Computational Intelligence in Empirical Software Engineering

Ali Idri
ENSIAS, BP. 713, Agdal, Rabat, University Mohamed V, Morocco,
Email: idri@ensias.ma

Alain Abran
Ecole de technologie Supérieure, 1100, Notre-dame Ouest, Montréal,
Email: aabran@ele.etsmtl.ca

Taghi M. Khoshgoftaar
Empirical Software Engineering Laboratory, Florida Atlantic University, Boca Raton,
Email: taghi@cse.fau.edu

Abstract

The objective of Empirical Software Engineering is to improve the software development and maintenance processes and consequently the quality of theirs various deliverables. This can be achieved by evaluating, controlling and predicting some important attributes of software projects such as development effort, software reliability, and programmers productivity. One of the most interesting sub-field of ESE is software estimation models. Software estimation models are used to predict some critical attributes of some entities that are not yet exist. For example, we often need to predict how much a development project will cost, or how much time and effort will be needed., so that we can allocate the appropriate resources to the project. In general, estimation models relate the attribute to be predicted to some other attributes, that we can measure now, by using mathematical formulas or other techniques such as neural networks, case-based reasoning, regression trees and rule-based induction. Currently, our research concerns software cost estimation models. We have developed an innovative approach referred to as Fuzzy Analogy for software cost estimation. Nevertheless, this approach can be used to evaluate and predict other attributes such as reliability, quality, safety, and maintainability. In this paper, we present some results of our recent research related to the cost estimation field.

1. Cost estimation: Techniques and challenges

Accurate and timely prediction of the development effort and schedule required to build and/or maintain a software system is one of the most critical activities in managing software projects, and has come to be known as '*Software Cost Estimation*'. In order to achieve accurate cost estimates and minimize misleading (under- and over-estimates) predictions, several cost estimation techniques have been developed and validated. These techniques may be grouped into two major categories:

- parametric models, and
- non-parametric models.

Parametric models are derived from statistical or numerical analysis of historical projects data (simple/multiple/stepwise regression, Bayesian approach, polynomial interpolation, …)

and are illustrated by estimations models such as COCOMO [7,8], PUTNAM-SLIM[37], and function points analysis [2,4,30]. There are two main disadvantages to these models. First, they make an assumption on the form of the prediction function, represented by: $Effort = a \times size^{b}$ where α represents a productivity coefficient and β an economies (or diseconomies) of scale coefficient. Second, they need to be adjusted or calibrated to local circumstances.

Non-parametric models are based on computational intelligence techniques such as analogy-based reasoning, artificial neural networks, regression trees, and rule-based induction. They have been developed to avoid the above mentioned shortcomings. Recently, many researchers have begun to turn their attention to this alternative[5,32,35,36,43-47,49,53,54]. This alternative has two significant advantages: First, the capability to model the complex set of relationship between the dependent variable to predict (cost, effort) and the independent variables (cost drivers) collected earlier in the lifecycle. Second, the capability to learn from historical projects data (especially for neural networks).

Experience has shown that there does not exist a 'best' prediction technique outperforming all the others in every situation. Indeed, Shepperd et al., Niessink and Van Vliet found that estimation by analogy generated better results than stepwise regression [34,45,46]. However, Briand et al., Stensrud and Myrtveit reported opposite results [9,32]. Recent research has been initiated to explain the relationship between different properties of historical projects dataset (size, number of attributes, presence of outliers, etc.) and the accuracy of a prediction system [46]. Beyond this interesting issue, we have identified three other challenges that cost estimation community must tackle to improve the existing models:

- Cost estimation models must be able to deal with vague information. Indeed, most of the software project attributes are measured on a scale composed of linguistic values such as *low* and *high*. For example, the well-known COCOMO'81 model has 15 attributes out of 17 (22 out of 24 in the COCOMO II) that are measured with six linguistics values: *very low*, *low*, *nominal*, *high*, *very high*, and *extra-high* [7,8].

- Cost estimation models must appropriately handle the uncertainty in estimates.

- Cost estimation models must learn from previous situations because software development technology is continuously evolving.

Currently, no cost estimation model has integrated in its modeling process the above three criteria. In our recent research, we have developed an estimation model based on reasoning by analogy, fuzzy logic, and possibility theory to satisfy to two first criteria. Further research work has been initiated to look at the integration of the third criterion, concerning learning capabilities, in our model.

2. Computational Intelligence: Case-Based Reasoning and Fuzzy Logic

Computational intelligence is concerned with the computational modeling of human intelligence. Its major objective is to attack the problem of understanding intelligence in computational terms. This idea comes from the first decade of Artificial Intelligence when Alan Turing has published his monumental paper entitled "Computing Machinery and Intelligence". Since that, two approaches known as symbolic and connectionist debate this idea [33,38,39,40,41,48]. Until the last decade, both approaches progressed independently.

Recently, many researchers started investigating ways of integrating both approaches[11,50,51]. In our research, we use a set of techniques inspired of both symbolic and connectionist approaches to develop an intelligent cost estimation model.

2.1 Case-Based Reasoning

Case-Based Reasoning (CBR) is a technology that is especially useful when there is limited domain knowledge and when an optimal solution process to the given problem in not known. Part of the computational intelligence field, it has proven useful in a wide variety of domains, including software quality classification [10], and software fault prediction [27]. The four primary steps comprising a CBR estimation system are [1,29]:

1-Retrieve the most similar case or cases, i.e., previously developed projects.
2-Reuse the information and knowledge represented by the case(s) to solve the estimation problem.
3-Revise the proposed solution.
4-Retain the parts of this experience likely to be useful for future problem solving.

In the context of software cost estimation, a CBR system is based on the assumption that '*similar software projects have similar costs*'. Following this simple yet logical assumption, a CBR system can be employed as follows. Initially, each software project (both historical and candidate projects) must be described by a set of attributes that must be relevant and independent of each other. Subsequently, the similarity between the candidate project and each project in the historical database is determined. Finally, the known development-effort values of historical (previously developed similar) projects is used to derive, i.e., case adaptation, an estimate for the new project.

There are two main advantages of analogy-based estimation: first, it is easy to understand and to explain its process to the users, and second, it can model a complex set of relationships between the dependent variable (such as, cost or effort) and the independent variables (cost drivers). However, its deployment in software cost estimation still warrants some improvements. Human intelligence comes in large part from the ability to reason by analogy. But human reasoning by analogy is always approximate rather than precise. Indeed, the human mind can handle imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness, and low solution cost. According to Zadeh, "the exploitation of these criteria underlies the remarkable human ability to understand distorted speech, decipher sloppy handwriting, drive a vehicle in dense traffic and, more generally, make decisions in an environment of uncertainty and imprecision" [61].

Zadeh's idea is to mimic the ability of the human mind in order to increase the Machine Intelligence Quotient (MIQ) of the new industrial products (microwave, washing machines, software, etc.). Thus, for more than 30 years, Zadeh has been involved in the foundation of a new strategy of computing that is different from the traditional (hard) computing. Zadeh's effort in this area begins with his paper on fuzzy sets followed by the paper on possibility theory, the paper on soft computing, and more recently his papers on computing with words [58-62]. Our intelligent cost estimation model is based on CBR technique to which we integrate fuzzy logic in order to deal with uncertainty and imprecision (like the human brain) in the analogy process

2.2 Fuzzy Logic

Since its foundation by Zadeh in 1965, Fuzzy Logic (FL) has been the subject of important research investigations. During the early nineties, fuzzy logic was firmly grounded in terms of its theoretical foundations and application in the various fields in which it was being used, such as robotics, medicine, and image processing.

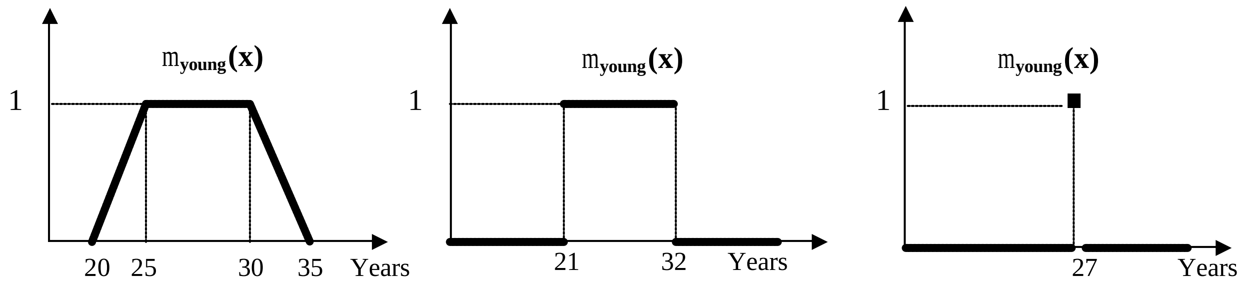
A fuzzy set is a set with a graded membership function, μ, in the real interval [0, 1]. This definition extends the one of a classical set where the membership function is in the couple {0, 1}. Fuzzy sets can be effectively used to represent linguistic values such as *low*, *young*, and *complex*, in the following two ways [25]:

- Fuzzy sets that model the gradual nature of properties, i.e., the higher the membership that a given property x has in a fuzzy set A, the more it is true that x is A. In this case, the fuzzy set is used to model the vagueness of the linguistic value represented by the fuzzy set A.

- Fuzzy sets that represent incomplete states of knowledge. In this case, the fuzzy set is a possibility distribution of the variable X, and consequently, is used to model the associated uncertainty. When considering that it is only known that x is A, and x is not precisely known, the fuzzy set A can be considered as a possibility distribution, i.e., the higher the membership x' has in A, the higher the possibility that x = x'.

For example, consider the linguistic value *young* (for attribute Age) that can be represented in three ways: by a fuzzy set, i.e., Figure 1 (a), by a classical interval, i.e., Figure 1 (b), and by a numerical value, i.e., Figure 1 (c). The representation by a fuzzy set is more advantageous than the other two approaches, because:

- It is more general,
- It mimics the way in which the human-mind interprets linguistic values, and
- The transition from one linguistic value to a contiguous linguistic value is gradual rather than abrupt.



(a) **(b)** **(c)**

Figure 1: (a) Fuzzy set representation of the *young* linguistic value. (b) Classical set representation of the *young* linguistic value. (c) Numerical representation of the *young* linguistic value.

3. Work In Progress

We have initiated preliminary investigations of applying computational intelligence techniques such as case-based reasoning, fuzzy sets, fuzzy inference, soft computing, possibility theory and fuzzy rule-based systems to software cost estimation. Promising results of initial studies has encouraged our team to start further advanced development and exhaustive empirical validation. A summary of these preliminary works are now presented.

COCOMO cost model using fuzzy logic. In this work, 15 cost drivers used in the intermediate COCOMO'81 are represented by means of fuzzy sets instead of classical sets. The study has shown that the representation by fuzzy sets, unlike the one using classical sets, tolerates imprecision in the input (cost drivers) of the intermediate COCOMO'81 and consequently it generates gradual outputs (costs). The approach was tested on three data sets and was shown to yield flexible and stable cost estimations [14,15]

Software project similarity measures based on fuzzy logic. The similarity of two software projects, which are described and characterized by a set of attributes, is often evaluated by measuring the distance between these two projects through their sets of attributes. The way in which the similarity of software projects is gauged is fundamental to the estimation of software cost estimation by CBR. In cost estimation, most researchers have used the Euclidean distance when projects are described by numerical data and the equality distance when they are described by linguistic values. However, the concept of similarity measures has also been discussed in fuzzy logic and in cognitive science [52]. To overcome the limitations of the similarity measures used in the cost estimation literature, we have proposed a set of new software project similarity measures based on fuzzy logic [16]. These measures evaluate the overall similarity of two projects P_1 and P_2, $d(P_1, P_2)$, by combining the individual similarities of P_1 and P_2 associated with the various linguistic variables (attributes) (V_j) describing P_1 and P_2, $d_{v_j}(P_1, P_2)$. After an axiomatic validation of some proposed candidate measures for the individual distances $d_{v_j}(P_1, P_2)$, we have retained two measures that use max-min and sum-product aggregations [17]. To evaluate the overall distance of P_1 and P_2, the individual distances $d_{v_j}(P_1, P_2)$ are aggregated using Regular Increasing Monotone (RIM) linguistic quantifiers such as '*all*', '*most*', '*many*', '*at most a*' or '*there exists*'. The choice of the appropriate RIM linguistic quantifier, Q, depends on the characteristics and the needs of each environment. Q indicates the proportion of individual distances that we feel is necessary for a good evaluation of the overall distance. The overall similarity of P_1 and P_2, $d(P_1, P_2)$ is given by one of the following formulas [18]:

$$d(P_1,P_2)=\begin{cases} all\,of\ (d_{v_j}(P_1,P_2)) \\ most\,of\ (d_{v_j}(P_1,P_2)) \\ many\,of\ (d_{v_j}(P_1,P_2)) \\ .... \\ there\,exists\,of\ (d_{v_j}(P_1,P_2)) \end{cases}$$

When choosing the appropriate RIM linguistic quantifier to guide the aggregation of the individual distances, its implementation is realized by an Ordered Weight Averaging operator [55-57].

Fuzzy Analogy for software cost estimation. We have proposed a new approach, referred to as Fuzzy Analogy, for software cost estimation [19]. It is based on CBR and fuzzy set theory. It can be used when software projects are described either by numerical or linguistic values. Fuzzy Analogy is a fuzzfication of the analogy procedure. It is also composed of three steps:

identification of cases, retrieval of similar cases and cases adaptation. Each step is a fuzzification of its equivalent in the classical analogy procedure:

- In the identification of cases step, the software projects can be described either by numerical or linguistic values. For linguistic values, we have represented them by fuzzy sets rather than classical intervals. The representation of the linguistic values by fuzzy sets has three advantages over the one using classical sets (1) it is more general, (2) it mimics the way in which human interpret these values, (3) and the transition from one linguistic value to the following is gradual rather than abrupt.

- In the retrieval of similar cases step, we have used our software projects similarity measures in order to take into account the case when software projects are described by linguistic values.

- In the cases adaptation step, we have proposed a new strategy to choose similar projects that will be used in the adaptation. Also, we have proposed how to adapt the chosen projects in order to generate an estimate for the new project.

We have validated this approach with the COCOMO'81 data set. The obtained results were compared with three other models: original intermediate COCOMO'81, fuzzy intermediate COCOMO'81, and classical CBR estimation model. Fuzzy Analogy performs better in terms of accuracy and adequacy in dealing with linguistic values [20,23].

Handling uncertainty in Fuzzy Analogy: We have improved Fuzzy Analogy by integrating the uncertainty criterion in its process. Managing the uncertainty in Fuzzy Analogy means that it can produce a set of estimates, rather than only one, with a possibility distribution. This set can be used to deduce, for practical purposes, a point estimate for the cost, and analyzing the risks associated with all possible estimates [24].

4. Future work

By using fuzzy logic and the possibility theory in its estimation process, Fuzzy Analogy satisfies the two first criteria, i.e., the tolerance of imprecision when describing software project, and the uncertainty when estimating the development cost. The third criterion of an intelligent model that Fuzzy Analogy has not yet incorporated into its process is to learn from previous experiences. The learning criterion is required for any cost estimation model. Indeed, the software industry is continuously evolving: engineers use more and more high level programming languages, application generators, web-based technology, etc.

We have initiated the inclusion of some learning functionality in our approach. In the identification step, we can update all information concerning the linguistic variables describing software projects, specifically, their linguistic values that depend on human judgement. For example, the linguistic value *high* for software reliability may mean that the failure intensity is lower than 6 per month, but in the future, we may require less than 3 software failures per month to evaluate it as *high*. In the case retrieval step, we can update the definition of the linguistic quantifier used in the environment. Here also, the meaning of a linguistic quantifier depends on human judgement. However, other learning characteristics that are not included in our approach remain to be examined. For example, Fuzzy Analogy must be able to provide its user with a subset of linguistic variables that have always led to accurate estimates in the past. We may then use this subset in the 'identification of cases' step of Fuzzy Analogy. Thus, the attribute selection problem can also be addressed. Moreover, Fuzzy Analogy must be able to identify the appropriate linguistic quantifier to be used in 'retrieval of similar cases' step by detecting those that have often yielded accurate cost estimates.

Also, neural networks can be used in Fuzzy Analogy to integrate a supervised learning approach by comparing the estimated and the actual values. However, there are some shortcomings that prevent neural networks from being well accepted in cost estimation modeling. The most important is that they are considered as a 'black boxes'. Consequently, it is not easy to understand and explain their process. To avoid this limitation, we have studied the interpretation of a cost estimation model based on a Backpropagation three-layer Perceptron network. This study has shown promising results [21].

5. Bibliography

[1] Aamodt and E. Plaza. "Case-Based Reasoning: Foundational Issues, Methodological Variations. and System Approaches", AI Communications, IOS Press, vol. 7:1, pp. 39-59, 1994.

[2] A. Abran and P.N. Robillard. "Functions Points Analysis: An Empirical Study of its Measurement Processes", IEEE Transactions on Software Engineering, 22(12): pp. 895-909, 1996.

[3] W. D. Aha, "Case-Based Learning Algorithms", DARPA Case-Based Reasoning Workshop, Morgan Kaufmann, LoaAtlos, CA, 1991

[4] Albrecht, A.J., Gaffney, J.E., "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation", IEEE Transactions on Soft Eng, Vol. SE-9, No. 6, Nov, 1983, pp. 639-647

[5] L. Angelis and I. Stamelos. "A Simulation Tool for Efficient Analogy Based Cost Estimation", Empirical Software Engineering, vol. 5, no. 1, pp. 35-68, 2000.

[6] Bisio R., Malabocchia F., "Cost estimation of software project through case base reasoning", First International conf. on Case-Based reasoning, Sesimbra, 1995, Springer pp.11-22

[7] B.W. Boehm. "Software Engineering Economics", Prentice-Hall, 1981.

[8] B.W. Boehm et. al. "Cost Models for Future Software Life Cycle Processes: COCOMO II". Annals of Software Engineering: Software Process and Product Measurement, Amsterdam, 1995.

[9] L. Briand, T. Langley, and I. Wieczorek.. "Using the European Space Agency Data Set: A Replicated Assessment and Comparison of Common Software Cost Modeling". In Proceedings of the 22nd International Conference on Software Engineering, Limerik, Ireland, 2000, pp. 377-386

[10] K. Ganesan, T. M. Khoshgoftaar, and E. Allen. "Case-Based Software Quality Prediction", International Journal of Software Engineering and Knowledge Engineering, 10(2):139-152, 2000.

[11] S. I Gallant, "Neural Network Learning and Expert Systems", MIT Press, Cambridge, 1993

[12] J. K. Holyoak, P. Taghard, "Mental Leaps: Analogy in Creative Thought', MIT Press, 1995.

[13] R.T Hughes, "An Evaluation of Machine Learning Techniques for Software Effort Estimation", University of Brighton, 1996

[14] A. Idri, L. Kjiri, and A. Abran. "COCOMO Cost Model Using Fuzzy Logic", In Proceedings of the 7th International Conference on Fuzzy Theory and Technology, Atlantic City, NJ, February, 2000, pp. 219-223.

[15] A. Idri, A. Abran, "La mise en valeur d'une approche floue pour l'évaluation du coût de développement de logiciels", CARI, Madagacscar, 16-19 Octobre, 2000 , pp.247-254

[16] A. Idri and A. Abran. "Towards A Fuzzy Logic Based Measures for Software Proje ct Similarity", In Proceedings of the 6th Maghrebian Conference on Computer Sciences, November, Fes, Morroco, 2000, pp. 9-18,

[17] A. Idri and A. Abran. "A Fuzzy Logic Based Measures For Software Project Similarity: Validation and Possible Improvements", In Proceedings of the 7th International Symposium on Software Metrics, April, London, IEEE Computer Society, 2001, pp. 85-96

[18] A. Idri and A. Abran. "Evaluating Software Projects Similarity by Using Linguistic Quantifier Guided Aggregations", In Proceedings of the 9th IFSA World Congress and 20th NAFIPS International Conference, July, Vancouver, 2001, pp. 416-421,

[19] A. Idri, A. Abran, and T. M. Khoshgoftaar. "Fuzzy Analogy: A new Approach for Software Cost Estimation", In Proceedings of the 11th International Workshop on Software Measurements, August, Montreal, 2001, pp. 93-101,

[20] A. Idri, A. Abran, T. Khoshgoftaar, S. Robert, "Estimating Software Project Effort by Analogy based on Linguistic Values", 8th International Symposium on Software Metrics, IEEE computer Society, June, Ottawa, 2002, pp.21-30

[21] A. Idri, T. Khoshgoftaar, A. Abran, S. Robert, "Can Neural Networks be easily Interpreted in Software Cost Estimation?", Fuzz-IEEE, 12-17 May, Hawai, 2002, pp. 1162-1166

[22] A. Idri, S. Mbarki, A. Abran, S. Robert, "Interprétation des réseaux de neurones en estimation du coût de logiciels ", to be presented at CARI, Octobre, Cameroun, 2002

[23] A. Idri, A. Abran, T. Khoshgoftaar, "Fuzzy Case-Based Reasoning Models for Software Cost Estimation", to appear in the book Soft Computing in Software Engineering: Theory and Applications, published by Springer-Verlag, 2002

[24] A.Idri, A. Abran, T. Khoshgoftaar, S. Robert, "Investigating Soft Computing in Case-Based Reasoning for Software Cost Estimation", International Journal of Engineering Intelligent Systems, vol. 10, no.3, September, 2002, pp.147-157

[25] R. Jager. "Fuzzy Logic in Control", Ph.D. Thesis, Technic University Delft, 1995, Holland.

[26] J. S. R. Jang, C. T. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing", Prentice-Hall, 1997

[27] T. M. Khoshgoftaar, B. Cukic, and N. Seliya. "Predicting Fault-Prone Modules in Embedded Systems Using Analogy-Based Classification Models". International Journal of Software Engineering and Knowledge Engineering : Special Volume on Embedded Software Engineering, 12(1):1-22, 2002.

[28] G. Kadoda, M. Cartwright, L. Chen, and M. Shepperd. "Experiences Using Case-Based Reasoning to Predict Software Project Effort", In Proceedings of EASE, p.23-28, 2000, Keele, UK.

[29] J.L. Kolodner, "Case-Based Reasoning", Morgan Kaufmann, 1993.

[30] J. Matson, E. B. E. Barrett, J. M. Mellichamp. "Software Development Cost Estimation Using Function Points", Transactions on Software Engineering, vol. 20, no. 4, pp. 275-287, April 1994, IEEE Computer Society.

[31] L. R. Medsker, "Hyprid Neural Networs and Expert Systems", Kluwer Academic Publishers, 1994

[32] I. Myrtveit and E. Stensrud. "A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models", Transactions on Software Engineering, vol.

[33] A. Newell, "Unified Theory of Cognition", Harvard University Press, 1990

[34] F. Niessink and H. Van Vliet. "Predicting Maintenance Effort with Function Points", In Proceedings of the International Conference on Software Maintenance, 1997, Bari, Italy,

[35] L. A. Porter, R. Selby, "Empiracally-Guided Software Development using Metric-based Classification Trees" , IEEE Software, Vol. 7, March, 1990, pp. 46-54

[36] A. Porter, R. Selby, "Evaluating Techniques for Generating Metric-Based Classification

[37] L. H. Putnam "A General Empirical Solution to the Macro Software Sizing and Estimation Problem", Transactions on Software Engineering, vol. 4, no. 4, July, 1978.

[38] Zenon W. Pylyshyn, "Computation and Cognition: Towards a Foundation for Cognitive Science", MIT Press, 1984

[39] Zenon W. Pylyshyn, "Computing in cognitive Science, Foundations of Cognitive Sciences", Ed. By M. I. Posner, MIT press, 1989, pp. 51-91

[40] David E. Rumelhart, J. McClelland, "Parralel Distributed Processing", Explorations in the Microstructure of Cognition, Vol. 1: Foundations, Cambridge, MIT Press, 1986

[41] David E. Rumelhart, "The Architecture of Mind : A connectionist Approach, Foundations of cognitive Sciences", 1989, pp.133-159

[42] R. Selby, A. Porter, "Learning from exemples: Generation and evaluation of Decion Trees for Software Resource Analysis", IEEE Transactions on Software Engineering, Vol. 14, 1988. pp. 1743-1757

[43] C. Serluca, "An Investigation into Software Effort Estimation using a Backpropagation Neural Network", M.Sc. Thesis, Bournemouth University, 1995

[44] M. Shepperd, C. Schofield, and B. Kitchenham. "Effort Estimation using Analogy", In Proceedings of the 18th International Conference on Software Engineering, 1996, pp. 170-178, Berlin.

[45] M. Shepperd and C. Schofield. “Estimating Software Project Effort Using Analogies”, Transactions on Software Engineering, vol. 23, no. 12, pp. 736-743, November 1997. IEEE Computer Society.
[46] M. Shepperd and G. Kadoda. “Using Simulation to Evaluate Predictions Systems”, In Proceedings of the 7th International Symposium on Software Metrics, pp. 349-358, April 2001, England, UK, IEEE Computer Society.
[47] K. K. Shukla, “Neuro-Genetic Prediction of Software Development Effort”, Information Software Technology, Vol. 42, 2000, pp. 701-713
[48] P. Smolensky, “On the Proper Treatment of Connectionnism”, Behavioral and Brain Sciences,
[49] K. Srinivasan, Fisher D, “Machine Learning Approaches to Estimating Software Development Effort”, IEEE Transaction on Software Engineering, Vol. 21, No. 2, February, 1995, pp. 126-136
[50] H. Tirri, “Implementing Expert System Rule Conditions by Neural Networks”, New Generation Computing, Vol. 10, 1991, pp. 55-71
[51] G. G. Towell, “Symbolic Knowledge and Neural Networks: Insertion, Refinement, and Extraction”, Ph.D dissertation, University of Wisconsin, 1992
[52] A. Tversky**,** “Features of Similarity”, Psycological Review, 84:327-352
[53] S. Vicinanza and M.J. Prietulla. “Case-Based Reasoning in Software Effort Estimation”, In Proceedings of the 11th International Conference on Information Systems, 1990.
[54] G. Wittig, G. Finnie, ‘Estimating Software Development Effort with connectionist Models’, Information and Software Technology, vol. 39, 1997, pp. 469-476
[55] R. R. Yager and J. Kacprzyk. “The Ordered Weighted Averaging Operators: Theory and Applications”, Kluwer Academic Publishing, Norwell, MA, 1997.
[56] R. R. Yager. “Quantifier Guided Aggregation using OWA Operators”, International Journal of Intelligent Systems, vol. 11, pp.49-73, 1996.
[57] R. R. Yager, “Fuzzy constraint Satisfaction for E-commerce Agents”, 7th International Conference on Fuzzy Theory and Technology, Atlantic City, February, 2000, pp. 111-114
[58] L. A. Zadeh. “Fuzzy Set”, Information and Control, vol. 8, pp. 338-353, 1965.
[59] L. A. Zadeh, “A theory of approximate reasoning”, Machine Intelligence 9, 1979, pp. 149-194
[60] L.A. Zadeh, “Fuzzy Sets as a basis for a theory of possibility”, Fuzzy Sets and Systems 1, 1979, pp. 3-28
[61] L. A. Zadeh. “Fuzzy Logic. Neural Networks and Soft Computing”, Communications of ACM, vol. 37, no. 3, pp.77-84, March 1994.
[62] L. A. Zadeh, “From computing with numbers to computing with words- from manipulation of measurements to manipulation of perceptions”, Computing with words, Ed. Paul P. Wang, John Wiley&Sons, 2001
[63] L.A. Zadeh, “Fuzzy Logic = Computing with Words”, IEEE Transactions on Fuzzy Systems , 1996, pp. 103-111