# Using Decision Trees for Determining Attribute Weights in a Case-Based Model of Early Cost Prediction

Sevgi Zeynep Doğan[1]; David Arditi, M.ASCE[2]; and H. Murat Günaydin[3]

**Abstract:** This paper compares the performance of three different decision-tree-based methods of assigning attribute weights to be used in a case-based reasoning (CBR) prediction model. The generation of the attribute weights is performed by considering the presence, absence, and the positions of the attributes in the decision tree. This process and the development of the CBR simulation model are described in the paper. The model was tested by using data pertaining to the early design parameters and unit cost of the structural system of residential building projects. The CBR results indicate that the attribute weights generated by taking into account the information gain of all the attributes performed better than the attribute weights generated by considering only the appearance of attributes in the tree. The study is of benefit primarily to researchers, as it compares the impact of attribute weights generated by three different methods and, hence, highlights the fact that the prediction rate of models such as CBR largely depends on the data associated with the parameters used in the model.

**DOI:** 10.1061/(ASCE)0733-9364(2008)134:2(146)

**CE Database subject headings:** Construction cost; Cost estimates; Decision making; Decision support systems; Computer software; Optimization models; Predictions; Weight.

## Introduction

The construction activity is experience oriented. Knowledge and appreciation of previous experience are critical to resolving problems that may reoccur. Case-based reasoning (CBR) is an effective technique in the machine learning domain, capable of solving or providing suggestions for a problem by storing and retrieving outcomes of previous experiences. CBR's benefits in solving construction management-related problems over other prediction techniques have been demonstrated by Arditi and Tokdemir (1999a,b) and Yau and Yang (1998). Recent research studies about the effectiveness of integrated machine learning approaches indicate that CBR systems could achieve better results when enhanced by other techniques (Cardie 1993; Jarmulak and Craw 1999; Jarmulak et al. 2000; Ling et al. 1997; Shin and Han 2002).

CBR directly interprets past experiences. In other words, CBR systems predict the outcome of new situations by retrieving previously stored outcomes of similar situations from a case base. Fig. 1 shows how the CBR system operates to select a retrieved case in order to make a prediction. Constructing CBR systems requires a significant knowledge engineering effort, because the system mainly relies on a good case base organization, an effective selection of case attributes, and the effectiveness of similarity assessment (Cunningham and Bonzano 1999). The selection of relevant case attributes can be done by a domain expert quite easily, assuming that data are available. Weighing the relative importance of these attributes is more difficult. In fact, determining the weights of the attributes is an important CBR problem. An effective similarity assessment should indeed take into consideration the relative importance of the attributes considered in the model. For example, Dogan et al. (2006) tested the performance of a case-based reasoning model by testing the impact of attribute weights generated by three different techniques, namely, feature counting, gradient descent, and genetic algorithms. The objective of the study reported in this paper is to use a totally different approach, namely, an induction algorithm named ID3 (Quinlan 1986) to determine attribute weights. ID3 is radically different from the three techniques mentioned earlier and offers a number of variations that are worth testing. This paper attempts to assess the performance of a spreadsheet-based CBR prediction model by testing the impact of attribute weights generated by three of these ID3 variations. The weight generation processes are described in detail and the subsequent CBR calculations are presented in a simple spreadsheet format that is transparent and easy to use. The model is tested by predicting the cost of the structure (excluding interior walls and finishes) of residential building projects at an early design stage. The results are compared and recommendations made.

## Methodology of the Study

Data were obtained from a research report that investigated the cost of the structural system in 29 building construction projects undertaken in Istanbul, Turkey (Saner 1993). All 29 buildings were located in the same earthquake zone and were subject to the

[1]Research Assistant, Dept. of Architecture, Izmir Institute of Technology, Izmir 35430, Turkey. E-mail: sevgidogan@iyte.edu.tr

[2]Professor, Dept. of Civil and Architectural Engineering, Illinois Institute of Technology, Chicago, IL 60616-3793. E-mail: arditi@iit.edu

[3]Associate Professor, Dept. of Architecture, Izmir Institute of Technology, Izmir, 35430, Turkey.

Note. Discussion open until July 1, 2008. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on June 14, 2005; approved on July 13, 2007. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 134, No. 2, February 1, 2008. ©ASCE, ISSN 0733-9364/2008/2-146–152/$25.00.

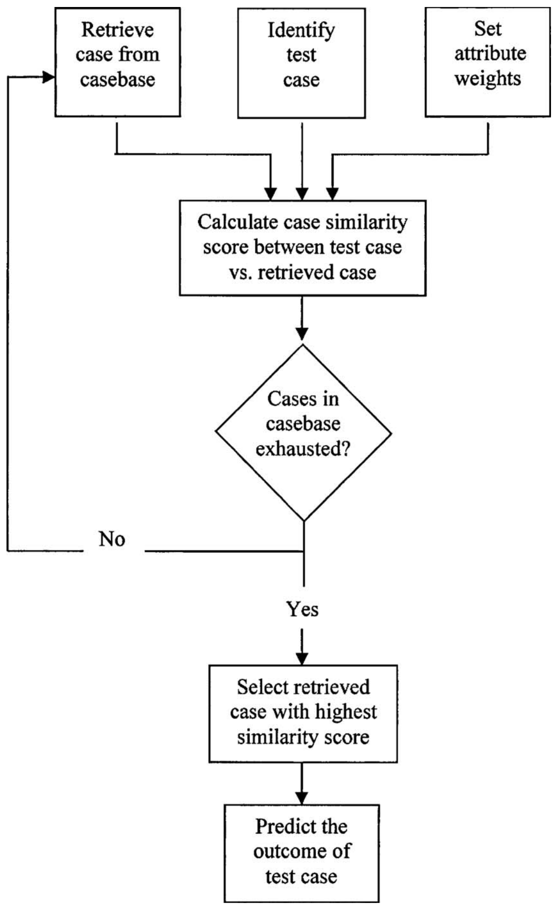

Retrieve case from casebase
Identify test case
Set attribute weights
Calculate case similarity score between test case vs. retrieved case
Cases in casebase exhausted?
No
Yes
Select retrieved case with highest similarity score
Predict the outcome of test case

**Fig. 1.** Basic process of CBR

same design codes and earthquake regulations. The unit cost of the structural system ($\$/\text{m}^2$) was used as the *output* of the model. Out of the variables that Saner (1993) used in his study, only those that can be identified in the early design stages and that have an impact on cost were selected as the predominant attributes. These attributes define the characteristics of the structure of a building (i.e., excluding interior walls and finishes) and include: (1) the total area of the building, which bears a strong linear relationship to the total cost of the building; (2) the ratio of the typical floor area to the total area of the building, which influences directly the cost of the components such as beams and columns of the load-bearing frame; (3) the ratio of the footprint area to the total area of the building, which is correlated with the width and depth of the foundation system; (4) the number of floors, which has a direct effect on the structural design and consequently, cost of columns; (5) the type of overhang, which may range between no overhang in the design to one-way design, increasing the cost of the structural system; (6) the location of the core of the building (i.e., vertical circulation system including stairs, elevators, and the service ducts), a central location requiring less cost than a side location, which necessitates extra shear walls to counteract torsion effects; (7) the type of floor, which includes cast in situ concrete floor systems or precast concrete structural units; and (8) the foundation system classified as pier, wall, or slab foundations, each necessitating different amounts of concrete and reinforcement. The information relative to the textual attributes and the range of values relative to numerical attributes are presented in the last column of Table 1.

With the eight input attributes and the output attribute defined, relevant data were then entered into a CBR-Excel model using the procedure described in Fig. 1. The dataset of 29 projects was randomly split into an *input* set containing 24 projects, and a *test* set containing five projects. It is customary to pick 10% of the total number of cases as test cases. However, given the small number of cases (only 29), 10% meant that only three cases were chosen for this purpose. But using only three test cases ended up destabilizing the model, and resulting in predictions that were either extremely accurate or extremely inaccurate. When seven or eight test cases were chosen, then the CBR process became very inefficient because there were not enough cases in the case base from which to retrieve similar cases to the test cases. Given these constraints, it was found that the best solutions were obtained using five cases as test cases, and the remaining 24 as input cases. The same five test projects were used to evaluate the effect of the attribute weights generated by different methods.

Attribute weights play an important role in CBR prediction because the overall error obtained in CBR is a function of attribute weights. The output values were divided into 13 classes, as presented in Table 2. The See5/C5.0. (1997) software package was used to construct a decision tree. The outcome of the decision analysis was used to assign weights to the attributes. The attribute weights were generated by using (1) the binary-dtree method; (2) the info-top method; and (3) the info-dtree method, all described in detail in the following section.

**Table 1.** Attributes Used in CBR Prediction Model

| Input attribute number | Attribute | Range |
|---|---|---|
| 1 | Total area of the building | $330\text{ m}^2$–$3{,}484\text{ m}^2$ |
| 2 | Ratio of the typical floor area to the total area of the building | 0.07–0.26 |
| 3 | Ratio of the footprint area to the total area of the building | 0.07–0.30 |
| 4 | Number of floors | 4–8 |
| 5 | Type of overhang design | No overhang or one way |
| 6 | Foundation system | Pier, wall, slab |
| 7 | Type of floor structure | Cast in situ concrete, precast concrete |
| 8 | Location of the core | At the sides, in the middle |
| Output | Cost of the structural system per $\text{m}^2$ | $\$30/\text{m}^2$–$\$160/\text{m}^2$ |

**Table 2.** Classes Specified for Output Attribute of Cost per $m^2$

| Class number | Cost |
|---|---|
| 1 | $\$30/m^2 < \text{cost} \leq \$40/m^2$ |
| 2 | $\$40/m^2 < \text{cost} \leq \$50/m^2$ |
| 3 | $\$50/m^2 < \text{cost} \leq \$60/m^2$ |
| 4 | $\$60/m^2 < \text{cost} \leq \$70/m^2$ |
| 5 | $\$70/m^2 < \text{cost} \leq \$80/m^2$ |
| 6 | $\$80/m^2 < \text{cost} \leq \$90/m^2$ |
| 7 | $\$90/m^2 < \text{cost} \leq 100/m^2$ |
| 8 | $\$100/m^2 < \text{cost} \leq \$110/m^2$ |
| 9 | $\$110/m^2 < \text{cost} \leq \$120/m^2$ |
| 10 | $\$120/m^2 < \text{cost} \leq \$130/m^2$ |
| 11 | $\$130/m^2 < \text{cost} \leq \$140/m^2$ |
| 12 | $\$140/m^2 < \text{cost} \leq \$150/m^2$ |
| 13 | $\$150/m^2 < \text{cost} \leq \$160/m^2$ |

## CBR Spreadsheet Model

A spreadsheet model of a CBR system was set up in Microsoft Excel. The details of this model can be seen in Dogan et al. (2006). The processing of the information involves six steps:

**Step 1**. *Organizing and formatting data*—The data are organized in the form of two matrices, one for the five test cases and one for the 24 input cases (see Fig. 2). The input and test cases are represented in rows and the eight input attributes in columns. The output attribute is placed in a column next to the input attributes. The values of the eight attributes for each of the five test cases and the remaining 24 input cases are presented in Fig. 2. The weights of the attributes $w_k (k=1,2,\ldots,8)$ are located at the top of the matrix in a row that corresponds to individual attributes. The way these weights are set is explained in Step 3. After formatting, semantic information is added to the data in the form of numerical and textual attribute values.

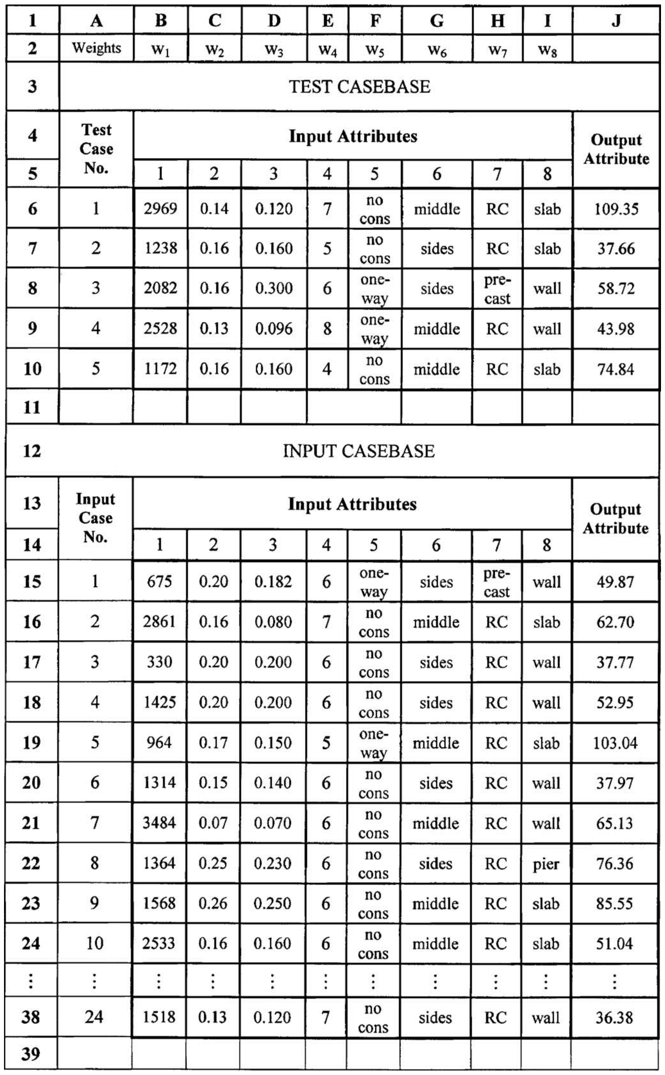
**Step 2**. *Calculating attribute similarities*—Attribute similarity functions are used to define how similar the attribute values are to each other. Attribute similarities are computed with respect to each test case versus every case retrieved from the input casebase. Examples of textual and numerical similarities of Test Case 1 with the 24 input cases are presented in Fig. 3. In this figure, see particularly the formulas at the top of the table that show how the values in the cells are calculated.

For example, since the value of the fifth attribute in cell F6 in Fig. 2 is textual, its similarity with the corresponding attribute value in cell F15 is established by using the following if/then rule:

*If* text in cell F6 appears to be exactly the same as text in cell F15, *then* similarity is 1, *or else* similarity is 0 [see Excel formula "=IF(F6=F15,'1','0")' in Fig. 3].

On the other hand, since the value of the first attribute in cell B6 in Fig. 2 is numerical, its similarity with the attribute value in the corresponding cell B15 is established by dividing the smaller of the cell values (B6 or B15) by the larger value [see Excel formula "=MIN(B6,B15)/MAX(B6,B15)" in Fig. 3].

**Step 3**. *Establishing attribute weights*—Setting the weights in the CBR model can affect the prediction rate. More important attributes should have larger weight, while totally irrelevant attributes should have no weight. After all the attribute similarity values are calculated in $(24\times8)$ matrices, once for each of the five test cases (the matrix for Test Case 1 is presented in Fig. 3), the next step is to construct the weight vector that will be used in computing case similarities. Weights assign a value of importance



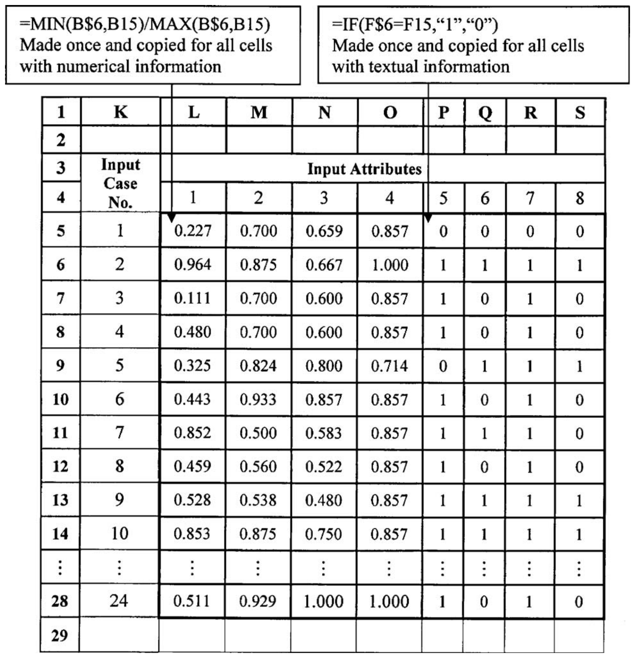
| 1 | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Weights | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $w_7$ | $w_8$ | |
| 3 | TEST CASEBASE | | | | | | | | | |
| 4 | **Test Case No.** | **Input Attributes** | | | | | | | | **Output Attribute** |
| 5 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 6 | 1 | 2969 | 0.14 | 0.120 | 7 | no cons | middle | RC | slab | 109.35 |
| 7 | 2 | 1238 | 0.16 | 0.160 | 5 | no cons | sides | RC | slab | 37.66 |
| 8 | 3 | 2082 | 0.16 | 0.300 | 6 | one-way | sides | pre-cast | wall | 58.72 |
| 9 | 4 | 2528 | 0.13 | 0.096 | 8 | one-way | middle | RC | wall | 43.98 |
| 10 | 5 | 1172 | 0.16 | 0.160 | 4 | no cons | middle | RC | slab | 74.84 |
| 11 | | | | | | | | | | |
| 12 | INPUT CASEBASE | | | | | | | | | |
| 13 | **Input Case No.** | **Input Attributes** | | | | | | | | **Output Attribute** |
| 14 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 15 | 1 | 675 | 0.20 | 0.182 | 6 | one-way | sides | pre-cast | wall | 49.87 |
| 16 | 2 | 2861 | 0.16 | 0.080 | 7 | no cons | middle | RC | slab | 62.70 |
| 17 | 3 | 330 | 0.20 | 0.200 | 6 | no cons | sides | RC | wall | 37.77 |
| 18 | 4 | 1425 | 0.20 | 0.200 | 6 | no cons | sides | RC | wall | 52.95 |
| 19 | 5 | 964 | 0.17 | 0.150 | 5 | one-way | middle | RC | slab | 103.04 |
| 20 | 6 | 1314 | 0.15 | 0.140 | 6 | no cons | sides | RC | wall | 37.97 |
| 21 | 7 | 3484 | 0.07 | 0.070 | 6 | no cons | middle | RC | wall | 65.13 |
| 22 | 8 | 1364 | 0.25 | 0.230 | 6 | no cons | sides | RC | pier | 76.36 |
| 23 | 9 | 1568 | 0.26 | 0.250 | 6 | no cons | middle | RC | slab | 85.55 |
| 24 | 10 | 2533 | 0.16 | 0.160 | 6 | no cons | middle | RC | slab | 51.04 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 38 | 24 | 1518 | 0.13 | 0.120 | 7 | no cons | sides | RC | wall | 36.38 |
| 39 | | | | | | | | | | |

**Fig. 2.** Formatting data to a case spreadsheet

to each attribute. In general, retrieval of the most relevant case is determined by the presence of a greater number of higher priority (more important) attributes matching between the test case and the retrieved case. In this study, weights for attributes were determined by using decision tree learning algorithms.

The decision tree constructed by See5 is presented in Fig. 4. Each *branch* node (oval shape) in the decision tree represents an attribute, and the branches correspond to the possible values of the attribute. Each *leaf* node (rectangular shape) represents a decision. See5 makes use of the basic tree induction algorithm ID3 (Quinlan 1986). Related information about decision trees can be found in Cardie (1993).

See5 builds a decision tree that consists of a sequence of logical decisions based on the attributes. It builds decision trees by employing a simple divide and conquer strategy. It first chooses an attribute as the current root, divides the input cases into subsets, and recursively tests the subsets, until all remaining cases belong to a single class. The choice of the attribute is based on the information gain. See5 always chooses the attribute with maximum information gain as the current root; such attributes tend to be most discriminative or informative for classification at that point. The computations usually result in a small decision tree. It is to be stressed that See5 was not used to predict the outcome of


=MIN(B$6,B15)/MAX(B$6,B15)
Made once and copied for all cells with numerical information

=IF(F$6=F15,"1","0")
Made once and copied for all cells with textual information

| 1 | K | L | M | N | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|
| 2 | | | | | | | | | |
| 3 | Input Case No. | Input Attributes | | | | | | | |
| 4 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 5 | 1 | 0.227 | 0.700 | 0.659 | 0.857 | 0 | 0 | 0 | 0 |
| 6 | 2 | 0.964 | 0.875 | 0.667 | 1.000 | 1 | 1 | 1 | 1 |
| 7 | 3 | 0.111 | 0.700 | 0.600 | 0.857 | 1 | 0 | 1 | 0 |
| 8 | 4 | 0.480 | 0.700 | 0.600 | 0.857 | 1 | 0 | 1 | 0 |
| 9 | 5 | 0.325 | 0.824 | 0.800 | 0.714 | 0 | 1 | 1 | 1 |
| 10 | 6 | 0.443 | 0.933 | 0.857 | 0.857 | 1 | 0 | 1 | 0 |
| 11 | 7 | 0.852 | 0.500 | 0.583 | 0.857 | 1 | 1 | 1 | 0 |
| 12 | 8 | 0.459 | 0.560 | 0.522 | 0.857 | 1 | 0 | 1 | 0 |
| 13 | 9 | 0.528 | 0.538 | 0.480 | 0.857 | 1 | 1 | 1 | 1 |
| 14 | 10 | 0.853 | 0.875 | 0.750 | 0.857 | 1 | 1 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 28 | 24 | 0.511 | 0.929 | 1.000 | 1.000 | 1 | 0 | 1 | 0 |
| 29 | | | | | | | | | |

**Fig. 3.** Attribute similarity matrix for Test Case 1 (five similar matrices are generated, one for each of the five test cases)

the test cases, but only to generate attribute weights to be used in the CBR model. The CBR model makes these predictions with the help of the attribute weights produced by See5.

The three different methods used in this study were named (1) binary-dtree method; (2) info-top method; and (3) info-dtree method by Ling et al. (1997).

## Binary-Dtree Method

Kibler and Aha (1987) first presented a simple approach that uses the presence or absence of attributes in the decision tree to determine their weights. If an attribute is present in the decision tree, then its weight is 1, otherwise, its weight is 0. The method is very efficient since it only involves running See5 over the input cases. Cardie (1993) used this method to improve case-based learning and pointed out that a strategy of considering the positions of attributes in the decision tree (such as in the info-top and info-dtree methods) may work better.

## Info-Top Method

Rather than considering only attributes with the maximum information gain (i.e., those appearing in the tree), this method considers the information gain of all attributes at the top level; that is, the information gain of all attributes based on all the input cases. Thus, there is no need to construct the decision tree. These information-gain values are used as the weights in the similarity assessment process. Clearly, the attribute with maximum information gain is assigned a maximum weight, but other attributes can have some smaller effects in the similarity assessment as well, rather than being completely ignored.

## Info-Dtree Method

This method takes into account the location of the attributes in the decision tree. Thus, a decision tree is first constructed using the input cases. For each attribute, which may appear in several places in the tree, the weight is determined by taking the sum of its information gain at each appearance multiplied by the percent of input cases classified by that attribute. For example, if an attribute appears three times in the tree, with information gain values of 0.9, 0.8, and 1.0 with 40, 20, and 10% of the input cases classified by the attribute, respectively, then the weight of this attribute is $(0.9\times0.4)+(0.8\times0.2)+(1.0\times0.1)=0.62$. Clearly, attributes at lower levels contribute less to their weight because the number of input cases they classify is smaller. This method, like binary dtree, considers only the information gain of those attributes that appear in the tree.

The impact of the three sets of attribute weights was evaluated using the same test set of five projects. The weights of the attributes generated by the three methods are presented in Table 3. The attribute weights obtained were used in Step 4 of the CBR process.

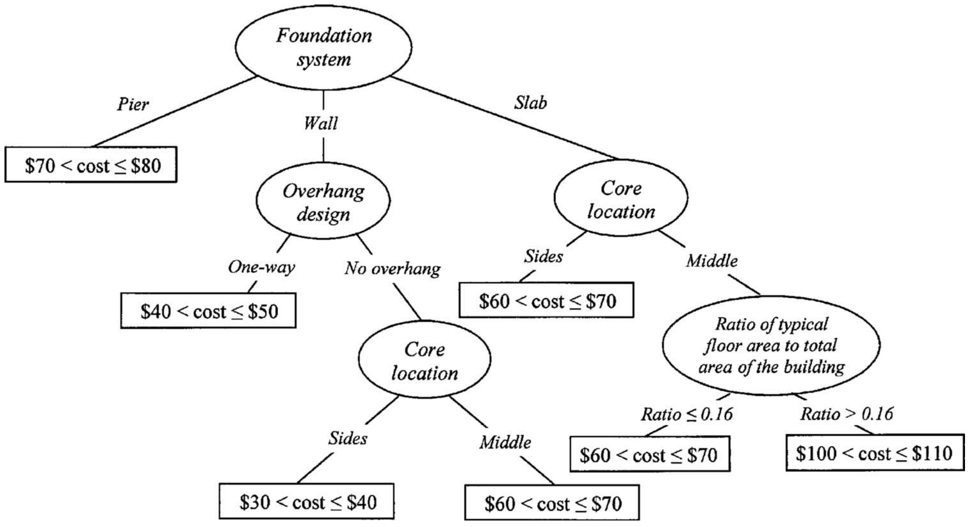


**Fig. 4.** Decision tree constructed by See5 according to the output attribute classes in Table 2

**Table 3.** Optimized Attribute Weights and Average Error Percentages for Three Methods

| Decision tree method of weight generation | Attribute weights | | | | | | | | Average error in CBR prediction (%) |
|---|---|---|---|---|---|---|---|---|---|
| | Total area | Ratio of floor area to total area | Ratio of footprint area to total area | Number of floors | Overhang design | Core location | Floor type | Foundation system | |
| Binary dtree | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 20.70 |
| Info top | 0,387,129 | 0,451,902 | 0,439,009 | 0,355,676 | 0,509,398 | 0,511,249 | 0,189,805 | 0,783,560 | 17.63 |
| Info dtree | 0 | 0,204,025 | 0 | 0 | 0,243,221 | 0,604,721 | 0 | 0,783,560 | 20.70 |

**Step 4**. *Calculating weighted case similarities*—Case similarities are computed for each test case with respect to each input case by using the attribute similarities calculated in Step 2 (see Fig. 3 for formulas and calculations) and the attribute weights generated in Step 3. For positive weights and normalized similarities, the weighted case similarities are always between 0 and 1, with a score of 1 indicating the case most similar to the test case and 0 the least. Weighted case similarities are computed and presented in Fig. 5 for the info-top method. In this figure, see particularly the formulas at the top of the table that show how the values in the cells are calculated. For example, the weighted case similarity values of the test cases relative to input case 1 (i.e., cells U5 to U9) are calculated by multiplying the corresponding attribute similarity values (i.e., cells L5 to S5 in Fig. 3) by their corresponding weights (to be found in Cells B2 to I2 in Fig. 2), taking their sum, and then dividing their sum by the sum of the weights (see the Excel formula at the top left corner of Fig. 5).

**Step 5**: *Sorting weighted case similarities and corresponding outputs*—The highest weighted case similarity for a test case indicates the closest matching input case in the case base. For example, the highest weighted case similarity value for test case 1 is obtained by using the formula "=MAX(U5:AF5)" at the top right corner in Fig. 5.

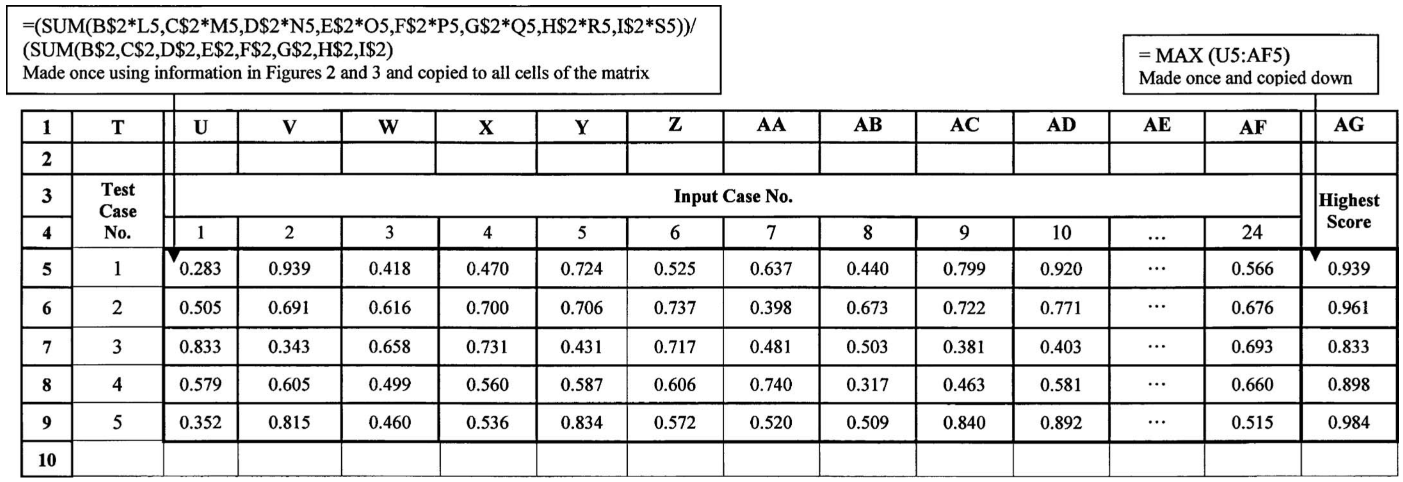
Once the highest weighted case similarities are identified for respective test cases (see Column AG in Fig. 5 and Column AI in Fig. 6), the corresponding case numbers and output values are also listed (see Columns AJ and AK in Fig. 6).

**Step 6**: *Calculating the error*—The output values listed in the preceding step (Column AK in Fig. 6) are compared with the respective actual output values (Column AL in Fig. 6, same as Column J in Fig. 2). The differences constitute the errors and are listed in Column AM in Fig. 6. The errors are calculated in absolute values by using the Excel formula "ABS((100-(AK4∗100)/AL4))/100" for Test Case 1 in Fig. 6. The formula is adjusted and repeated for the remaining test cases. The average of the error values of all test cases is the overall error of the CBR process.

## Results and Discussion

After the attribute weights were determined by using the binary-dtree, info-top, and info-dtree methods, the CBR-Excel model was run and the performance of the model was evaluated vis à vis each method. The tests were repeated ten times with ten different test sets that were selected at random, and the best results obtained in these tests were selected and are reported in this paper. The results presented in the last column of Table 3 indicate that info top+CBR yielded an average error of 17.63%, whereas binary dtree+CBR and info dtree+CBR had average errors of 20.70%.

The setting up of the attribute weights in the **binary–dtree method** was straight forward in that the attributes appearing in the decision tree (the foundation system, the type of overhang design, the location of the core, and the ratio of the typical floor area to the total area of the building) (see Fig. 4) were weighted as 1, whereas the attributes that did not appear in the tree (the total area of the building, the ratio of the footprint area to the total area of the building, the number of floors, the type of floor structure) were weighted as 0, as seen in Table 3. In the **info–top method**, all of the eight attributes were given weights according to their information gain values. The attribute with the

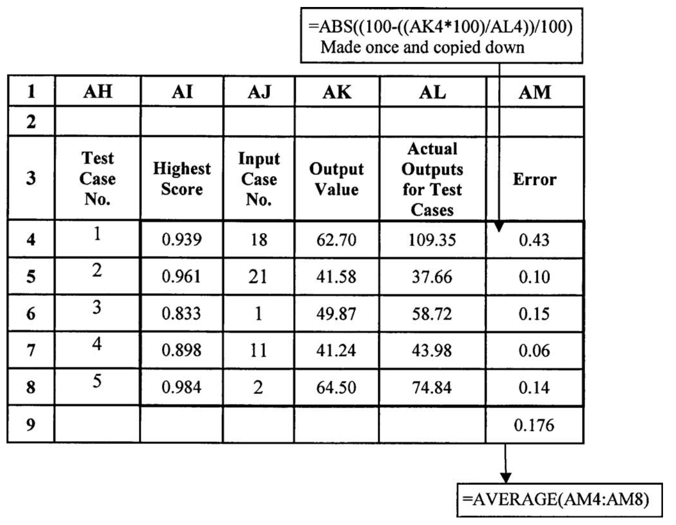


=(SUM(B$2*L5,C$2*M5,D$2*N5,E$2*O5,F$2*P5,G$2*Q5,H$2*R5,I$2*S5))/(SUM(B$2,C$2,D$2,E$2,F$2,G$2,H$2,I$2)
Made once using information in Figures 2 and 3 and copied to all cells of the matrix

= MAX (U5:AF5)
Made once and copied down

| 1 | T | U | V | W | X | Y | Z | AA | AB | AC | AD | AE | AF | AG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | | | | | | | | | | | | | | |
| 3 | Test Case No. | Input Case No. | | | | | | | | | | | | Highest Score |
| 4 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | … | 24 | |
| 5 | 1 | 0.283 | 0.939 | 0.418 | 0.470 | 0.724 | 0.525 | 0.637 | 0.440 | 0.799 | 0.920 | ··· | 0.566 | 0.939 |
| 6 | 2 | 0.505 | 0.691 | 0.616 | 0.700 | 0.706 | 0.737 | 0.398 | 0.673 | 0.722 | 0.771 | ··· | 0.676 | 0.961 |
| 7 | 3 | 0.833 | 0.343 | 0.658 | 0.731 | 0.431 | 0.717 | 0.481 | 0.503 | 0.381 | 0.403 | ··· | 0.693 | 0.833 |
| 8 | 4 | 0.579 | 0.605 | 0.499 | 0.560 | 0.587 | 0.606 | 0.740 | 0.317 | 0.463 | 0.581 | ··· | 0.660 | 0.898 |
| 9 | 5 | 0.352 | 0.815 | 0.460 | 0.536 | 0.834 | 0.572 | 0.520 | 0.509 | 0.840 | 0.892 | ··· | 0.515 | 0.984 |
| 10 | | | | | | | | | | | | | | |

**Fig. 5.** Case similarity matrix for all test cases



=ABS((100-((AK4*100)/AL4))/100)
Made once and copied down

| 1 | AH | AI | AJ | AK | AL | AM |
|---|---|---|---|---|---|---|
| 2 | | | | | | |
| 3 | Test Case No. | Highest Score | Input Case No. | Output Value | Actual Outputs for Test Cases | Error |
| 4 | 1 | 0.939 | 18 | 62.70 | 109.35 | 0.43 |
| 5 | 2 | 0.961 | 21 | 41.58 | 37.66 | 0.10 |
| 6 | 3 | 0.833 | 1 | 49.87 | 58.72 | 0.15 |
| 7 | 4 | 0.898 | 11 | 41.24 | 43.98 | 0.06 |
| 8 | 5 | 0.984 | 2 | 64.50 | 74.84 | 0.14 |
| 9 | | | | | | 0.176 |

=AVERAGE(AM4:AM8)

**Fig. 6.** CBR outputs and error

highest information gain value is selected as the root of the decision tree by See5. In this study, the foundation system with the information gain value of 0,783560 was selected as the root (see Fig. 4). The information gain values of all the attributes (i.e., their weights) are presented in Table 3. In the **info−dtree method**, the attributes that appear in the tree constructed by See5 (Fig. 4) were given weights in consideration of their information gain values and their positions in the tree. For example, the attribute "console direction" appeared twice in the tree, with information gain values of 0.750 and 0.918 with 50 and 25% of input cases classified by the attribute, respectively; the weight of this attribute is calculated as $(0.750\times0.5)+(0.918\times0.25)=0.6045$. The weights of the attributes in the decision tree were calculated using the same principle and are presented in Table 3.

As discussed by Ling et al. (1997), if the number of input cases is small, See5 constructs an overly simple decision tree, overlooking relevant attributes. In the case study presented in this paper, there were 3 continuous and 5 discrete attributes, but only 29 cases. Because 5 cases had to be used as test cases, only 24 cases were left as input cases. As a result, See5 constructed a tree that included only four attributes. When this happens, the performance of the binary-dtree and info-dtree methods (that consider only the attributes in the decision tree) is bound to be worse than the info-top method (that considers the information gain of all attributes). It was, therefore, not surprising to find out that binary dtree+CBR and info dtree+CBR did not generate predictions that are as strong as the prediction generated by the info-top +CBR alternative, because they only use the attributes that appear in the decision tree and, therefore, do not take into account the information gain of the other relevant attributes, even though it is likely that such information gain affects the classification of some cases used in the study. Our findings support the conclusion of Ling et al. (1997) that info dtree and binary dtree are immune to irrelevant attributes, and that info top is suitable for situations where there are not enough input cases and where all attributes may be relevant.

On the other hand, it was surprising to see that binary dtree+CBR performed as well as info dtree+CBR. After all, info dtree is considered to be a more sophisticated method than binary dtree that assigns a weight of 1 to all attributes in the decision tree, regardless of their position in the tree (Ling et al. 1997). While binary dtree was found to be as effective as info dtree in the case study presented in this paper, it must be noted that a limited number of input cases were available. The performance of the info-dtree method could possibly improve with larger numbers of input cases.

One of the reasons why the average errors obtained (last column in Table 3) were not very low, had to do with the nature of the output attribute. The output of the cases considered in this study was the unit cost of construction of the structural system, and its value ranged between \$30 and \$160/m$^2$ (see Tables 1 and 2). In order to achieve high prediction accuracy, one should have at least two or more cases with not only quite similar input attributes, but also almost identical outputs, which is most improbable given the small number of cases (total 29) that were available for this study and the wide range of unit costs associated with the cases considered. The average errors reported in this study could have been lower had the output variables been binary or had there been a larger number of cases with an output attribute whose value varied in a smaller range.

## Concluding Remarks

CBR was used in this study to develop a prediction model where attribute weights were generated by means of three methods, namely, binary dtree, info top, and info dtree. The structure of the CBR model was modeled using an Excel spreadsheet to provide a transparent and simplified representation of this technique. The attribute weights that were generated by the three methods were then plugged into the CBR-Excel model. The model was tested by using cost data pertaining to the early design parameters and unit cost of the structural system of 29 residential building projects. The results indicated that info top+CBR performed better than CBR used in association with the other two methods.

The study demonstrated the practicality of using a spreadsheet in developing a CBR model for use in construction management. A spreadsheet simulation of an artificial neural network model developed by Hegazy and Ayed (1998) was the motivation of the study. A commercial CBR software using a spreadsheet-based user interface helped to facilitate the construction of the model (Induce-It 2000).

Info top+CBR performed well, considering the fact that the number of cases in the case base was small and the output attribute was not binary. All of the weight generation methods (binary dtree, info top, and info dtree) and the CBR prediction suffered from the fact that not many of the 29 cases considered in the study had outputs that were close to each other. More consistent outputs could have resulted in splitting the cases into fewer classes in Table 2, and consequently producing smaller prediction errors in Table 3. The likelihood of seeing stronger similarities and making more accurate classifications is much higher if the number of cases is larger than 29.

In this study, the experimentation involved testing ten sets with different test cases selected randomly. Future research may involve the development of a computer program that makes use of all different combinations of test cases and input cases (a total of 118,775 combinations). Even if it requires a long processing time, it is likely that this approach may reduce the error obtained in this study (17.6%) and may more fully justify the implementation of this method in the industry.

Despite the limitations cited above, the study is of benefit to researchers, as it illustrates the importance of attribute weights in the performance of a CBR prediction tool and demonstrates that this can be achieved by using simple spreadsheet operations.

It also indicates that it is worth experimenting with different inductive learning techniques for weight generation, rather than being confined by the standard methodologies provided by CBR software.

## References

Arditi, D., and Tokdemir, O. B. (1999a). "Comparison of case-based reasoning and artificial neural networks." *J. Comput. Civ. Eng.*, 13(3), 162–169.

Arditi, D., and Tokdemir, O. B. (1999b). "Using case-based reasoning to predict the outcome of construction litigation." *Comput. Aided Civ. Infrastruct. Eng.*, 14, 385–393.

Cardie, C. (1993) "Using decision trees to improve case-based learning." *Proc., 1993 Int. Conf. on Machine Learning*, 25–32.

Cunningham, P., and Bonzano, A. (1999). "Knowledge engineering issues in developing a case-based reasoning application." *Knowledge-Based Systems*, 12, 371–379.

Dogan, S. Z., Arditi, D., and Günaydin, H. M. (2006). "Determining attribute weights in a CBR model for early cost prediction of structural systems." *J. Constr. Eng. Manage.*, 132(10), 1092–1098.

Hegazy, T., and Ayed, A. (1998). "Neural network model for parametric cost estimation of highway projects." *J. Constr. Eng. Manage.*, 124(3), 210–218.

*Induce-it user's manual*. (2000). Inductive Solutions, Inc., New York.

Jarmulak, J., and Craw, S. (1999). "Genetic algorithms for feature selection and weighting." *Proc., IJCAI-99 Workshop on Automating the Construction of Case-Based Reasoners*, S. S. Anand, A. Aamodt, and D. W. Aha, eds., 28–33.

Jarmulak, J., Craw, S., and Rowe, R. (2000). "Genetic algorithms to optimize CBR retrieval." *Proc., EWCBR 2000, LNAI 1898*, E. Blanzeri and L. Portinale, eds., Springer-Verlag, Berlin, Heidelberg, Germany, 136–147.

Kibler, D., and Aha, D. (1987). "Learning representative exemplars of concepts: An initial case study." *Proc., Int. Workshop on Machine Learning*, Morgan Kaufmann, Irvine, Calif., 24–30.

Ling, C. X., Parry, J. J., and Wang, H. (1997). "Setting attribute weights for nearest neighbor learning algorithms using C4.5." *Int. J. Pattern Recognit. Artif. Intell.*, 11(3), 405–415.

Quinlan, J. R. (1986). "Induction of decision trees." *Mach. Learn.*, 1, 81–106.

Saner, C. (1993). "A proposal for cost estimation for structural systems of 4–8 storey residential buildings." MS thesis, Istanbul Technical Univ., Istanbul, Turkey.

See5/C5.0. (1997). *Data mining tools*, Rulequest, Australia.

Shin, K., and Han, I. (2002). "A case-based approach using inductive learning for corporate bond rating." *Decision Support Sys.*, 32(4), 41–52.

Yau, N. J., and Yang, J. B. (1998). "Case-based reasoning in construction management." *Comput. Aided Civ. Infrastruct. Eng.*, 13, 143–150.