Back-Propagation Artificial Neural Network for ERP Adoption Cost Estimation

Mohamed T. Kotb[1], Moutaz Haddara[2], and Yehia T. Kotb[3]

[1] Software Engineer, London, Ontario, Canada
[2] University of Agder, Norway
[3] University of Western Ontario, Canada
mthabet@gmail.com, moutaz.haddara@uia.no,
ykotb@csd.uwo.ca

Abstract. Small and medium size enterprises (SMEs) are greatly affected by cost escalations and overruns Reliable cost factors estimation and management is a key for the success of Enterprise Resource Planning (ERP) systems adoptions in enterprises generally and SMEs specifically. This research area is still immature and needs a considerable amount of research to seek solid and realistic cost factors estimation. Majority of research in this area targets the enhancement of estimates calculated by COCOMO family models. This research is the beginning of a series of models that would try to replace COCOMO with other models that could be more adequate and focused on ERP adoptions. This paper introduces a feed-forward back propagation artificial neural network model for cost factors estimation. We comment on results, merits and limitations of the model proposed. Although the model addresses SMEs, however, it could be extended and applied in various environments and contexts.

Keywords: ERP, cost estimation, neural networks, SMEs.

1 Introduction

Due to their large scale, complexity and substantial investments, ERP systems have been a center of attention from academia and practice. For various reasons, more and more SMEs are adopting ERP systems. SMEs are fundamentally different environments when compared to large enterprises [1]. SMEs are greatly affected and more sensitive to costs than large enterprises, as they have limited budgets and scarce resources [2]. ERP adoption projects are non trivial, and they require careful planning, budgeting, management, and execution. ERP adoptions research shows that cost overruns usually occur during the ERP adoptions, and companies cross their estimated budgets significantly [3-6]. Project budgeting and cost estimation are necessary in the preparation and planning phase for ERP systems' adoptions. They give an insight into the roadmap of the adoption project boundaries, as they highly contribute to project success and the prevention of cost overruns, and in some cases, cost overruns have driven the implementing companies to bankruptcy [5, 7, 8]. There is few research that targets ERP cost estimations in SMEs [2]. Thus the need for more reliable and realistic ERP cost estimation models persists [2, 7, 9-11].

M.M. Cruz-Cunha et al. (Eds.): CENTERIS 2011, Part II, CCIS 220, pp. 180–187, 2011.

This paper aims at calculating weights for ERP cost factors using synthetic data coupled with artificial neural network technology (ANN). The objective of using the ANN technology is to calculate the weights based on real scenarios that represent successes and failures in industry. The advantage of using this technology is that whenever the parameters change, or a new parameters/perspective is added to the parameters profiles, all is needed is to have the ANN retrained. In future research, the proposed model will be fed with historical data.

The rest of this paper is organized as follows: Section 2 provides an overview of literature in the area of ERP cost estimation. Section 3 formulates the problem under discussion. Section 4 shows the chosen optimization method for cost factor estimation. Section 5 discusses the architecture and training phases of the artificial neural network. Section 6 illustrates the experiments and results, and finally section 7 concludes the work and highlights the proposed solution extensions and future work.

2 Study Background

In literature, very few studies address ERP costs identification and ex-ante evaluation in SMEs environments [2], as well as, there is an evident gap in ERP adoption cost management and estimation research areas [2, 5, 10, 12, 13]. The gap is partly because the ERP adoption cost identification and estimation is a complex chore [4, 5, 10, 14-19]; it requires attentive analysis of both direct and indirect (*usually hidden*) costs. Moreover, the established and extensively used software cost estimation models e.g. COCOMO (COnstructive COst MOdel) [20] are not adequate to an ERP setting [5, 10-12, 14, 21]. COCOMO and analogous models are primarily focused on estimating software development costs, and some of their considered cost factors might not be valid for ERP adoption projects, as lines of code (KLOC) and development time (D) are not pertinent factors in an ERP context [10, 14, 17, 21-23]. Nevertheless, these models could be relevant to ERP vendors when pricing their ERP packages.

As an effort to extend the application of COCOMO into ERP systems implementations, Maya Daneva [13] introduced a model that complements the classic COCOMO model [20] with Monte Carlo simulation to introduce errors that results in more realistic estimates. The model has been designed to take the management portfolio under consideration [24]. The model has been seen be one good alternative to ERP adopters as it does not need inputs from the ERP vendors or consultants. Other researchers have investigated software development cost estimation as a guide for package pricing, like [5, 20, 25, 26]. Another paper explored cost estimates for cross-organizational ERP projects [10], while Brocke, et al. [27] adopted a Transaction Costs theory in order to govern ERP costs in a service oriented architecture (SOA) implementation context. Moreover, another research focused on adoption long-term business opportunity costs in contrast to benefits [28].

Although it wasn't applied in ERP cost estimation for SMEs, however, the use of artificial neural network for cost estimation is not new. It has been used in different fields in industry [24, 29, 30]. On the other hand, the input variables values ranges in application domains in [24, 29, 30] can be predictable. This leads to a faster weight convergence. ERP cost factors estimator can receive inputs with a very large range of values. Data preprocessing is required to facilitate the training task on the training algorithm.

3 Problem Formulation

The cost factors estimation for ERP products adoption is an NP-Complete problem for its nature. NP-Complete problems are defined to be the set of problems where an accurate solution using the available processing time takes Millions or even billions of years. Solutions for this problem class is always approximate within an acceptable error range.

The Problems under discussion can be seen as a set of profiles, every profile is collected from a company that went through the ERP adoption process, costs that map to the set of profiles and a set of weights where every weight is associated to a factor. We will call the cost factors, the costs and the associated weights as ERP adoption cost components.

The relationship between the ERP adoption cost components is governed by the following equation:

$$\begin{pmatrix} P_{1,1}\ldots\ldots\ldots\ldots\ldots P_{1,m} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ P_{n,1}\ldots\ldots\ldots\ldots\ldots P_{n,m} \end{pmatrix} \begin{pmatrix} \omega_1 \\ . \\ . \\ \omega_m \end{pmatrix} = \begin{pmatrix} \phi_1 \\ . \\ . \\ \phi_m \end{pmatrix}, \tag{1}$$

Where $(P_{1,1}\ldots\ldots\ldots\ldots P_{1,m})$ are the cost factors for the first profile, $\varnothing 1$ is the first profile associated cost and $\omega_1$is the weight associated to the cost factor.

The Weight vector ω is the unknown piece of equation. It is imperative to determine the weights values that minimize the following equation:

Minimize
$$\sum_{k=1}^{m}\phi_k - \sum_{i=1}^{n}\sum_{j=1}^{m}P_{i,j}\omega_j \tag{2}$$

4 Optimization Method

The choice of the artificial neural network technique came as an answer for the following two questions:

1- Are the equations simultaneous? Simultaneous equations are those that have the same unknowns and the same solutions. Some of the unknowns in cost factors equations are highly subjective. An example is giving a value that best represents the learning curve of the staff in the company. This means that there can be a lack of integrity and inconsistency between the same parameter collected from two different institutions. The other issue is that not always the same parameters types or numbers are collected.

2- Are the collected parameters accurate? Like mentioned in the above paragraph, some parameters values are very subjective. Values that are given to those parameters are typically holding considerable errors when projected on the big picture. An example is comparing expertise in one company to another.

The answers to the above two questions favor optimization techniques that will traverse a solution surface and select, if the data is error free, the global minimum. If the data contain a considerable amount of errors, a local optimum is found. The vectors in Equation (1) are not simultaneous, so whether they are linear or nonlinear, there is no unique solution for vector ω values. The equations parameters are holding a considerable amount of errors due to human errors in estimation and sizing. The neural network with its inherent capability in generalization can overcome the challenging nature of the data.

5 Proposed Back-Propagation Artificial Neural Network Model

In section 3 the characteristics of the solution method is summarized as a method that is tolerant to errors and imposes a generalization behavior throughout the solution life cycle. Nevertheless, able to traverse solution surface and reach to an optimum solution. All those characteristics favor the artificial neural network solutions. Neural networks can be black boxes that do not need a neural network expert to train them. Profiles are introduced to these networks as inputs and classifications as outputs.

Neural networks training is either supervised (feed-forward back-propagation networks) or unsupervised (associative, competitive, etc.). Supervised training is the paradigm that is chosen in this research because the output during training is controlled. Basic principles of artificial neural network can be found in [31, 32].

The number of neurons in the input layer in the proposed architecture equals to the number of cost factors. The number of neurons in the hidden layer is chosen to be the number of input neurons. Optimum selection of the number of hidden layer neurons is explored in [33].

The number of neurons in the output layer is thirty six neurons, every neuron gives either 0 or one. Every four neurons represent the BCD coding for the set of numbers from 1 to 9. The Highest twelve neurons represent the billion range, where the lower twelve neurons represent the million range, and the lowest twelve represent the thousands. A cost of 1 billion, three hundred million and fifty thousands are represented as {0000 0000 0001 - 0011 0000 0000- 0000 0101 0000}.

Table 1 shows possible values for two independent parameters. Parameter 1 can be more effective to cost factors than parameter 2, and therefore its weights or cost factor should be more significant than those of parameter 2. It is a good practice to put both parameters in one domain, so that their values can be more comparable. This domain, we call it the normal form, should reflect how effective the rate if change is to the final total cost. The best representative of this is standard deviation. Comparing the standard deviation of every parameter values against the corresponding costs not actually normalized all data but gives indications about the sensitivity of the cost against every parameter.

Table 1. Abstraction of two independent variables possible values

Parameter 1	Parameter 2
0.1	500
0.3	300
0.5	1000

The training algorithm for the neural network is on two stages, feed-forward and back-propagation. Weights are given initial values all over the network. These initial weights can be of the same value or random. The input vectors are fed to the network and propagated though all the connections and the actual outputs are perceived. The forward propagation is governed by the following equations:

$$O_j = \sum I_i \omega_{ij} + \theta_j \tag{3}$$

Where O_j is the output of node j, I_i is the output of the input node I_i, ω_{ij} is the weight that links input node I_i with output node O_j and θ_j is the threshold on the activation function for output node j.

The sum square error of the outputs of the network on all patterns is:

$$\varepsilon = \frac{1}{2}\sum_{i=1}^{n}(d_i - O_i)^2 \tag{4}$$

ε is the SSE, d_i is the desired output on the output layer neuron I, and O_i is the actual output on the same neuron. The network weights update is governed by the equation:

$$\omega_{i+1} = \omega_i + (d - O)O(1 - O) \tag{5}$$

Equation (5) assumes that the activation functions on the output nodes are step functions (output is either 0 or 1). O in equation (5) is the actual output, d is the desired output and i is the iteration number.

6 Experiments and Discussions

Five hundred different patterns are used in training and five hundred patterns are used for testing. The testing results show that the success of cost estimation highly depends on the correlation between the input pattern and at least one of the training patterns. Estimation error is inversely proportional to the correlation. Figure 1 shows the relationship.

The experiments were conducted using synthetic data driven from both Gaussian and uniform distributions. Uniform distributions showed better success rate than the Gaussian. The difference in success rate depends on the standard deviation which the Gaussian variants are driven from. High standard deviations lead to higher failure rates. In general, the experiments indicate the necessity of having high population of data profiles in order to develop a robust kernel (neural network) for cost factors estimation. Moreover, it is expected that two data profiles that lead to the same cost will have high correlation.

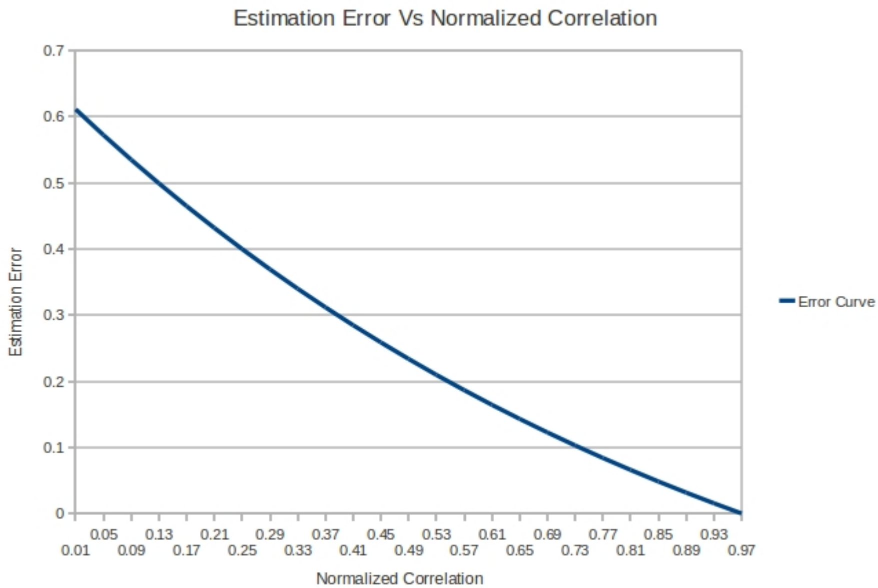

Fig. 1. Estimation error versus correlation between the input data and correlation data. X-Axis is the normalized correlation and Y-Axis is the estimation error.

7 Conclusion and Future Work

The proposed framework allows for cost factors estimation without the need of the involvement of architects or project managers to define function points as an input for systems such as COCOMO. The proposed framework is neural network based. A feed-forward back-propagation artificial neural network is proposed for cost factor estimation. The neural network is composed of an input layer with a number of neurons equals to the number of the data factors, a hidden network with a number of neurons equals to the number of input neurons and a thirty six output neurons that cover the cost ranges from thousands to billions. Every digit in this range is represented by four neurons, BCD encoding.

Figure 1 reveals the most obvious limitation to the model. The model is limited to its high noise and inaccuracy. The experiments show that the accuracy of data collection is a key factor for successful and accurate cost factor estimation. As a future work, data used for training should be grouped according to relevance and correlations. Inaccurate data should be identified and factored out.

An extension for this work is a correlation based technique that should identify dependencies and relations between factors. Input patterns should be classified according to correlation and training takes role on relevant data patterns separately. When an input pattern to be tested, it gets classified first and then propagated through the relevant network.

Another promising venue is building a case-based reasoning on the classified patterns. Once data is classified, it is much easier to be processed among closer and more relevant peers.

References

1. Welsh, J.A., White, J.F.: A small business is not a little big business. Harvard Business Review 59(4), 18–27 (1981)
2. Haddara, M., Zach, O.: ERP Systems in SMEs: A Literature Review. In: HICSS, Kauai, Hawaii, vol. 44 (2011)
3. Haddara, M., Päivärinta, T.: Why Benefits Realization from ERP in SMEs Doesn't Seem to Matter? In: HICSS, vol. 44. IEEE, Kauai (2011)
4. Holland, C.R., Light, B.: A critical success factors model for ERP implementation. IEEE Software 16(3), 30–36 (1999)
5. Jones, C.: Estimating software costs Bringing realism to estimating, 2nd edn. McGraw-Hill Companies, New York (2007)
6. Martin, M.H.: An ERP Strategy. In: Fortune, pp. 95–97 (1998)
7. Haddara, M.: ERP Adoption Cost Factors in SMEs. In: European and Mediterranean Conference on Information Systems (EMCIS 2011), Athens, Greece, June 30-31 (2011)
8. Newman, M., Zhao, Y.: The process of enterprise resource planning implementation and business process re-engineering: tales from two chinese small and medium-sized enterprises. Information Systems Journal 18(4), 405–426 (2008)
9. Daneva, M.: Approaching the ERP Project Cost Estimation Problem: an Experiment. In: Proceedings of the First International Symposium on Empirical Software Engineering and Measurement. IEEE Computer Society Press, Los Alamitos (2007)
10. Daneva, M., Wieringa, R.: Cost estimation for cross-organizational ERP projects: research perspectives. Software Quality Journal 16(3), 459–481 (2008)
11. Elragal, A., Haddara, M.: The Use of Experts Panels in ERP Cost Estimation Research. In: Quintela Varajão, J.E., Cruz-Cunha, M.M., Putnik, G.D., Trigo, A. (eds.) CENTERIS 2010. CCIS, vol. 110, pp. 97–108. Springer, Heidelberg (2010)
12. Al-Mashari, M.: Enterprise resource planning (ERP) systems: a research agenda. Industrial Management & Data Systems 102(3), 165–170 (2002)
13. Daneva, M.: Complementing approaches in ERP effort estimation practice: an industrial study. In: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering. ACM, New York (2008)
14. Abdel-Hamid, T.K., Sengupta, K., Swett, C.: The impact of goals on software project management: an experimental investigation. MIS Q 23(4), 531–555 (1999)
15. Buonanno, G., et al.: Factors affecting ERP system adoption: a comparative analysis between SMEs and large companies. Journal of Enterprise Information Management 18(4), 384–426 (2005)
16. Fui-Hoon Nah, F., Lee-Shang Lau, J., Kuang, J.: Critical Factors for Successful Implementation of Enterprise Systems. Business Process Management Journal 7(3), 285–296 (2001)
17. Thayer, R.H., Fairley, R.: Project Management. In: Marciniak, J. (ed.) Encyclopedia of Software Engineering, pp. 900–923. John Wiley, New York (1994)
18. Vogt, C.: Intractable ERP: a comprehensive analysis of failed enterprise-resource-planning projects. SIGSOFT Softw. Eng. Notes 27(2), 62–68 (2002)

19. Stensrud, E.: Alternative Approaches to Effort Prediction of ERP Projects. Inf. & Soft. Techn. 43(7), 413–423 (2001)
20. Boehm, B.: Software Cost Estimation with COCOMO II. Prentice Hall, Upper Saddle River (2000)
21. Jorgensen, M., Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies. IEEE Trans. Softw. Eng. 33(1), 33–53 (2007)
22. Daneva, M.: ERP Requirements Engineering Practice: Lessons Learnt. IEEE Software 21(2), 26–33 (2004)
23. Myrtveit, I., Stensrud, E.: A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models. IEEE Trans. Softw. Eng. 25(4), 510–525 (1999)
24. Fewster, R.M., Mendes, E.: Portfolio management method for deadline planning. In: Proceedings of METRICS 2003. IEEE, Los Alamitos (2003)
25. Boehm, B., Sullivan, K.J.: Software economics: a roadmap. In: Proceedings of the Conference on The Future of Software Engineering. ACM, Limerick (2000)
26. Boehm, B.: Software Engineering Economics. In: Advances in Computing Science & Technology, Prentice Hall, Upper Saddle River (1981)
27. van Brocke, J., Schenk, B., Sonnenberg, C.: Classification Criteria for Governing the Implementation Process of Service-Oriented ERP Systems - An Analysis Based on New Institutional Economics. In: AMCIS, San Francisco (2009)
28. Lindley, J.T., Topping, S., Lindley, L.T.: The hidden financial costs of ERP software. Managerial Finance 34(2), 78–90 (2008)
29. Kim, G., Seo, D., Kang, K.: Hybrid models of neural networks and genetic algorithms for predicting preliminary cost estimates. Journal of Computing in Civil Engineering 19, 208 (2005)
30. Rush, C., Roy, R.: Analysis of cost estimating processes used within a concurrent engineering environment throughout a product life cycle. In: International Conference on Concurrent Engineering: Research and Applications, p. 58. CRC, Boca Raton (2000)
31. Daniel, L.S., Robert, E.M.: Toward a Model of Consolidation: The Retention and Transfer of Neural Net Task Knowledge. In: Proceedings of the INNS World Congress on Neural Networks. Lawrence Erlbaun Assosciates, Mahwah (1995)
32. Towell, G., Shavlik, J.W.: Interpretation of artificial neural networks: Mapping knowledge-based neural networks into rules. In: Advances in Neural Information Processing Systems (1993)
33. Teoh, E.J., Tan, K.C., Xiang, C.: Estimating the Number of Hidden Neurons in a Feedforward Network Using the Singular Value Decomposition. IEEE Transaction on Neural Network 17, 1623–1629 (2006)