A hierarchical design of case-based reasoning in the balanced scorecard application

Fong-Ching Yuan *, Chaochang Chiu

Department of Information Management, Yuan Ze University, 135 Yuan-Tung Road, Chungli 320, Taiwan, ROC

Abstract

A balanced scorecard (BSC) is a management decision tool intended to be the corporate performance measurement. It also can play an important role in transforming an organization's mission and strategy into a balanced set of integrated performance measures. Assigning suitable weight to each level of balanced scorecard is crucial to conduct performance evaluation effectively.

In this research a case-based reasoning (CBR) system has been developed to assist in assigning the suitable weights. Based on the balanced scorecard design, this study proposed a three-level feature weights design to enhance CBR's inference performance. For effective case retrieval, a genetic algorithm (GA) mechanism is employed to facilitate weighting all of levels in balanced scorecard and to determine the most appropriate three-level feature weights. The proposed approach is compared with the equal weights approach and the analytical hierarchy process (AHP) approach. The results indicate that the GA-CBR approach is able to produce more effective performance measurement.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Balanced scorecard (BSC); Case-based reasoning (CBR); Genetic algorithms (GA); Analytical hierarchy process (AHP); Performance measurement

1. Introduction

The BSC (Kaplan & Norton, 1992) is a performance measurement framework to allow managers to look at their business performance from four performance perspectives – financial, customer, internal business and innovation and learning. The weight of each feature in balanced scorecard is an impact factor to evaluate performance. The AHP method (Saaty, 1990) is used to generate the weights (Stewart & Mohamed, 2001).

The AHP method is often used as an effective tool in structuring and modeling multi-criteria problems because it attempts to quantify human judgment and opinion that other approaches might ignore. However, by using pairwise comparison the calculation of preference between criteria is mainly based on some quantitative business data

and the subjective judgment from senior management level. No matter how professional they are, the results based on the judgment of those decision-makers somewhat are subject to imprecision.

A proposed CBR approach is suggested to handle subjective judgment problems. CBR is a machine reasoning that adapts previous similar cases to infer further similarity. It allows a computer program to propose solutions in domains that are not completely understood (Kolodner, 1992). To develop a CBR system, a set of useful case features must first be determined to differentiate one case from the others. Furthermore, weights representing the importance of features have to be assigned in the case-matching process.

In order to apply CBR to the balanced scorecard, this study adopts the CBR system with a three-level weight design. The weights are usually determined by subjective judgments or the trial and error approach. Instead of subjective judgments or the inefficient way of trial and error, the GA is adopted to determine the weights (Chiu, 2002).

* Corresponding author. Tel.: +886 03 4638800x2606; fax: +886 03 4352077.
E-mail address: imyuan@saturn.yzu.edu.tw (F.-C. Yuan).

To evaluate business performance, this study presents a CBR system with the GA which is used to determine the optimal three-level weights. Without any human judgments with questionnaires, such as AHP method, the weight can be produced automatically by computer. Then, the weight produced can be used in balanced scorecard to evaluate performance.

2. Case-based reasoning

CBR is a problem-solving method that is similar to the analogical decision making process used in real world applications. Several recent works have applied GA-based CBR approach to different domains, including corporate bond rating (Shin & Han, 1999), housing customization (Juan, Shih, & Perng, 2006), Q&A system (Fu & Shen, 2004), e-learning system (Huang, Huang, & Chen, 2007), and etc.

The CBR execution cycle is illustrated in Fig. 1 (Bradley, 1994). In the presentation stage, a description of the current problem is input to the CBR systems. The system then retrieves the closest-matching cases stored in a case base and uses the current problem and closest-matching cases to construct a solution to the current problem. The solution is later validated through feedback from the user or the environment. Finally, the validated solution can be added into the case base for use in future problem-solving if appropriate.

In general, a CBR system consists of a database of previous cases and their corresponding solutions, features for retrieving previous cases and storing new cases, a function or functions for measuring the degree of match, and methods for adapting recalled case solutions. A case represents specific knowledge tied to a context and records knowledge at an operational level (Kolodner, 1993). In a case-based approach, representing a case with features tied to a con-

text is an important issue. A CBR system first gains an understanding of the problem by collecting case feature values. A similarity function or functions are used to compute the degree of match between the input case and the target case.

Every feature in the input case is matched to a corresponding feature in the retrieved case. For each feature in the input case, a corresponding feature is found in the retrieved case. The two values are then compared and the degree of match is computed. A weight is usually assigned to each case feature representing the importance of that feature to the match. A nearest-neighbor matching function which contains the weights in the formula is shown in the following equation (Eq. (1)) (Kolodner, 1993). Usually, cases with higher degrees of match are retrieved.

$$\frac{\sum_{i=1}^{n} W_i^* \mathrm{sim}(f_i^{\mathrm{I}}, f_i^{\mathrm{R}})}{\sum_{i=1}^{n} W_i} \tag{1}$$

where f_i^{I} is the value of the ith feature for the input case; f_i^{R} is the value of the ith feature for the retrieved case; sim() is the similarity function that exams the degree of similarity between f_i^{I} and f_i^{R}, W_i is the weight applied to the sim() of the ith feature.

There are many evaluation functions for measuring the degree of feature match. A simplified version of a similarity function can be a standard Euclidean-based distance function shown in the following formula (Eq. (2)).

$$\sqrt{\sum_{i=1}^{n} W_i \times (f_i^{\mathrm{I}} - f_i^{\mathrm{R}})^2} \tag{2}$$

where W_i is the weight of the ith feature, f_i^{I} and f_i^{R} are the value of feature i in the input and retrieved case, respectively.

To design an appropriate case-matching mechanism in the retrieval stage, several approaches have been presented to improve case retrieval effectiveness. These include the parallel approach (Kolodner, 1988), goal-oriented model (Seifert, 1988), decision trees induction approach (Quinlan, 1986; Utgoff, 1989), instance-based learning algorithms (Kohavi, Langley, & Yun, 1995), fuzzy logic method (Jeng & Liang, 1995), and etc. These methods have been demonstrated effective in case retrieval. However, most of these researches focused on similarity functions rather than determining a set of optimal weights for the case features (Chiu, 2002).

The feature weights can be statically assigned to a set of prior known fixed values or all set equal to 1 if no arbitrary priorities are determined. However, the retrieved solution cannot always be guaranteed if the weights are determined using human judgment. A mechanism for determining a set of optimal weights could also improve case retrieval effectiveness. Kohavi et al. (1995) observed that feature weighting methods have superior performance compared to feature selection methods. However, the search space for determining the most appropriate weight for each case feature is usually quite huge. This is because the search pro-
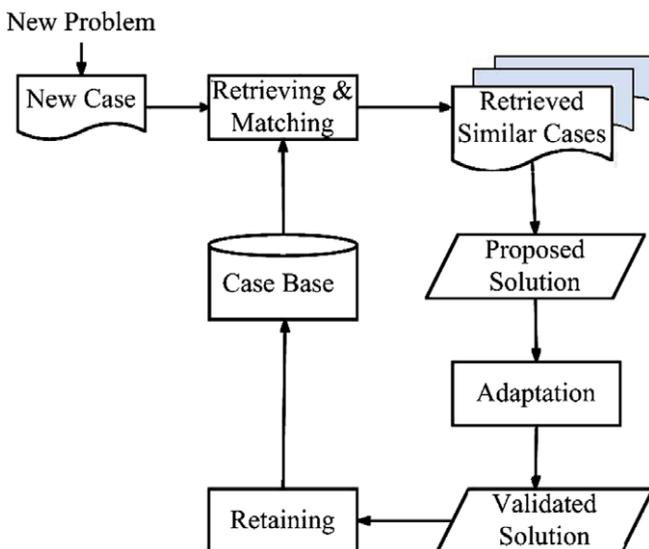

Fig. 1. CBR cycle (adaptation from Bradley, 1994).