estimation Configuration with Rules[1]

Stamatia BIBI[2], Ioannis STAMELOS
Department of Informatics, Aristotle University of Thessaloniki, Greece

Abstract. Analogy-based estimation is a widely adopted method in software cost estimation that identifies analogous projects to the one under estimation and uses their data to derive an estimate, i.e. it is a Case Based Reasoning approach. The similarity measures between pairs of projects are critical for identifying the most appropriate historical data from which the estimation will be generated. Usually the similarity measures are selected empirically, using jackknife-like procedures. Typically, the measures that identify the most similar projects in most of the cases are considered the most appropriate ones and are applied in every new estimation procedure. However there are situations that the default similarity measures may not be the most appropriate ones. In this study we determine the situations in which the default parameters are not the best and we propose the similarity measures for these cases. In particular we provide rules that point out which projects are not accurately estimated with the default parameters.

Keywords. Analogy, Software Cost estimation, Case based reasoning, Rules,

1. Introduction

Software cost estimation is the process of predicting the amount of effort required to develop a software system. Usually the estimation is performed before the initialization of a project and is utilized throughout all software lifecycle, determining the feasibility of a project, the project plan, the allocation of resources and finally the project progress. Accurate and consistent estimates are fundamental to several success-critical project factors, justifying the existence of a variety of estimation models [4], [5],[6], [14].

It is common practice in almost all automated estimation models to use past historical data of already completed projects to predict future ones. One of the most common methods that utilizes past project data is Estimation based on Analogy (EBA) [12]. EBA identifies one or more historical projects that are similar to the project being developed and, based on the data of these projects, derives an estimate. Frequent application of EBA [9] in software cost estimation has indicated certain advantages of the method. EBA can be applied in the early stages of software development when few

[1] This work has been funded by the Greek Secretary of Research and Technology, action 4.3.1, DIERGASIA project.

[2] Corresponding author

data are available and it produces results, easily interpreted by software managers that opposed to other formal models remain unaffected by outliers.

However, there are also some limitations of current methods for effort estimation by analogy. The accuracy and consistency of the derived estimate depends on the quality of the historical data and also on whether the method is able to find analogies between the historical projects and the one being estimated. In the first situation it is useful to calibrate the method to the local data while in the second case it is useful to utilize a tool that identifies projects that cannot be estimated with the classical EBA approach.

Recently, EBA has been improved significantly as a method [10] In this study however, EBA calibration is again done globally, without paying attention to potential estimation inaccuracies for specific projects.

In this study we will utilize EBA approach as implemented by BRACE tool [15]. BRACE tool has a tuning phase during which it determines the best parameters and the attribute subset that will participate in the estimation procedure based on certain accuracy statistics calculated during the estimation process of the historical data. We will further extend EBA method by:

a) Determining the situations under which the most critical configuration parameters of the method (i.e. the measures of similarity) are not appropriate for the estimation of particular projects.

b) Identifying a new set of parameters and attributes that are more appropriate for the estimation of the particular projects.

In particular data that are generated during the selection of the best parameters and attribute subset are analyzed in order to identify project attributes that lead to decreased estimation accuracy. Projects that present best configuration parameters different from the default ones are isolated in order to extract rules that identify more appropriate configuration parameters for the particular projects.

The proposed method is applied and evaluated in the widely known ISBSG data set release 7 [8]. The paper is organized as following: Sections 2, 3 and 4 involve the description of the methods, section 5 presents the data set used, in section 6 we present and discuss the results and in section 7 we conclude the paper.

2. Analogy Based Estimation

Analogy based estimation is essentially a form of case based reasoning. The basic aspect of the method is the utilization of historical information from completed projects with known size, effort or productivity. The most appropriate attributes are selected according to which the new project is compared with the old ones in the historical dataset. The attribute values are standardized (between 0 and 1) so that they have the same degree of influence and the method is immune to the choice of measurement units.

Initially, it is necessary to characterize the new active project, with attributes identical to the ones of the completed projects registered in the database. Examples of project attributes are the implemented functionality, programming language and application type. Attributes are distinguished as quantitative (such as function points [1], measuring the functionality of a software system) or qualitative [3] (such as programming language, measured in a nominal scale with values, “c”, “java” e.t.c).

The next step is to calculate how much the new project differs from the other projects in the available database. This can be done by using a «distance» metric between two projects, based on the values of the selected attributes for these projects. The most known distance metric is the Euclidean or straight-line distance which has a straightforward geometrical meaning as the distance of two points in the k-dimensional Euclidean space:

$$d_{new,i} = \left\{ \sum_{j=1}^{k} (Y_j - X_{ij})^2 \right\}^{1/2} , \qquad i = 1, 2, \ldots, n$$

Other possible distance metrics are the Minkowski distance, the Canberra distance, the Czekanowski coefficient and the Chebychev or «Maximum» distance (see [2] for definitions).

Eventually the estimation of the effort using analogies is based on the completed projects that are similar to the new one. The user of the method has to calculate the distances of the new project from all the database projects and identify few «neighbour» projects, i.e. those with relatively small distance value. The estimation of the effort is eventually obtained by some combination of the efforts of the neighbor projects. Typically, the statistic used is the mean or the median of these effort values. This statistic may be weighted according to the size (physical or functional) of the system under development compared to the sizes of the neighbor projects.

The prediction accuracy of the method may differ according to the peculiarities of the historical data set to which the target project is compared. For this reason, it is a good practice to calibrate the method, i.e. find out empirically the combination of method parameter values that provide the best accuracy results. The method options that may be adjusted are:

(a) The distance metric by which the projects of the database will be sorted according to their similarity to the one under estimation (e.g. Euclidean distance, Manhattan distance)

(b) The number of closest projects (analogies) – it is reasonable to expect that for small data sets a low number of neighbor projects must be used (typically one or two), while for larger data sets the choice depends on the homogeneity of the data,

(d) The statistic for productivity estimation

(e) The weighting of the chosen statistic according to the projects size

In this study we apply and calibrate analogy based estimation for size and effort prediction with the help of Brace tool [15]. The tool finds the best overall parameter configuration given a data set, by trying all feasible combinations of parameter values. Because of the computation complexity for large data sets, the user has to choose the range of values of certain parameters, deciding in advance values for some parameters based on his intuition and experience, in order to reduce the search space. Other researchers follow the same approach as well [13].

3. Rule Induction

Rule Induction [7] is a particular aspect of inductive learning. Inductive learning is the process of acquiring general concepts from specific examples. By analyzing many examples, it may be possible to derive a general concept that defines the production conditions.

Rule induction takes each class separately and tries to cover all examples in that class, at the same time excluding examples not in the class. This is a so called, covering approach, because at each stage a rule is determined that covers some of the examples. Covering algorithms operate by adding tests to the rule that is under construction, always trying to create a rule with maximum accuracy. Unlike other algorithms that choose an attribute to maximize the separation between the classes (using information gain criterion), the covering algorithm chooses an attribute-value pair to maximize the probability of the desired classification.

In this paper, we apply the PART algorithm that is based on extracting partial decision trees utilizing the Weka machine learning library [16].

A simple rule coming from the domain of software cost estimation will have as Rule Body certain software project attribute values and as a Rule Head a productivity (or cost, or effort) value. A simple example of a rule is the following:

If language used = java and development type= enhancement then 40<productivity ≤60 total no of projects= 10 wrong estimates=2

.

This rule is interpreted as following: If the language that will be used for the development of new project is java and the development type of the project is enhancement then there is (10-2)/10 = 80% (confidence value) probability that the productivity value of the project will be between 40 and 60 lines of code per hour.

One advantage of inductive learning over other machine learning methods is that the rules are transparent and therefore can be read and understood. Proponents of RI argue that this helps the estimator understand the predictions made by systems of this type.

4. Proposed Method

The proposed method involves initially the application of analogy based estimation for the identification of the best similarity measures using jackknife method. In each turn, one project is drawn out of the historical data set and used for estimation purposes. For this project we test a variety of combinations of similarity measures and we select the combination that presents the best MMRE estimation accuracy. MMRE is the Mean Magnitude Relative Error and is defined as following:

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{P_i - \hat{P}_i}{P_i} \right|$$

P_i , is the actual productivity required for the completion of a new project *i*, $\hat{P}_i$ the estimated productivity of anew project and *n* is the number of projects in the data set.

This procedure is repeated until all projects are excluded once and estimated by the rest of the projects. Finally the combination of similarity measures that presents the best overall estimation accuracy measured with MMRE is selected as the best configuration.

At this point we have determined a way to select similarity measures in order to apply EBA and generate an estimate for a certain unknown project. However, there maybe exceptions in which the new unknown projects present few or no similarities at all to the historical data based on the selected measures. In such situations it is useful to have an enhancement of a method that specifies such projects and suggests alternative similarity measures.

For this purpose we further analyze the data produced for each project separately by jackknife method. We keep meta-data relative to the estimation procedure of each project separately, containing information about the most accurate distance metric, statistic point estimate and number of analogies for each project. The next step is to compare the accuracy of the estimations produced using the default similarity measures identified globally from the previous step to the accuracy of the best configuration identified for each project separately. Then we analyze the above data in order to produce rules that will indicate project attribute combinations that may lead to different best configuration.

5. Data set description

The data set used in the study is ISBSG release 7 [8], a publicly available multi organizational data set. The International Software Benchmarking Standards Group maintains a repository of international software project metrics to help developers with project estimation and benchmarking. The repository contains 1239 projects that cover the software development industry from 1989 to 2001.

Table 1. Data set description

Variable	Name	Levels values

Development Type	DT	1={enhancement, re – development}
		2={new development}

Development Platform	DP	1={MainFrame}
		2={PC,MidRange}

Language Type	LT	1={3GL, ApG}
		2={4GL}

Programming Language	PPL	1={access} 2={cobol, cobolII, easytrieve, visual basic, natural, other4GL, otherApG, PL/I, powerbuilder, talon}
Database	DBMS	1={acess}, 2={ims} 3={adabas, db2, db2v2, foxpro, idms, sql, oracle, other, watcom}

Used Methodology	UM	1={yes}
		2={no}

Organization Type	OT	1={Banking, ElectricityGasWater} 2={communication, community services, computers, defense, energy, financial, government, medical, professional services, wholesail&retail trade}
Business area type	BAT	1={accounting,banking,engineering} 2={r&d,claims processing, financial, insurance, inventory, legal, personnel, s&m, telecommunications}
Application type	AT	1={MIS, advertising, corporate taxation, data warehouse, DSS} 2={Transaction Processing System, Office Information System}

Package Customization	PC	1={Don’t Know, Yes}
		2={No}

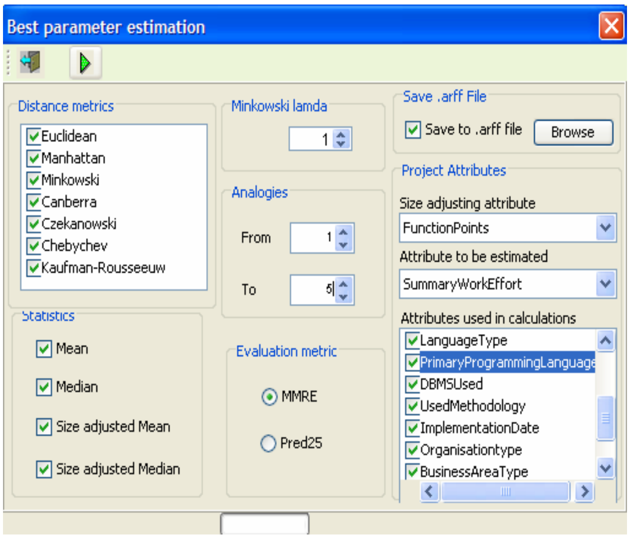
In many records a number of fields are empty or even measured with different approaches. Our target was to include in the study the majority of the projects but also to ensure data validity minimizing the variance between the data because of the differences in measurement, or quality, two conflicting targets. The preparation and transformation of data performed involved the selection of projects with data quality rating A and B (projects with data quality rating C were excluded). Projects for which only the development team effort and support was counted and only staff hours were recorded were selected. At this point 556 projects are considered but if we delete the cases with missing values, the dataset is restricted to 52 projects. Most of the predictor variables are categorical and since the building of a reliable model requires the existence of enough observations in every interaction of the values between dependent and independent variables, we chose to work with fewer categories. This approach is adopted in a similar study [11] where for each one of these categorical variables, one-way ANOVA was performed in order to check the impact of every factor on the original-dependent variable. Every factor with significance less than 0.05 was considered important and was included in the analysis. In [11] the authors used also post hoc tests (Tukey, Tukey's-b, Bonferroni, LSD, Scheffe and Duncan) in order to identify the various homogeneous categories that have to be concatenated in every factor. The categories that are not significantly different can be concatenated. Table 1, presents the variables that participate in the study along with their possible values.

6. Results

As mentioned above, we chose 52 projects characterized by the attributes presented in table 1. From these 52 projects the ten most recent ones implemented after 1997 were used as a validation set while the rest 42 projects were used to train the EBA method. Initially the BRACE tool was used to identify the most appropriate estimation parameters based on the projects in the training set. The parameters for which the tool decided upon are the distance metric, the number of analogies, the statistic used and whether the statistic will be adjusted based on the size of the project.

Figure 1 presents the alternative parameters that can be used for analogy based estimation, among which the tool specifies the optimum ones. The selection of the distance metrics by which the projects of the database will be sorted according to their similarity to the one under estimation is performed comparing 7 possible metrics, the Euclidean, the Manhattan, the Minkowski, the Canberra, the Czekanowski, the Chebychev and the Kaufmann-Rousseeuw distance metrics. The number of neighbor projects from the historical set that will be used for the estimation of a new project are limited from 1 to 5. The upper limit of 5 is selected based on the general assumption that for small data sets the number of analogies used should be around three [3].

The next choice is the statistic used to calculate a point estimate derived. A closely related decision with that choice is whether the statistic will be size adjusted, based on a size attribute. This statistic is actually used to calculate a point estimate based on the effort interval predicted. Possible statistics that the method compares is the mean point, the median point, the size adjusted mean and the size adjusted median. Another choice that can be made at this point is the selection of the project attributes that will participate in the estimation. In our case we select the participation of all attributes in identifying neighbor projects. The combination of parameter values that maximize the overall accuracy of analogy based estimation in the whole training set is the one selected as the best configuration of the model. The best parameters for EBA in the ISBSG data subset considered are presented in figure 2.



Best parameter estimation

Distance metrics

☑ Euclidean
☑ Manhattan
☑ Minkowski
☑ Canberra
☑ Czekanowski
☑ Chebychev
☑ Kaufman-Rousseeuw

Statistics

☑ Mean
☑ Median
☑ Size adjusted Mean
☑ Size adjusted Median

Minkowski lamda: 1

Analogies

From: 1
To: 5

Evaluation metric

● MMRE
○ Pred25

Save .arff File

☑ Save to .arff file

Project Attributes

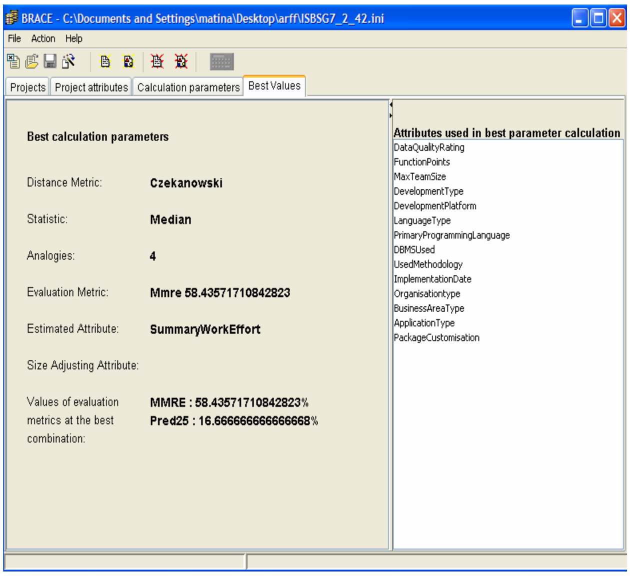
Size adjusting attribute: FunctionPoints

Attribute to be estimated: SummaryWorkEffort

Attributes used in calculations

☑ LanguageType
☑ PrimaryProgrammingLanguage
☑ DBMSUsed
☑ UsedMethodology
☑ ImplementationDate
☑ Organisationtype
☑ BusinessAreaType

Figure 1: Similarity measures that participate in the configuration of ABE



BRACE - C:\Documents and Settings\matina\Desktop\arff\ISBSG7_2_42.ini

Best calculation parameters

Distance Metric: Czekanowski

Statistic: Median

Analogies: 4

Evaluation Metric: Mmre 58.43571710842823

Estimated Attribute: SummaryWorkEffort

Size Adjusting Attribute:

Values of evaluation metrics at the best combination: MMRE : 58.43571710842823%
Pred25 : 16.666666666666668%

Attributes used in best parameter calculation

DataQualityRating
FunctionPoints
MaxTeamSize
DevelopmentType
DevelopmentPlatform
LanguageType
PrimaryProgrammingLanguage
DBMSUsed
UsedMethodology
ImplementationDate
Organisationtype
BusinessAreaType
ApplicationType
PackageCustomisation

Figure 2: Best overall configuration of EBA method for ISBSG data set.

The optimal EBA based on ISBSG data takes into consideration the values of 4 neighbor projects measuring similarity using Czekanowski distance metric and provides point estimates using the median point of the interval without size adjustment. Using this configuration the overall fitting accuracy of the model to the training data evaluated with MMRE (Mean Magnitude Relative Error) metric is 58.43% and with Pred(25) (percentage of projects that are estimated with error less than 25%) is 16.66%. Such metrics are rather modest, since a combination of MMRE less than 25% and Pred(25) greater than 75% is considered quite satisfactory.

While searching for the best global configuration, the values of the best combination of parameters have been recorded for each project in a separate file. This file contains information regarding the best distance metric, number of analogies and statistic for each project alone. Using this information we extract rules that based on the projects attributes suggest the selection of a particular a) distance metric, b) statistic c) number of analogies.

The tool used to extract such models is the Weka Machine Learning Library. The files that are generated by BRACE tool can be readily processed by Weka.

Table 2. Rule set for the estimation of distance metric

PPL=1	Euclidean (8.0)
OT = 2	Czekanowski (4.0/2.0)
MTS <= 2 *and* LT = 1	Czekanowski (3.0)
DBMS =2 *and* FP <= 124 *and* LT = 2	Minkowski (4.0)
DBMS= 2AND FP <= 385 *and* LT = 1 *and* MTS <= 5	Euclidean (3.0/1.0)
LT = 1 *and* FP <= 385: Minkowski	Minkowski (13.0/3.0)
LT = 2	Euclidean (4.0/1.0)
	Chebychev (3.0/1.0)

Table 3 Rule set for the estimation of the statistic

PC=2 *and* BAT=2	Size adjusted Mean (11.0/4.0)
DP=1 *and* PC=2 *and* MTS > 7	Size adjusted Mean (7.0/1.0)
DP=1 *and* AT=2 *and* FP > 109	Mean (9.0/4.0)
DP=1 *and* AT = OTHER	Size adjusted Median (5.0/2.0)
DP=1 *and* LT=1	Size adjusted Mean (4.0/1.0)
FP <= 174	Size adjusted Median (3.0/1.0)
	Median (3.0/1.0)

Table 4. Rule set for the estimation of the number of analogies.

DT = 2 *and* OT = 2 *and* DBMS = 1	2 (6.0/4.0)
DT = 2 *and* OT = 1 *and* AT = 2 *and* FP <= 247	4 (16.0/8.0)
DT=2 *and* OT = 1 *and* AT = 1	3 (8.0/4.0)
DT = 1	2 (5.0/1.0)
OT = 1	5 (4.0/1.0)
	1 (3.0)

The Weka tool is further utilized to provide rules that will help the estimator select the best configuration for the estimation of a particular project based on the values of its attributes. The algorithm used is PART with the default parameters apart from the minimum number of projects per class which is set to 3. Tables 2,3, and 4 present the rules extracted for deciding the distance metric, the number of analogies and the statistic.

For example the first rule of table 2 suggests the use of Euclidean distance when the project under estimation is implemented in a “Programming Language” that belongs to the first category (MS Access). The particular rule suggests the Euclidean distance metric for 8 projects in the training set. For all of these projects the Euclidean distance is the optimal distance metric.

In the rule set that estimates the statistic metric the rule that classifies most of the projects is the first one that suggests the use of the size adjusted mean when PC belongs to the second PC category (see Table 1) and the BAT variable takes values from the second BAT category. This rule suggests the optimal statistic for 7 out of 11 projects which are classified by this rule.

The estimation of effort for these projects based on the optimal overall configuration and the configuration suggested by the rules is presented in table 5. We should mention that three projects from the test set were removed because they were considered as outliers (IDs 25965, 29418 and 19037) because of their unusually small values of effort that did not appear in the training set.

The accuracy of ABE using the overall configuration and the accuracy of ABE with the configuration suggested by the rules is presented in table 6. Results show that there is significant improvement in the accuracy of the models, and that accuracy metrics are much closer to be considered satisfactory.

Table 5. Estimation of effort with the two configurations

Project ID	Actual effort	Global configuration	Rule based configuration
19165	893	961	719
23229	591	528	643,4
31085	170	546	115, 4
25965	21	546	84,1
15199	183	546	174,8
18268	212	732,5	353,6
29418	17	546	155,9
19037	30	423	176,5
23358	391	528	258,7
112583	1464	1038,5	1438,2

Table 6. Accuracy metrics for the two configurations.

	EBA (global configuration)	EBA (rule configuration)
Mmre	106,77	23,91
pred(25)	28,57	57,14

7. Conclusions

In this paper we have suggested a new approach to configure the method of EBA based on the historical data set used for the evaluation. During the training of the model in order to find the best overall configuration information about the best individual project configuration is maintained. This information is further analyzed in order to produce rules that will try to estimate the most appropriate distance metric and number of analogies and statistic for each project.

The results indicate an improvement in terms of accuracy of the proposed method compared to the default configuration. The method, unlikely global configuration, is able to identify for each project separately a particular set of similarity measures values that can identify the most analogous projects in the historical data set. The proposed method takes into consideration the unique attributes of each project deciding upon the distance metric, the number of analogies and the statistic providing a non-parametric model. The new approach increases estimation accuracy and reliability regarding the measures MMRE, Pred25 and provides another way of calibrating the method to the local data.

Future work involves the extraction of rules that will estimate, based on the projects attributes, the attribute subset that is best to participate in identifying neighbor projects. The method looks promising and should be further evaluated in larger data sets.

8. References

[1] Allan J. Albrecht, John E. Gaffney Jr.: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. IEEE Trans. Software Eng. 9(6): 639-648 (1983)

[2] L. Angelis, I. Stamelos. A Simulation Tool for Efficient Analogy Based Cost Estimation, Empirical Software Engineering, Kluwer Academic Publishers, Vol 5 (1) pp. 35-68, March 2000

[3] Angelis, I. Stamelos, M. Morisio: "Building a Software Cost Estimation Model Based on Categorical Data", Proc. 8th IEEE International Symposium on Software Metrics, pp. 4-15 (2001).

[4] S. Bibi, I. Stamelos, L. Angelis: Software Productivity estimation based on Association Rules, in the proceedings of the European Software Process Improvement Conference, 13 A.6, Trondheim, Norway, November 2004.

[5] B. Boehm. Software Engineering Economics, Prentice-Hall, Englewood Cliffs, NJ (1981).

[6] Nan-Hsing Chiu, Sun-Jen Huang: The adjusted analogy-based software effort estimation based on similarity distances. Journal of Systems and Software 80(4): 628-640 (2007)]

[7] D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, MIT Press, 2001

[8] International Software Benchmarking Standards Group, http://www.isbsg.org

[9] R. Jeffery, I. Wieczorek, A comparative study of cost modeling techniques using public domain multi-organizational and company-specific data, Proceedings of ESCOM-SCOPE 2000, Munich Germany, 18-20 Apr. 2000, Shaker Publishing B.V, pp239-248.

[10] Li, J., Ruhe, G., Al-Emran, A., and Richter, M. M. 2007. A flexible method for software effort estimation by analogy. Empirical Softw. Engg. 12, 1 (Feb. 2007), 65-106.

[11] P. Sentas, L. Angelis, I. Stamelos: Software Productivity and Effort Prediction with Ordinal Regression, Journal of Information & Software Technology, 47 (1): 17-29, (2005).

[12] M. Shepperd, C. Schofield, B. Kitchenham: Effort Estimation Using Analogy. ICSE 1996: 170-178

[13] Martin Shepperd, Chris Schofield, "Estimating Software Project Effort Using Analogies," IEEE Transactions on Software Engineering, vol. 23, no. 12, pp.736-743, 1997.

[14] Stamelos, L. Angelis, M. Morisio, E. Sakellaris, G. Bleris, Estimating the development cost of custom software, Information and Management 40 (2003) 729-741.

[15] Stamelos, L. Angelis, E. Sakellaris. BRACE: BootstRap based Analogy Cost Estimation, Proc. 12th European Software Control Metrics, pp. 17-23 (2001)

[16] Witten, I., and Frank, E. "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.