Software Cost and Duration Estimation Based on Distributed Project Data : A general framework

Safae Laqrichi, François Marmier, Didier Gourc

1

Software cost and duration estimation based on distributed project data: A general framework

S. Laqrichi, F. Marmier and D. Gourc
Université de Toulouse, Mines Albi, Centre Génie Industriel, Route de Teillet, Campus Jarlard, 81013 Albi Cedex 09, France

Abstract. Effort estimation is one of the most challenging tasks in the process of software project management. Enhancing the accuracy of effort estimation remains a serious problem for software professionals. Accurate estimation is difficult to achieve. The main difficulty is to collect distributed knowledge as data and information are often dispersed over different services, departments or organisations. Other main difficulty is to propose a model representative enough of this multi-partner behaviour. The objective of this study is to propose a general framework of the estimation starting from the analysis of the available projects database, the choice and establishment of estimation model, up to the use of this model to make estimation for new projects. In this paper, a comparative study between regression models and neural network models is performed. The proposed study is applied on a dataset of an automotive company.

Keywords: neural network, regression, duration estimation, cost estimation, comparison

1.1 Introduction

Effort estimation is an important activity in software project management. Estimation in software projects consists in predict likely amount of effort, time and staffing level that are required to build a software system. It is used in the whole development life cycle of the software project from the bidding until the maintenance of the software. Both project managers and clients use effort estimation to predict the effort, the duration and the cost required to develop their software projects in order to establish contracts. Under estimating the effort and the cost required to develop a project results in budget overruns, while over estimating can lead to miss of biddings. Accurate estimation is then very important for companies' benefit and success.

Estimation is a complex activity that requires a high level of interoperability in both steps of model establishment and estimation for new projects. Indeed, the

modeling step needs various data about previous projects that are dispersed over different services and entities (financial data, technical project data…). The estimation step requires various distributed data about new projects to estimate as well. Thus, entities have to communicate and collaborate to provide required information.

Various effort and duration estimation approaches were been developed. Traditional and well-known ones include expert judgment, Delphi, COCOMO and Putnam's SLIM. However, estimation methods did not produce sufficiently accurate results, this is why approximately 44% of software projects according to the Standish Group International fail on meeting the commitment on quality, time and cost.

A set of factors can influence the estimation accuracy and lead to estimations far from the reality. It includes, among others: lack of information about completed previous projects; use of new technologies; lack of experience with similar projects; choice of estimation approach and more [1].

The challenge of improving estimates accuracy has led to the development of several new methods and techniques for effort, duration and cost estimation. These methods are based on artificial intelligence such as NN (Neural Network) models.

Our work first focuses in formalizing the general estimation framework. This framework enables to compare different models. In this paper, we study the case of regression and NN estimation models applied on a big and diversified case study. This case study does not contain size project that is usually considered to be an important cost driver in estimation model establishment.

The present paper is organized into three sections: the first section presents literature review on effort estimation process, regression and NN models and their comparisons work carried by researchers. In the second section, a general framework for estimation is proposed. Finally, in the third section, the proposed framework is applied to a case study from the automotive industry.

1.2 Literature review

1.2.1 Estimation in software projects

The estimation process is based on two principal activities that are: (i) project size measure and (ii) effort, cost and duration estimation.

Project size (i) expresses the size of the software that is derived from the quantification of functional requirements specified by users [ISO/IEC14143]. Project size can be calculated by several methods and techniques of functional measurement such as FPA (Function Point Analysis) and COSMIC FP (COSMIC Function Point), thus it can be expressed in different units such as function points (FP) and source lines of code (SLOC).

The development effort (ii) is a function of the project size; it is expressed in man-hours, man-days or man-months. Duration estimation is either a function of project size or can be derived from the development effort. Effort and duration estimation, once estimated enable to calculate the project cost and staffing.

Various effort estimation methods can be used in the estimation process. They can be grouped in three main categories: (1) experience based methods, which is based on the expert intuition and experience drawn from previous executed project, such as expert judgment and analogy, (2) algorithmic model based methods, which are mainly based on equations expressing the effort as a function of discriminant parameters influencing the effort called effort drivers. Parametric models are established using historical data from complete projects, some of commonly used models are regression based models and Bayesian analysis based models [2]. (3) Non-algorithmic model based methods, which model the relationship between the estimated variable and cost drivers using artificial intelligence techniques like NN and fuzzy logic. The relationship is not assumed to be well known or modelizable to specific shapes or equations [3].

Regression is a widely used modeling technique and NN is a recent and evolutionary modeling technique. In this study, our attention was drawn to these two modeling techniques for the estimation activity of the global process (ii) because they seem to provide good estimates.

1.2.2 Regression models versus Neural Network models

Regression models are still the most popular models in literature; they include COCOMO [4] and Putnam [5]. Regression aims to model the relationship between inputs and outputs. In software estimation, the inputs are the discriminant parameters influencing the estimated variable called effort or cost drivers, the mean cost driver is the software size that is usually expressed in the Source Lines of Code. The output is the estimated variable that can be effort or duration or cost.

There are various types of regression that have been used in effort estimation models namely linear or multi linear regression[6], non-linear regression that was [4], and ordinal regression [7].

NN is a massively parallel adaptive network of simple nonlinear computing elements called Neurons, which are intended to abstract and model some of the functionality of the human nervous system in an attempt to partially capture some of its computational strengths [8].

NN is used in effort estimation due to its ability to learn from previous data. It is also able to model complex relationships between the dependent (effort or duration or cost) and independent variables (cost drivers). In addition, it has the ability to learn from the training data set thus enabling it to produce acceptable results for unseen data [4]. But NN has one short coming that its estimation reason or relation between inputs and outputs cannot be justified. Different NN based model are proposed to predict and estimate effort and duration as COCOMO based NN model [9][10] and radial basis function NN model [11].

Several research works have compared NN models with regression models. Finnie [12] compared regression model with two AI based estimation models that are case based reasoning and NN for software development effort. Authors found that AI based models perform better for complex software projects and outliers in training dataset than regression model. However, they gave no justification or explanation for the obtained results.

Heiat [13] experimented FFNN (Feed-Forward Neural Network) with function point and RBNN (Radial Basis Neural Network) with SLOC for a dataset of 67 projects. Author concluded that NN approach is competitive with regression for same cases and significantly more accurate for others. This study presents some limitations: the size of data sets is small, the data sets used varied only in size and complexity varied in terms of language platform.

1.3 Framework of estimation of effort, duration and cost

The study prensented in this paper relies on the framework shown below (Fig 1.1.)

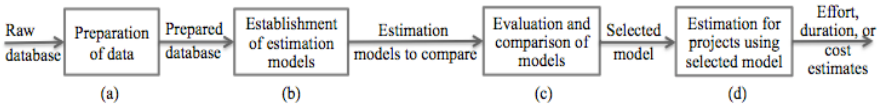

Fig 1.1. Framework of estimation for a software project

This methodology consists on three major steps:

- (a) Preparation of data: based on raw database containing information collected from previous achieved software projects
- (b) Establishment of estimation models: consists on the establishment of estimation models to compare. For this study, the estimation models processed and compared within this framework are (b1) regression model and (b2) NN model.
- (c) Evaluation and comparison of models using evaluation criteria and performance indicators.
- (d) Estimates of effort, duration or cost using the selected estimation model

These four steps are detailed below.

1.3.1 Data preparation (a)

Organization's projects database is built over years by projects teams in order to capitalize the experience and information related to completed projects. It contains information about previous achieved projects such as project duration, project cost, project type, and platform development.

Steps for database preparation can be summarized as follows:

- Cleaning database: datasets related to irrelevant project parameters, such as parameters concerning information capitalization are discarded. Also, duplicate projects, that are projects with the same parameters but different estimates, are reduced.
- Performing statistical tests: The projects database is explored to determine cost drivers. For this purpose, the statistical test of Pearson correlation and one-way ANOVA can be used [14], they enables to examine the significance between the projects parameters and the variables to estimate in order to select the parameters with significant influence on these variables. The Pearson's correlation test is used for parameters with the ratio scale

[15] whereas One-Way Analysis of Variance (ANOVA) is used for parameters with the nominal scale. After determining cost drivers, the other parameters are discarded from the database. Projects with missing values in cost driver fields are discarded. Then, the variables to estimate are adjusted to be normal by discarding projects identified as outliers. A step of data normalization is required for the NN model establishment.

- Dividing database: the projects data should be divided into two segments, one used to establish and train the effort estimation model and the other used to test and validate it. The holdout method or the k fold cross-validation [16] approach can be used for this purpose.

In this stage, the projects database is prepared in order to establish effort estimation models.

1.3.2 Establishment of estimation models

(b1) Regression model establishment

The regression model establishment consists on modeling the relationship between dependant variables Y and independent variables X_i in the form of a linear equation as:

$$Y=a_1X_1+a_2X_2+\ldots+a_nX_n \quad (1.2)$$

For our study, dependent variables are variables to estimate and independent variables are cost drivers. In order to establish the multi linear regression between these variables based on database, many statistical tools can be used such as XLSTAT.

(b2) NN model establishment

Steps for NN model establishment can be summarized in the figure below (Fig 1.2.)

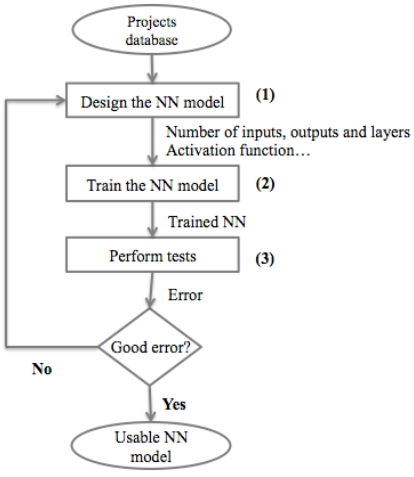

Fig 1.2. Algorithm for NN model establishment

1. NN design consists in defining the architecture of the NN (the number of inputs, the number of output, the number of hidden layers and nodes, activation function). The number of inputs is the number of the projects characteristics, the number of outputs is the number of variable to estimate, the numbers of hidden layer and hidden layer nodes are less than or equal to twice the number of inputs [17]. Activation function is used to transform and squash the amplitude of the output signal of a neuron to some finite value. Some of the most commonly used activation functions are sigmoid, Tanh, and Gaussian [18]. There are a multitude of NN architecture and structure; the most used one is called Multilayer Perceptron (MLP) that is a feed forward artificial NN, i.e. the network is structured in a hierarchical way. It consists of different layers where the information flows only from one layer to the next layer. In this NN, Each node in one layer connects with a certain weight w_{ij} to every node in the following layer. Input nodes distribute the signals to the nodes of the first hidden layer without processing it while nodes of hidden layers are neurons (or processing elements) with a nonlinear activation function [19,20] (view Fig 1.2.).

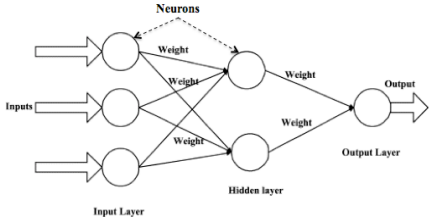

Fig 1.3. Architecture of a Multilayer Perceptron (MLP)

2. Training: The two main techniques employed by neural networks are known as supervised learning and unsupervised learning. In unsupervised learning, the NN requires no initial information regarding the correct classification of the data it is presented with. Supervised training works in much the same way as a human learns new skills, by showing the network a series of examples. The most used supervised training algorithm is back propagation algorithm [13]. The objective of training is to determine the weights of the NN to bring the outputs of the whole network closer to the desired outputs.
3. Test: After training NN, the test is performed on the database reserved for testing. It makes it possible to measure the potential of success of the trained NN using evaluation criteria. As the figure show, if the results of test are not satisfying, the architecture of the NN model is modified until reaching good error.

1.3.3 Evaluation and comparison of models (c)

Model evaluation aims to measure how much model fits the context of the study. This context is defined by the variables to estimate, the database and the cost drivers used in estimation process.

The evaluating of estimation models makes it possible to compare them in order to choose the most adequate one. For this purpose, different accuracy indicators can be used for this study such as the Mean Magnitude of Relative Error (MMRE), the Pred(0.25) [21] [22] [23].

1.3.4 Estimation of effort, duration or cost (d)

This last step consists on the implementation of the established models in order to estimate effort, duration or cost for new projects. For estimation using regression model, parameters of the model must be determined for the project to estimate. Then the variable to estimate is simply calculated. For estimation using NN model, parameters must be determined then normalized in the same manner as projects database was normalized during database preparation (a). After that, the variable (s) to estimate is (are) then calculated then converted back to their real scale.

1.4 Case Study

The experiment described in this paper is carried on the data provided by an industrial company operating in the automotive sector. The main mission of the company is the design, production and sale of vehicles and mechanical components. The company is also involved in financing vehicle sales and dealership inventories.

The database used in this case study consists on 6078 projects that are either carried or under way. These projects concern several domains of software industry (Cars, finance, commerce…) and their informations are organized into fourteen categories that involve 364 attributes. Due to lack of information about development effort, our focus will be put on the estimation of duration and cost.

1.4.1 Implementation of the methodology

The analysis of the database and the statistical tests enables to obtain a database of 214 projects and 4 attributes that are: project type, project BU, project difficulty, and domain. All cost drivers are qualitative, hence they should be transformed into dummy or binary variables [24].

For regression model establishment (b1), software of data and statistic analysis is used in this work; it makes it possible to automatically establish the regression model using the database. As cost drivers are qualitative in this case study, a special case of linear regression called Analysis of variance (ANOVA) [25] is used to both transform cost drivers into binary variables and establish the estimation model.

For NN model establishment (b2), after transforming cost drivers into binary variables, outputs, that are duration and cost, are normalized to values between zero and one. The resulting database for NN model establishment consists of 27 binary cost drivers and 2 outputs.

For this case study, MLP architecture is used with 27 inputs and 2 outputs. In order to determine the hidden layers numbers, training is repeated many times with the variation of the hidden layers number. The best number of hidden layers is that which provide the best performance in test phase. For training, as many experiences have shown that most NN are trained enough in less than 1000 epochs [26], the number of epochs in this study is set to 1000 epochs.

1.4.2 Experimental results

The established regression and NN model for estimating duration and cost are applied on the case study, and then the evaluation criteria are calculated. Table 1.1. presents accuracy indicators calculated for duration and cost estimation using regression and NN.

Table 1.1. Comparison of results

Evaluation criteria	**Duration estimation models**		**Cost estimation models**	
	Using regression	**Using neural network**	**Using Regression**	**Using neural network**
MMRE %	69	23	658	14
Pred(0.25) %	20	72	7	76

We use these models to estimate duration and cost of a project for which real duration and cost has been measured at the end of the project by the project management service. The proposed approach gives the results in Table 1.2.

Table 1.2. Estimation for a project

Variables to estimate	**Real achievement**	**Estimation using neural network**	**Estimation using regression**
Duration (day)	73	62,8	36
Cost (K€)	14,02	17,18	4556,5

1.4.3 Discussion

In this study case, effort estimation model cannot be established because information about effort is not provided in the database.

Table 1.1 shows that, compared to regression model, NN model provides more accurate estimation for both duration and cost. This can be explained by the capacity of NN to model complex relationship between cost drivers and variables to estimate. A second explanation has to do with the complexity of the case study. This complexity manifests through the lack of relevant parameters due to the lack of information in database and the use of only qualitative cost drivers.

Our approach was applied on an example of a project with known achievement characteristics to concretely observe the estimation results (table1.2.), for this example NN model shows better results than regression as well. The differences between the estimated variables and the real ones may be due to the uncertainty in the model parameters and components. For neural network there is a significantly

small uncertainty because NN has the ability to deal with the lack of data and cost drivers, in fact, it adjusts the model's weights so that it covers this lack. But for regression models, a complete database and a complete list of cost drivers is necessary in order to achieve good results. Otherwise, there will be a bigger uncertainty in regression model's coefficients.

1.5 Conclusion

The more accurate estimation is, the better the software project complies with the contractual commitments in terms of budget and duration. The model used for estimation is a crucial factor that affects estimation accuracy.

We presented a framework for estimation starting from the analysis of the available database up to the selection of the estimation model and its use on new projects. This framework is sufficiently flexible to provide estimation of different variables such as effort, duration and cost, depending on the available database about previous completed projects.

The attention was drawn to two models that are regression and NN models. The proposed framework was then applied on an industrial study case that consists of multisite IT projects. This study has shown that NN model is more accurate than regression model even with an important lack of information about previous projects. This lack of information can explain the uncertainty in estimations. Thus, it will be important to be able to measure this uncertainty in order to take it into account in the estimation process.

Future research studies can focus on the need of more realistic estimations by providing not a single value but an interval of estimation and a degree of trust associated to this interval.

Acknowledgment

This work has been funded by the Fund Unique Interministerial (FUI) through the project Projestimate. We wish to acknowledge our gratitude and appreciation to all project partners for their contribution.

References

[1] Heemstra FJ. Software cost estimation. Inf Softw Technol. 1992;34(10):627‑39.
[2] Laqrichi S, Marmier F, Gourc D. Toward an effort estimation model for information system project integrating risk. 22nd ICPR; 2013.
[3] Idri A. Un modèle intelligent d'estimation des coûts de développement de logiciels. Université du Québec à Montréal; 2003.
[4] Boehm BW. Software engineering economics. Upper Saddle River: Prentice Hall; 1981.
[5] Basha S, Ponnurangam D. Analysis of Empirical Software Effort Estimation Models. 2010;7(3):68‑77.

[6] Kok P a. M, Kitchenham DBA, Kirawkowski DJ. The MERMAID Approach to software cost estimation. ESPRIT '90. Springer Netherlands; 1990. p. 296‑314.
[7] Sentas P, Angelis L, Stamelos I, Bleris G. Software productivity and effort prediction with ordinal regression. Inf Softw Technol. 1 janv 2005;47(1):17‑29.
[8] Haykin SS. Neural Networks: A Comprehensive Foundation. Prentice Hall International; 1999.
[9] De Barcelos Tronto IF, Da Silva JDS, Sant'Anna N. Comparison of artificial neural network and regression models in software effort estimation. 2007; 771‑6.
[10] Idri A, Khoshgoftaar TM, Abran A. Can neural networks be easily interpreted in software cost estimation? IEEE;1162‑1167.
[11] Idri A, Zakrani A, Zahi A. Design of radial basis function neural networks for software effort estimation. IJCSI Int J Comput Sci Issues. 2010;7(4).
[12] Finnie GR, Wittig GE, Desharnais J-M. A comparison of software effort estimation techniques: Using function points with neural networks, case-based reasoning and regression models. J Syst Softw. déc 1997;39(3):281‑289.
[13] Heiat A. Comparison of artificial neural network and regression models for estimating software development effort. Inf Softw Technol; 2002;44(15):911‑22.
[14] Huang S-J, Chiu N-H, Liu Y-J. A comparative evaluation on the accuracies of software effort estimates from clustered data. Inf Softw Technol. août 2008;50(9–10):879‑888.
[15] Neto AM, Rittner L, Leite N, Zampieri DE, Lotufo R, Mendeleck A. Pearson's Correlation Coefficient for Discarding Redundant Information in Real Time Autonomous Navigation System. IEEE Int Conf Control Appl 2007 CCA 2007; 2007; 426 ‑431.
[16] Refaeilzadeh P, Tang L, Liu H. Cross-Validation. In: Liu L, Özsu T, éditeurs. Encycl Database Syst [Internet]. Springer US; 2009; p. 532‑538.
[17] Boetticher G. An assessment of metric contribution in the construction of a neural network-based effort estimator. Second Int Work Soft Comput Appl Softw Eng Enschade NL; 2001.
[18] Karlik B, Olgac AV. Performance analysis of various activation functions in generalized MLP architectures of neural networks. Int J Artif Intell Expert Syst. 2011;1(4):111‑22.
[19] Trenn S. Multilayer perceptrons: approximation order and necessary number of hidden units. IEEE Trans Neural Networks Publ IEEE Neural Networks Counc. mai 2008;19(5):836‑844.
[20] Rosenblatt F. Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Spartan Books; 1962.
[21] Conte SD. Software Engineering Metrics and Models. Benjamin-Cummings Pub Co; 1986.
[22] Kemerer CF. An empirical validation of software cost estimation models. Commun ACM. 1987;30(5):416‑29.
[23] Moores TT. Developing a software size model for rule-based systems: a case study. Expert Syst Appl. nov 2001;21(4):229‑237.
[24] Garavaglia S, Sharma A. A smart guide to dummy variables: four applications and a macro. Proc Northeast SAS Users Group Conf;1998.
[25] Gelman A, Tjur T, Mccullagh P, Hox J, Hoijtink H. Analysis of variance: Why it is more important than ever. Discussions. Author's reply. Ann Stat. 33(1):1‑53.
[26] Boetticher G, Eichmann D. A neural network paradigm for characterizing reusable software. Citeseer; 1993.