

# Fuzzy Decision Tree Approach for Embedding Risk Assessment Information into Software Cost Estimation Model

SUN-JEN HUANG, CHIEH-YI LIN AND NAN-HSING CHIU

*Department of Information Management  
National Taiwan University of Science and Technology  
Taipei, 106 Taiwan*

As software cost drivers are fuzzy and uncertain, software cost estimates are prone to a certain degree of estimation errors especially in their early stages of software development life cycle. However, most of the existing software cost estimation models in present literature only generate a single point estimate and do not explicitly reveal the degree of risks caused by their inaccuracies. This paper proposes a fuzzy decision tree approach for embedding risk assessment information into a software cost estimation model. Using this model, one may be able to determine the software cost estimate as well as the estimation error in the form of a fuzzy set. In verifying the merits of this model, we have used the 63 historical project data in the COCOMO model. The validation result shows that our proposed model reveals the risk assessment of the generated software cost estimate, and at the same time yields an even more accurate result as compared to the original COCOMO model.

**Keywords:** software cost estimation model, risk assessment, software project planning, fuzzy decision tree, software measurement and analysis

## 1. INTRODUCTION

Accuracy and consistency in estimating software development effort or cost are very important for project managers in planning and conducting software development activities well, especially in the early stages of the software development life cycle. The importance of accuracy and consistency is reflected by the proliferation of software cost estimation models (refer to Section 2 of this paper) in the last three decades. These models aimed for precision of their generated software cost estimates which greatly influence the software price determination, resource allocation, schedule arrangement, and progress monitoring. Furthermore, to meet the demand for accuracy in software cost estimation, many techniques have been proposed including statistic regression and variations [6], analogy and case-based reasoning [24], fuzzy logic [10] and neural network [29].

However, as varied as they may seem, most model-building techniques employed in software cost estimation models involve analytical equations. These analytical equations relate the software project cost to a number of input parameters using historical data on the past software projects. These input parameters are known as cost drivers, because they influence the required cost of software projects. Some examples of cost drivers in-

---

Received July 1, 2005; accepted November 24, 2005.  
Communicated by Sung Shin.

clude software size and complexity, team capability and experience, platform constraints, use of modern programming, and volatility of requirements [4, 17].

In constructing a software cost estimation model, the software cost drivers need to be measured first. Unfortunately, these drivers are vague, thus causing software development cost estimates to be often associated with uncertainty and inaccuracy [23]. For example, “software complexity” has a wide range of definitions and, as such, one may find several different quantitative software metrics existing in recent literature. Furthermore, mistakes in human judgment frequently happen especially when converting the measures of software cost drivers into the ordinal scale type during the construction of models. This is because qualitative and quantitative measures used for software cost drivers do not have standardized definitions.

Aside from the uncertainty that comes with software cost drivers, the predictive accuracy may also be significantly affected by the data used to build models [18, 28]. It is important to note that software project data has characteristics that make estimation difficult, such as missing data, heteroscedasticity, and outliers [9]. Such a difficulty may also be reflected by the fact that, despite extensive research on the comparative evaluations of the predictive accuracy of different techniques, only few tangible conclusions may be drawn based on existing results [7, 8]. Furthermore, the accuracy of cost estimates varies in different software development stages. Interested readers can refer to Boehm’s work for the effect of project uncertainties on the accuracy of software size and cost estimates [4].

To address the inevitability of software cost estimation errors, existing models have been produced, providing a single point estimate for the required software cost. However, the single point estimate does not explicitly reveal the degree of risks caused by inaccuracies. It is important to note that the risk assessment information reveals the level of uncertainty in the software development project and is therefore very important for project managers in allocating and distributing resources appropriately in the early software development life cycle. As such, this study is motivated by the necessity of explicitly embedding the risk assessment information into a software cost estimation model.

This paper proposes a new method for building a software cost estimation model that does not only generate a point estimate of the required software development cost, but also reveals the estimation error in the form of a fuzzy set. The adopted model-building technique is the fuzzy decision tree approach, which combines the comprehensibility of rules generated based on the ID3 decision tree and the expressive power of fuzzy sets. We illustrate our approach by using the 63 historical project data in the well-known COCOMO model. The comparisons of the estimation results in our proposed model and the original COCOMO model are also provided in this paper.

## 2. SOFTWARE COST ESTIMATION MODELS

Software cost drivers greatly affect software development effort and cost in the software development life cycle. Among the software cost drivers, the most important one is software size. The primary software size metrics in existing software cost estimation models are lines of code (LOC) and function point (FP). As shown in Table 1, these

**Table 1. Software cost estimation models based on LOC and function point.**

Software Size Metrics	Estimation Models	Equations
LOC	Walston-Felix [30]	$E = 5.2 * (KLOC)^{0.91}$
	Bailey-Basili [2]	$E = 5.5 + 0.73 * (KLOC)^{1.16}$
	Boehm Simple [4]	$E = 3.2 * (KLOC)^{1.05}$
	Boehm Average [4]	$E = 3.0 * (KLOC)^{1.12}$
	Boehm Complex [4]	$E = 2.8 * (KLOC)^{1.20}$
	Doty [12]	$E = 5.288 * (KLOC)^{1.047}$ for $KLOC > 9$
FP	Albrecht and Gaffney [1]	$E = -13.39 + 0.0545 * FP$
	Kemerer [14]	$E = 60.62 * 7.728 * 10^{-8} * FP^3$
	Matson <i>et al.</i> [19]	$E = 585.7 + 15.12 * FP$

**Table 2. Software cost estimation models based on model-construction techniques.**

Construction Techniques	Estimation Models
Expert Experience	Expert System [11]
Linear Model	Albrecht and Gaffney [1] Matson, Barnett and Mellichamp [19]
Non-linear Model	COCOMO I [4], COCOMO II [5] Bailey-Basili [2], Doty [12], Kemerer [14], Walston-Felix [30] Boehm Simple [4], Boehm Average [4], Boehm Complex [4]
Artificial Intelligence Model	Neural Network [3], Case-Based Reasoning [20]

diverse cost estimation models are classified into two groups, according to the two software size metrics – LOC and FP. Conte *et al.* [9] and Matson *et al.* [19] have presented an overview of each of these software cost estimation models.

Furthermore, the existing software cost estimation models can also be clustered into several groups according to their model-construction techniques, as shown in Table 2. Although the constructional techniques of these models are different, these models need to first measure all software cost drivers and their generated estimation results are all a single point estimate for the required software development cost.

The study of fuzzy logic started in 1965 by Zadeh, who published the paper “Fuzzy Sets” [32]. Klir also provided a comprehensive overview of fuzzy sets, fuzzy logic, and fuzzy systems in 1995 [16]. Fuzzy logic has been often used as an approximate reasoning technique. Several researchers have also reported progress towards the successful application of using the fuzzy logic technique in assessing software cost to solve the inaccuracies in software cost attributes: Pedrycz *et al.* [22] introduced a fuzzy set approach to cost estimation of software projects while Ryder [25] presented an application of fuzzy modeling techniques to COCOMO and FP models.

Cognitive uncertainties, such as vagueness and ambiguity, have also been incorporated into the induction process by using fuzzy decision trees [31]. The fuzzy decision tree is one of the popular inductive learning methods that include two techniques – fuzzy logic and decision tree. In a fuzzy decision tree, each path from the root node to a termi-

nal node corresponds to a fuzzy rule and a partitioned fuzzy subspace in the whole pattern space.

The introduction of the decision tree is an efficient way of learning from examples. ID3 [26] and C4.5 [27] are the most widely used methods for constructing a decision tree. In 2003, Olaru and Wehenkel [21] introduced a complete fuzzy decision tree technique. The fuzzy decision tree method has been demonstrating its superiority over the popular ID3 in terms of predictive accuracy [13]. It has also been applied to software quality assessment model [23].

However, most of the existing decision tree methods in present literature are based on exact concepts that are weak in handling uncertainty and fuzziness values [26, 27]. Moreover, the hurdle values for attribute segmentation are certain, which is inconsistent with the properties of uncertainty and fuzziness of software cost drivers. Hence, the fuzzy decision tree approach is very suitable in overcoming this shortcoming in software cost estimation domains. In this paper, we adopt the fuzzy decision tree approach to reveal the degree of estimation errors in the form of a fuzzy set in the proposed software cost estimation model.

### 3. MODEL CONSTRUCTION METHODS AND PROCEDURES

The procedures for constructing software cost estimation models integrating risk assessment information by using the fuzzy decision tree approach are shown in Fig. 1. The software project database provides the cost drivers of historical software projects with an estimation model, i.e. COCOMO. The output of software cost estimation model includes a point estimate and a relative error ( $RE$ ) in order to present the degree of estimation risk.  $RE$  is calculated as shown in Eq. (1), where *estimated* is the output of the estimation model for each observation, and *actual* is the actual development cost value. The overall model construction procedures include three major steps: fuzzification, fuzzy decision tree, and reasoning mechanism.

$$RE = \left( \frac{actual - estimated}{actual} \right) \times 100 \% . \quad (1)$$

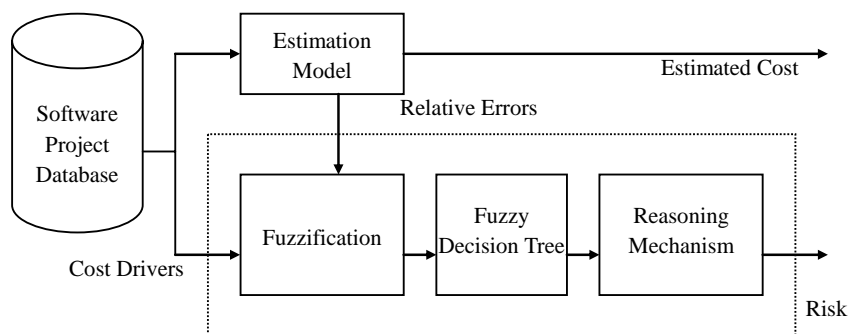


Fig. 1. Model construction procedures.

**Step 1:** Fuzzy the software cost drivers.

As software cost drivers are fuzzy and uncertain, they are first transferred from original assessments to the form of a fuzzy set. All cost drivers of historical software projects may be one of the three scales: nominal, ordinal, and ratio. The fuzzy equations for each of the three scales are respectively shown below.

- (1) **Nominal Scale** The fuzzy equation of software cost drivers with a nominal scale is shown in Eq. (2), where  $x$  is the original assessment class of cost driver  $n$ ;  $i$  is the class of cost driver  $n$ . If an original assessment class  $x$  is equal to class  $i$ , the membership function of  $\mu_{ni}(x)$  is one and the others are zero.

$$\mu_{ni}(x) = \begin{cases} 1, & x = i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

- (2) **Ordinal Scale** The fuzzy equation of software cost drivers with an ordinal scale is shown in Eq. (3), where  $x$  is the original assessment class of cost driver  $n$ ;  $i$  is the class of cost driver  $n$ . If an original assessment class  $x$  is equal to class  $i$ , the membership function of  $\mu_{ni}(x)$  is one. For the assessment class  $x$  is  $i - 1$  and  $i + 1$ , the membership function of  $\mu_{ni}(x)$  is 0.5, and the others are zero.

$$\mu_{ni}(x) = \begin{cases} 0.5, & x = i - 1 \\ 1, & x = i \\ 0.5, & x = i + 1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

- (3) **Rational Scale** For cost drivers with a rational scale, every fuzzy number of the original meaning scale adopts a triangular fuzzy membership function as shown in Eq. (4). A triangular fuzzy membership function can be represented by three values ( $l$ ,  $m$ ,  $u$ ), where  $l$ ,  $m$ , and  $u$  represent the left, center and right points of the triangular membership function.

$$\mu_{ni}(x) = \begin{cases} \frac{x-l_i}{m_i-l_i}, & l_i \leq x \leq m_i \\ \frac{u_i-x}{u_i-m_i}, & m_i \leq x \leq u_i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

**Step 2:** Construct the risk assessment model by the fuzzy decision tree approach.

The fuzzy number of software cost drivers and estimation errors are adapted to construct a fuzzy decision tree for risk assessment. The fuzzy decision tree construction is based on ID3 decision tree technique. ID3 decision tree uses entropy to evaluate the classified contribution of every attribution in a node. The major difference of the ID3 tradi-

tional decision tree and the fuzzy decision tree can be found in the calculation method of the entropy value. In the ID3 fuzzy decision tree, the entropy value of node  $i$  is expressed as  $E^i$  in Eq. (5), where  $C$  is the set of classes and  $P_k^i$  represents the ratio of the patterns of the class  $k$  to all patterns.

$$E^i = - \sum_{k \in C} P_k^i \log_2 P_k^i. \quad (5)$$

In the ID3 traditional decision tree, an input case  $l$  just may or may not belong to node  $i$ . However, the degree of an input case  $l$  belonging to node  $i$  represents  $V_{il}$  in ID3 fuzzy decision tree as shown in Eq. (6), where  $Z_i$  is the attribution group being used from root node to node  $i$  in a decision tree. If  $Z_i$  is an empty group, node  $i$  is the root node.  $X_{lf}$  is the feature value of the attribution  $f$  of the input case  $l$ .  $\mu_{f_i}$  is the degree function of the attribution  $f$  of the input case  $l$ .  $P_k^i$  is the ratio of the patterns of the class  $k$  to all patterns of the fuzzy decision tree as shown in Eq. (7).

$$v_{il} = \begin{cases} \prod_{f \in Z_i} \mu_{f_i}(x_{lf}) & : Z_i \neq \phi \\ 1 & : Z_i = \phi \end{cases}, \quad (6)$$

$$P_k^i = \frac{\sum_{m \in k} v_{im}}{\sum_{l \in D} v_{il}}. \quad (7)$$

The construction procedures of integrating risk assessment information into software cost estimation model by using the fuzzy decision tree approach are presented as follows.

**Step 2.1:** Generate the root node that has a set of all training data.

**Step 2.2:** If one of the following conditions is satisfied, assign node  $i$  as the terminal node.

$$(1) \frac{1}{|D|} \sum_{l \in D} v_{il} \leq \theta_s,$$

$$(2) \frac{\sum_{m \in S(k^*)} v_{im}}{\sum_{l \in D} v_{il}} \geq \theta_d, \text{ where } k^* = \max_{k \in C} \left( \sum_{m \in S(k)} v_{im} \right),$$

(3) No more attributions are available.

**Step 2.3:** Find minimal  $E_f^i$  from an unused attributive group and make child nodes from the node associated with attribution  $f$ .

**Step 2.4:** Go to step 2.2 and apply the algorithm to all newly generated child nodes, recursively.

The threshold parameters  $\theta_s$  and  $\theta_d$  are determined when we terminate the pattern space partitioning.  $\theta_s$  prohibits generating fuzzy rules for sufficiently sparse fuzzy subspaces. On the other hand,  $\theta_d$  prohibits generating fuzzy rules for the minor classes having no sufficient number of patterns in subspace.

**Step 3:** Reason the fuzzy number as the risk estimate of a software project.

Inference in the ID3 decision tree is executed by starting from the root node. Then, repeatedly test the attribute at the node and branch to an edge by its value until a leaf node is reached. A class attached to the leaf node is regarded as the final estimation result. Inference in the fuzzy decision tree must match approximately more than one branch than the traditional ID3 decision tree [15].

It consists of three operations. First, for the operation to aggregate membership values for the path of edges, the multiplication is adopted from many alternatives. Second, for the operation of the total membership value of the path of edge and degree of the class attached to the leaf node, the multiplication is also adopted. Finally, for the operation to aggregate the degree of the same class from the different paths of edges, addition is adopted from several alternatives. As shown in Eqs. (8) and (9),  $r_{jk}$  represents the degree of the software project  $j$  that belongs to the class  $k$ .  $P_k^i$  is the class degree of these rules.  $d_{ji}$  represents the degree of the software project  $j$  belonging the rule  $I$ , while  $r_{jk}^*$  represents the normalization of  $r_{jk}$ . In Eq. (10),  $C_k^m$  represents the  $k$  classes and the scope of  $C_k^m$  is  $l_{C_k^m} \leq x \leq u_{C_k^m}$ ,  $C_k^m \in C$ ,  $1 \leq m \leq k$ . The fuzzy software cost estimate of the software project  $j$  is indicated as  $\mu_j(x)$ .

$$r_{jk} = \sum_i d_{ji} P_k^i, \quad (8)$$

$$r_{jk}^* = \frac{r_{jk}}{\sum_{m \in C} r_{jm}}, \quad (9)$$

$$\mu_j(x) = \begin{cases} \max \left\{ \min(\mu_{C_k^1}(x), r_{jC_k^1}^*), \min(\mu_{C_k^2}(x), r_{jC_k^2}^*), \dots, \min(\mu_{C_k^{m-1}}(x), r_{jC_k^{m-1}}^*) \right\}, \\ \quad \quad \quad x \in C_k^1, C_k^2, \dots, C_k^{m-1} \\ \max \left\{ \min(\mu_{C_k^2}(x), r_{jC_k^2}^*), \min(\mu_{C_k^3}(x), r_{jC_k^3}^*), \dots, \min(\mu_{C_k^m}(x), r_{jC_k^m}^*) \right\}, \\ \quad \quad \quad x \in C_k^2, C_k^3, \dots, C_k^m \\ \quad \quad \quad \vdots \\ \max \left\{ \min(\mu_{C_k^n}(x), r_{jC_k^n}^*), \min(\mu_{C_k^{n+1}}(x), r_{jC_k^{n+1}}^*), \dots, \min(\mu_{C_k^k}(x), r_{jC_k^k}^*) \right\}, \\ \quad \quad \quad x \in C_k^n, C_k^{n+1}, \dots, C_k^k \end{cases} \quad (10)$$

where  $k \geq m \geq 2$ ,  $k \geq n \geq 1$ .

#### 4. AN EMPIRICAL EXAMPLE

To demonstrate the applicability of our proposed model, 63 historical project data in COCOMO were used for constructing a software cost estimation model integrating risk assessment information by means of the fuzzy decision tree approach. We used 62 historical projects for model construction and one historical project for model testing. The software cost drivers in COCOMO are shown in Table 3. The COCOMO model generates effort estimation equation in person-month ( $PM$ ) from 63 historical project data. The risk assessment construction procedures using the fuzzy decision tree are shown in the following steps.

**Table 3. Software cost drivers in COCOMO.**

Development mode	Main storage constraint	Modern programming practices
Adjusted KDSI	Virtual machine volatility	Requirement variation level
Product complexity	Language experience	Required software reliability
Use of software tools	Execution time constraint	Computer turnaround time
Database size	Programmer capability	Virtual machine experience
Analyst capability	Application experience	Required development schedule

**Step 1:** Fuzzy the software cost drivers.

The  $RE$  (in %) in each historical project of COCOMO is fuzzied by 5-class and 7-class models in accordance with the membership functions as shown in Figs. 2 and 3, respectively. The fuzzied  $RE$  attribute is treated as a dependent variable in model-construction procedures using the fuzzy decision tree approach.

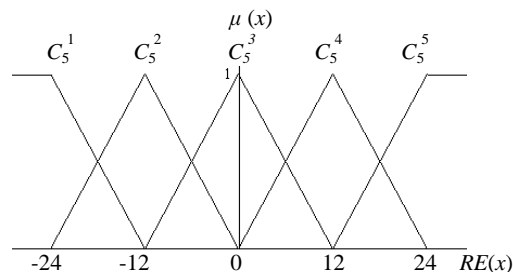


Fig. 2. 5-class triangle fuzzy numbers of  $RE$  value.

The independent variables in the fuzzy decision tree approach are derived from transferring original software cost drivers into fuzzy numbers according to the membership functions as demonstrated in Eqs. (1), (2) and (3). The adjusted KDSI of COCOMO software cost drivers is ratio scale type. It is divided into six groups and fuzzied by Eq. (4) as illustrated in Fig. 4. The other software cost drivers are ordinal scale and thus fuzzied by Eq. (3).

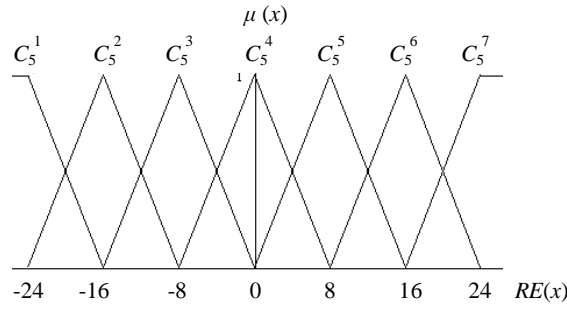


Fig. 3. 7-class triangle fuzzy numbers of *RE* value.

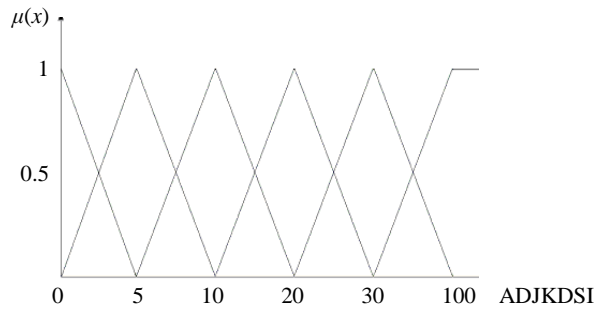


Fig. 4. The triangle fuzzy number of the adjusted KDSI.

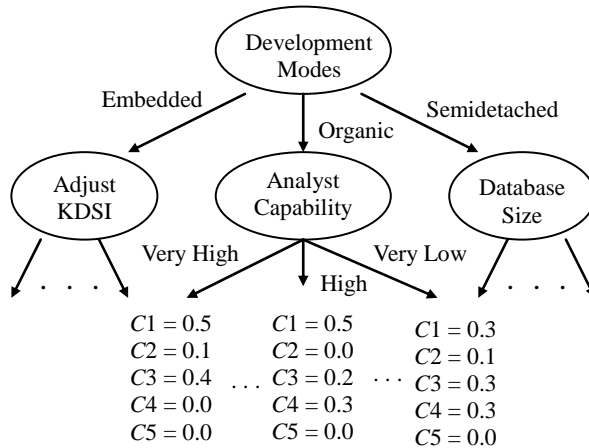


Fig. 5. An example of 5-class fuzzy decision tree model.

**Step 2:** Construct the risk assessment model by fuzzy decision tree approach.

The software cost drivers with different fuzzy numbers are treated as independent variables, and the *RE* fuzzied by 5-class or 7-class models is the dependent variable. The fuzzy decision tree is constructed based on 62 COCOMO projects with these fuzzy software cost drivers. In the fuzzy decision tree, leaf nodes represent multiple classes with

different membership degrees. A demonstration of the fuzzy decision tree within 5-class risk assessment model is shown in Fig. 5, where  $C_i$  represents the membership degree of class  $i$ .

**Step 3:** Reason the fuzzy number as the risk estimate of a software project.

Once a fuzzy decision tree is built from a fuzzy training set, the fuzzy decision tree is used to reason the fuzzy numbers for a test project. The fuzzy decision tree infers the fuzzy numbers as the risk assessment for a test project  $j$  by Eq. (11) as 5-class model and Eq. (12) as 7-class model, respectively. After constructing the fuzzy decision tree, the test data fed into the root node may arrive at several leaf nodes and multiple classes with different membership degrees. The addition operation to aggregate degrees of the same class from the different paths of edges is the risk estimate for that class. The risk estimation results for each class of the 5-class fuzzy decision tree model are  $C_5^1 = 0.1257$ ,  $C_5^2 = 0.1247$ ,  $C_5^3 = 0.2111$ ,  $C_5^4 = 0.0247$  and  $C_5^5 = 0.5137$ . Fig. 6 shows the risk estimation results of a software project based on the estimated fuzzy numbers from the 5-class fuzzy decision tree model.

$$\mu_j(x) = \begin{cases} \min(\mu_{C_5^1}(x), r_{jC_5^1}^*), & x < -24 \\ \max\{\min(\mu_{C_5^1}(x), r_{jC_5^1}^*), \min(\mu_{C_5^2}(x), r_{jC_5^2}^*)\}, & -24 \leq x < -12 \\ \max\{\min(\mu_{C_5^2}(x), r_{jC_5^2}^*), \min(\mu_{C_5^3}(x), r_{jC_5^3}^*)\}, & -12 \leq x < 0 \\ \max\{\min(\mu_{C_5^3}(x), r_{jC_5^3}^*), \min(\mu_{C_5^4}(x), r_{jC_5^4}^*)\}, & 0 \leq x < 12 \\ \max\{\min(\mu_{C_5^4}(x), r_{jC_5^4}^*), \min(\mu_{C_5^5}(x), r_{jC_5^5}^*)\}, & 12 \leq x < 24 \\ \min(\mu_{C_5^5}(x), r_{jC_5^5}^*), & x \geq 24 \end{cases} \quad (11)$$

$$\mu_j(x) = \begin{cases} \min(\mu_{C_7^1}(x), r_{jC_7^1}^*), & x < -24 \\ \max\{\min(\mu_{C_7^1}(x), r_{jC_7^1}^*), \min(\mu_{C_7^2}(x), r_{jC_7^2}^*)\}, & -24 \leq x < -16 \\ \max\{\min(\mu_{C_7^2}(x), r_{jC_7^2}^*), \min(\mu_{C_7^3}(x), r_{jC_7^3}^*)\}, & -16 \leq x < -8 \\ \max\{\min(\mu_{C_7^3}(x), r_{jC_7^3}^*), \min(\mu_{C_7^4}(x), r_{jC_7^4}^*)\}, & -8 \leq x < 0 \\ \max\{\min(\mu_{C_7^4}(x), r_{jC_7^4}^*), \min(\mu_{C_7^5}(x), r_{jC_7^5}^*)\}, & 0 \leq x < 8 \\ \max\{\min(\mu_{C_7^5}(x), r_{jC_7^5}^*), \min(\mu_{C_7^6}(x), r_{jC_7^6}^*)\}, & 8 \leq x < 16 \\ \max\{\min(\mu_{C_7^6}(x), r_{jC_7^6}^*), \min(\mu_{C_7^7}(x), r_{jC_7^7}^*)\}, & 16 \leq x < 24 \\ \min(\mu_{C_7^7}(x), r_{jC_7^7}^*), & x \geq 24 \end{cases} \quad (12)$$

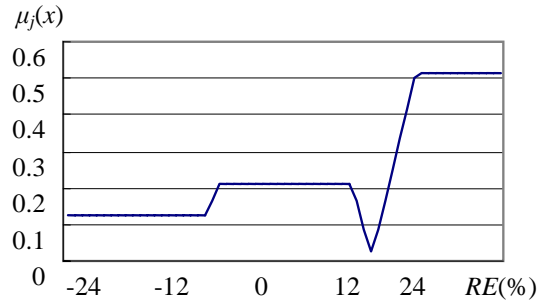


Fig. 6. Risk estimate of a software project.

## 5. VALIDATION

Since the fuzzy numbers of risk assessment in the fuzzy decision tree is not a single point estimate, it cannot be compared with the results of other models. These fuzzy numbers are defuzzified in accordance with the Eqs. (13), (14) and (15), introduced by Boyen and Wehenkel in 1999, where  $d_j^a$  is the single point estimate of the risk assessment of the software project  $j$ . The final estimation result is calculated by Eq. (16), where  $PM$  is the cost estimate in the original COCOMO model and  $PM_{Risk}$  is the cost estimate embedding risk assessment information into the software cost estimation model.

$$f_{left} = \int_{-a}^0 (\mu_j(x) \times x) dx, \quad a \geq 0, \quad (13)$$

$$f_{right} = \int_0^a (\mu_j(x) \times x) dx, \quad a \geq 0, \quad (14)$$

$$d_j^a = \begin{cases} \frac{f_{left}}{\int_{-a}^0 \mu_j(x) dx}, & \text{where } |f_{left}| - f_{right} > \varepsilon \\ \frac{f_{right}}{\int_0^a \mu_j(x) dx}, & \text{where } f_{right} - |f_{left}| > \varepsilon, \\ 0, & \text{where } ||f_{left}| - f_{right}| \leq \varepsilon \end{cases} \quad (15)$$

$$PM_{Risk} = \left(1 + \frac{d_j^a}{100}\right) PM. \quad (16)$$

The adopted indicators for comparing the prediction accuracy of software cost estimation models are Mean Magnitude of Relative Error ( $MMRE$ ) and the prediction accuracy rate at level  $L$  ( $PRED(L)$ ). The widely used and acceptable values for  $MMRE$  and  $PRED(L)$  for a good software cost estimation model are  $MMRE \leq 0.25$  and  $PRED(0.25) \geq 0.75$ . The  $PRED(0.25) \geq 0.75$  means the model should have at least 75% of the predicted values that fall within 25% of their actual values. The  $MMRE$  and  $PRED(0.25)$  are

also applied to evaluate the performance of our proposed software cost estimation model embedded with risk assessment information. The model-construction procedures are repeated 63 times, each using 62 software projects for constructing the model and one remnant project for verifying the estimation accuracy under different stop criteria of  $\theta_s$  and  $\theta_d$ . The verification results with 5-class and 7-class models are shown in Table 4. The proposed software cost estimation model embedding risk assessment model with different stop criteria of  $\theta_s$  and  $\theta_d$  has lower *MMRE* and higher *PRED(0.25)* values than the original COCOMO model in both 5-class and 7-class models. Thus it demonstrates that the model integrating risk assessment information into COCOMO using the fuzzy decision tree approach has improved the estimation accuracy than the original COCOMO model.

**Table 4. The verification results of 5-class and 7-class models.**

Models	Method ID	$\theta_d$	$\theta_s$	5-class model		7-class model	
				MMRE	PRED(0.25)	MMRE	PRED(0.25)
COCOMO	0	–	–	0.1919	0.746	0.1919	0.746
COCOMO + fuzzy decision tree	1	0.6	0.10	0.1761	0.810	0.1815	0.762
	2		0.09	0.1745	0.810	0.1819	0.762
	3		0.08	0.1667	0.810	0.1712	0.810
	4		0.07	0.1653	0.825	0.1716	0.810
	5	0.5	0.10	0.1761	0.810	0.1815	0.762
	6		0.09	0.1745	0.810	0.1819	0.762
	7		0.08	0.1667	0.810	0.1712	0.810
	8		0.07	0.1653	0.825	0.1716	0.810
	9	0.4	0.10	0.1758	0.810	0.1813	0.762
	10		0.09	0.1742	0.810	0.1818	0.762
	11		0.08	0.1669	0.810	0.1712	0.810
	12		0.07	0.1655	0.825	0.1716	0.810
	13	0.3	0.10	0.1671	0.810	0.1807	0.762
	14		0.09	0.1702	0.794	0.1803	0.762
	15		0.08	0.1702	0.794	0.1707	0.794
	16		0.07	0.1706	0.778	0.1684	0.794

The results of *MMRE* of the model integrating risk assessment information into COCOMO using the fuzzy decision tree approach are shown in Fig. 7. In general, the 5-class model has lower *MMRE* than the 7-class model. The stop criteria of  $\theta_s$  with values 0.08 and 0.07, i.e. method ID 3, 4, 7, 8, 11, 12, 15 and 16, have lower *MMRE* than the stop criteria of  $\theta_s$  with values 0.09 and 0.10, i.e. method ID 1, 2, 5, 6, 9, 10, 13 and 14. However, the stop criteria  $\theta_d$  has obviously no difference in improving the model accuracy. In assessing the acceptability of cost estimation models by *MMRE*, the model integrating risk assessment information into COCOMO using the fuzzy decision tree approach with stop criteria  $\theta_s$  of value 0.07 and 0.08 outperforms the other methods.

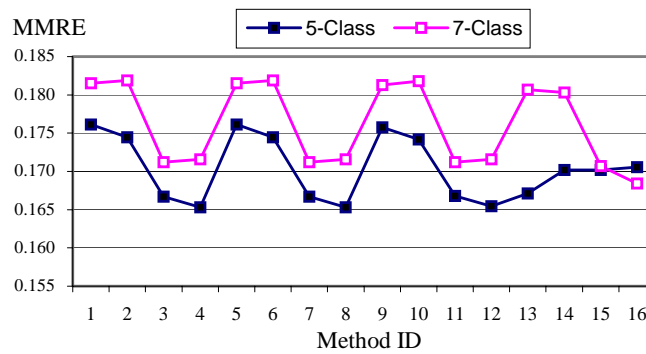


Fig. 7. The *MMRE* in 5-class and 7-class models.

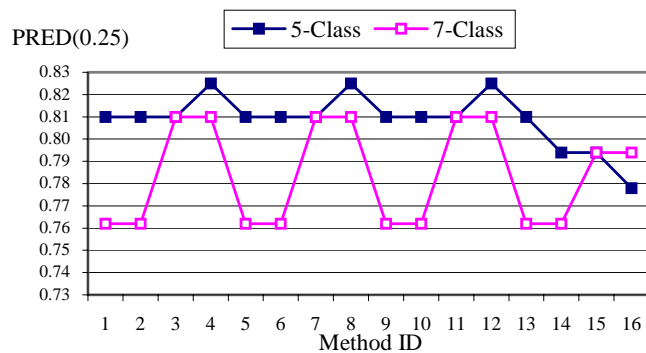


Fig. 8. The *PRED(0.25)* in 5-class and 7-class models.

The acceptability assessment of the cost estimation models by *PRED(0.25)* is shown in Fig. 8. All models integrating risk assessment information into COCOMO by using the fuzzy decision tree approach have more than 75% predicted values falling within 25% of their actual values. Furthermore, the stop criteria  $\theta_s$  of values 0.07 and 0.08, i.e. method ID 3, 4, 7, 8, 11, 12, 15 and 16, have higher *PRED(0.25)* values than the other methods of  $\theta_s$  with values 0.09 and 0.10, i.e. method ID 1, 2, 5, 6, 9, 10, 13 and 14. The assessment results by *MMRE* and *PRED(0.25)* are consistent, thus demonstrating that the model integrating risk assessment information into COCOMO by using the fuzzy decision tree approach can improve the accuracy of the generated software cost estimates.

## 6. CONCLUSIONS

The gap between the estimated costs and the actual costs derived from software cost drivers is inevitable, especially in the early stage of the software development life cycle. One of the main reasons for this is the fact that software cost drivers have the property of uncertain and fuzzy. This explains why the information on the estimation error is important for model builders to present the software risk caused by the estimation error in any

software cost estimation model. It is also important for project managers to effectively conduct the software project planning activities, such as allocating the required effort, cost and schedule. Unfortunately, most of the existing software cost estimation models do not explicitly reveal such important information about the uncertainty or risk of their cost estimates.

The proposed method in this paper is embedding risk assessment information into software cost estimation model by using the fuzzy decision tree approach. The fuzzy decision tree in the model captures uncertainty associated with the software cost drivers and the generated software cost estimate. The estimation result in our proposed model does not only give a single point of software cost estimate, but also explicitly reveal the risk of the generated cost estimate. With the availability of the risk information of the software cost estimate, project managers are able to better allocate and distribute the required resources and perform risk-handling activity when needed at any stage of the software development life cycle.

### ACKNOWLEDGMENTS

This research was supported by the National Science Council (NSC) of Taiwan and Chung-Shan Institute of Science and Technology under the contract NSC 90-2623-7-011-002. The authors also wish to thank the guest editor Professor Sung Shin for his editorial effort.

### REFERENCES

1. A. J. Albrecht and J. Gaffney, "Software function source lines of code and development effort prediction," *IEEE Transactions on Software Engineering*, Vol. 9, 1983, pp. 639-648.
2. J. W. Bailey and V. R. Basili, "A meta-model for source development resource expenditures," in *Proceedings of 5th International Conference on Software Engineering*, 1981, pp. 107-116.
3. J. Bode, "Neural networks for cost estimation," *Cost Engineering*, Vol. 40, 1998, pp. 25-30.
4. B. W. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.
5. B. W. Boehm, *Software Cost Estimation with COCOMO II*, Prentice Hall PTR, Englewood Cliffs, New Jersey, 2000.
6. X. Boyen and L. Wehenkel, "Automatic induction of fuzzy decision trees and its application to power system security assessment," *Fuzzy Sets and Systems*, Vol. 102, 1999, pp. 3-19.
7. L. C. Briand, T. Langley, and I. Wiczorek, "A replicated assessment and comparison of common software cost modeling techniques," in *Proceedings of 22nd International Conference on Software Engineering*, 2000, pp. 377-386.
8. L. C. Briand, K. E. Eman, and K. D. Maxwell, "An assessment and comparison of common software cost modeling techniques," in *Proceedings of 21st International*

- Conference on Software Engineering*, 1999, pp. 313-322.
9. S. D. Conte, H. E. Dunsmore, and V. Y. Shen, *Software Engineering Metrics and Models*, Benjamin Cummings, New York, 1986.
  10. A. R. Gray and S. G. MacDonell, "Applications of fuzzy logic to software metric models for development effort estimation," in *Proceeding of Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, 1997, pp. 394-399.
  11. F. J. Heemstra, "Software cost estimation," *Information and Software Technology*, Vol. 34, 1992, pp. 627-639.
  12. J. R. Herd, J. N. Postak, W. E. Russel, and K. R. Stewart, "Software cost estimation study-study results," Final Technical Report No. RADC-TR-77-220, Doty Associates, Inc., Rockville, MD, 1977.
  13. B. Jeng, Y. M. Jeng, and T. P. Liang, "FILM: a fuzzy inductive learning method for automated knowledge acquisition," *Decision Support Systems*, Vol. 21, 1997, pp. 61-73.
  14. C. F. Kemerer, "An empirical validation of software cost estimation models," *Communications of ACM*, Vol. 30, 1987, pp. 416-429.
  15. M. W. Kim, J. G. Lee, and C. Min, "Efficient fuzzy rule generation based on fuzzy decision tree for data mining," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Vol. 3, 1999, pp. 1223-1228.
  16. G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic Theory and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.
  17. A. L. Lederer and J. Prasad, "A causal model for software cost estimating error," *IEEE Transactions on Software Engineering*, Vol. 24, 1998, pp. 137-148, 1998.
  18. S. G. MacDonell and M. J. Shepperd, "Combining techniques to optimize effort predictions in software project management," *Journal of Systems and Software*, Vol. 66, 2003, pp. 91-98.
  19. J. E. Matson, B. E. Barrett, and J. M. Mellichamp, "Software development cost estimation using function points," *IEEE Transactions on Software Engineering*, Vol. 20, 1994, pp. 275-287.
  20. T. Mukhopadhyay, S. S. Vicinanza, and M. J. Prietula, "Examining the feasibility of a case-based reasoning model for software effort estimation," *MIS Quarterly*, Vol. 16, 1992, pp. 155-171.
  21. C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique," *Fuzzy Sets and Systems*, Vol. 138, 2003, pp. 221-254.
  22. W. Pedrycz, J. F. Peters, and S. Ramanna, "A fuzzy set approach to cost estimation of software projects," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, 1999, pp. 1068-1073.
  23. W. Pedrycz and Z. A. Sosnowski, "The design of decision trees in the framework of granular data and their application to software quality models," *Fuzzy Sets and Systems*, Vol. 1234, 2001, pp. 271-290.
  24. M. J. Prietula, S. S. Vicinanza, and T. Mukhopadhyay, "Software effort estimation with a case-based reasoner," *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 8, 1996, pp. 341-363.
  25. J. Ryder, "Fuzzy modeling of software effort prediction," in *Proceedings of the IEEE Information Technology Conference*, 1998, pp. 53-56.

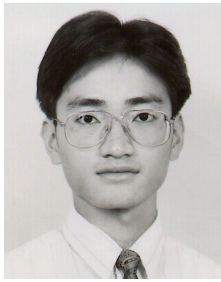
26. J. R. Quinlan, "Induction on decision tree," *Machine Learning*, Vol. 1, 1986, pp. 81-106.
27. J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
28. M. J. Shepperd and C. Schofield, "Estimating software project effort using analogies," *IEEE Transactions on Software Engineering*, Vol. 23, 1997, pp. 736-743.
29. K. Srinivasan and D. Fisher, "Machine learning approaches to estimating software development effort," *IEEE Transactions on Software Engineering*, Vol. 21, 1995, pp. 126-137.
30. C. E. Walston and C. P. Felix, "A method of programming measurement and estimation," *IBM Systems Journal*, Vol. 16, 1971, pp. 54-73.
31. Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, Vol. 69, 1995, pp. 125-139.
32. L. Zadeh, "Fuzzy sets," *Journal of Information and Control*, Vol. 8, 1965, pp. 338-353.



**Sun-Jen Huang (黃世禎)** received his B.A. in Industrial Management in 1988, and his M.S. in Engineering and Technology in 1991, both from the National Taiwan Institute of Technology, Taipei, and the PhD degree from the School of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia, in 1999. He is currently an assistant professor in the Department of Information Management, National Taiwan University of Science and Technology, Taipei. He is also the head of the Software Engineering and Management Laboratory (SEML), which hosts several research projects from National Science Council, Taiwan. Dr. Huang is also a member of Software Quality Promotion Committee at the Chinese Society for Quality. His research interests include software measurement and analysis, software project estimation, software process improvement, and software project management. Dr. Huang has published papers in journals including *IEEE Transactions on Software Engineering*, *Information & Management*, *Software Practice & Experience*, *Journal of Systems and Software*, and *Information and Software Technology*.



**Chieh-Yi Lin (林傑毅)** received his Bachelor and Master degrees both from the Department of Information Management, National Taiwan University of Science and Technology, in 2000 and 2002, respectively. He is currently a senior software analyzer in charge of developing the device driver and embedded applications for the Network Attached Storage system in Linux platform at Software Department of Wistron Corporation. His research interests include software project estimation, software verification and validation, and software quality assurance.



**Nan-Hsing Chiu (邱南星)** is currently a Ph.D. candidate in the Department of Information Management, National Taiwan University of Science and Technology, Taiwan. He received his Bachelor and Master degrees in Information Management from Yuan-Ze University, Taiwan, in 2000 and 2001 respectively. He is also a member of the Software Engineering and Management Laboratory (SEML), which participates in several projects from National Science Council, Taiwan. His main research interests include software effort estimation, data mining, optimization, grey system theory and simulation.