PERGAMON

Expert Systems with Applications 22 (2002) 163–168

Expert Systems with Applications

www.elsevier.com/locate/eswa

A case-based customer classification approach for direct marketing

Chaochang Chiu*

Department of Information Management, Yuan Ze University, Neili 320, Taiwan, ROC

Abstract

Case-based reasoning (CBR) shows significant promise for improving the effectiveness of complex and unstructured decision making. CBR is both a paradigm for computer-based problem-solvers and a model of human cognition. However the design of appropriate case retrieval mechanisms is still challenging. This paper presents a genetic algorithm (GA)-based approach to enhance the case-matching process. A prototype GA–CBR system used to predict customer purchasing behavior is developed and tested with real cases provided by one worldwide insurance direct marketing company, Taiwan branch. The results demonstrate better prediction accuracy over the results from the regression-based CBR system. Also an optimization mechanism is integrated into the classification system to reveal those customers most likely and most unlikely customers to purchase insurance.

Keywords: Direct marketing; Case-based reasoning; Genetic algorithms; Customer classification

1. Introduction

Case-based reasoning (CBR) shows significant promise for improving the effectiveness of complex and unstructured decision making. It is a problem-solving technique that is similar to the decision making process used in many real world applications. CBR is both a paradigm for computer-based problem-solvers and a model of human cognition. The reasoning mechanism in the CBR system is based on the synergy of various case features. Therefore this method differs from a rule-based system because of its inductive nature. That is CBR systems reason using analogy concepts rather than the pure decision tree (or IF-THEN rules) usually adopted in rule-based systems.

Basically the CBR core steps are (1) retrieving past cases that resemble the current problem; (2) adapting past solutions to the current situation; (3) applying these adapted solutions and evaluating the results; and (4) updating the case base. Basically CBR systems make inferences using analogy to obtain similar experiences for solving problems. Similarity measurements between pairs of features play a central role in CBR (Kolodner, 1992). However the design of an appropriate case-matching process in the retrieval step is still challenging. Some CBR systems represent cases using features and employ a similarity function to measure the similarities between new and prior cases (Shin & Han, 1999). Several approaches have been presented to improve the case retrieval effectiveness. These include the parallel approach (Kolodner, 1988), goal-oriented model (Seifert, 1988), decision trees induction approach (Quinlan, 1986; Utgoff, 1989), domain semantics approach (Pazzani and Silverstein, 1991), instance-based learning algorithms (Aha, 1992), fuzzy logic method (Jeng & Liang, 1995), etc. These methods have been demonstrated effective in retrieval processes. However, most of these research works focused on the similarity function aspect rather than synergizing the matching results from individual case features. In essence when developing a CBR system, determining useful case features that are able to differentiate one case from others must be resolved first. Furthermore the weighting values used to determine the relevance of each selected feature has to be assigned before proceeding with the case matching process. Rather than being precisely or optimally constructed, the weighting values are usually determined using subjective judgment or a trial and errors basis. To provide an alternative solution this article presents a genetic algorithm (GA)-based approach to automatically construct the weights by learning the historical data. A prototype CBR system used to predict which customers are most likely to buy life insurance products is developed. The data provided by one worldwide insurance direct marketing subsidiary in Taiwan was used for constructing this model. The results show that the GA-based design of CBR system generates more accurate and consistent decisions than the regression-based CBR system.

* Fax: +886-3-435-2077.
E-mail address: imchiu@im.yzu.edu.tw (C. Chiu).

0957-4174/02/$ - see front matter
PII: S0957-4174(01)00052-5

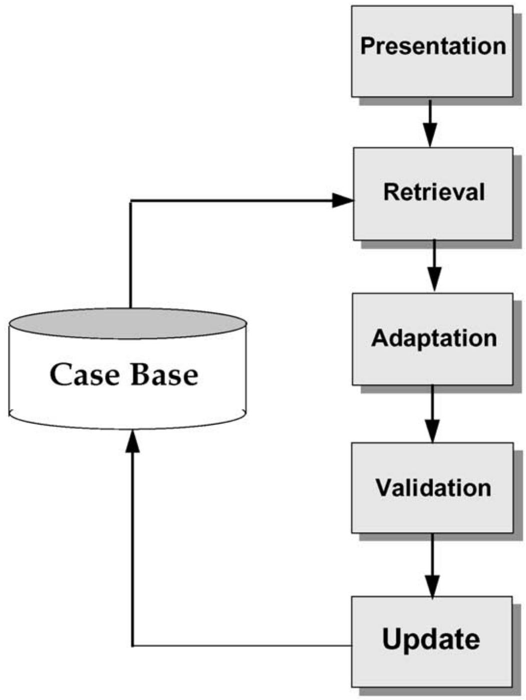

Fig. 1. The general CBR process.

2. An overview of CBR

Analogy, one way of human reasoning, is the inference that a certain resemblance implies further similarity. The CBR is a similar machine reasoning that adapts previous similar cases to solve new problems. It can be considered as a five-step reasoning process shown in Fig. 1 (Bradley, 1994).

Presentation: a description of the current problem is input into the system.
Retrieval: the system retrieves the closest-matching cases stored in a case base (i.e. a database of cases).
Adaptation: the system uses the current problem and closest-matching cases to generate a solution to the current problem.
Validation: the solution is validated through feedback from the user or the environment.
Update: if appropriate, the validated solution is added to the case base for use in future problem-solving.

Case retrieval searches the case base to select existing cases sharing significant features with the new case. Through the retrieval step, similar cases that are potentially useful to the current problem are retrieved from the case base. That is, previous experience can be recalled or adapted for the solution(s) to the current problem and mistakes made previously can be avoided.

The computing of the degree of similarity between the input case and the target case can usually be calculated using various similarity functions among which the nearest-neighbor matching is one of the frequently used methods.

2.1. Nearest-neighbor matching

Nearest-neighbor matching is a quite direct method that uses a numerical function to compute the degree of similarity. Usually, cases with higher degree of similarities are retrieved. A typical numerical function (Eq. (1)) is shown in the following formula (Kolodner, 1993).

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

where W_i is the weight of the ith feature, 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, and sim() is the similarity function for f_i^{I} and f_i^{R}.

The implicit meaning of nearest-neighbor matching is that each feature of a case is a dimension in the search space. A new case can be added into the same space according to its feature values and the relative importance of the features. The nearest-neighbors identified can then be presented as similar cases. However, the matching process can only be executed with prior known weighting values as well as clearly defined similarity functions. Most of times weighting values are determined using human judgement, and thereby the retrieved solution(s) cannot always be guaranteed. Though Brieman et al. argued that the nearest-neighbor algorithms are sensitive to the similarity functions (Brieman, Friedman, Olshen, & Stone, 1984), the additional effects from weighting synergy could leverage the potential uncertainty. Wettschereck, Aha, and Mohri, (1997) organized feature weighting methods and summarized that feature weighting methods have a substantially higher learning rate than k-nearest-neighbor. Kohavi, Langley, and Yun, (1995) described the evidence that feature weighting methods lead to superior performance as compared to feature selection methods for tasks where some features are useful but less important than others. Though Kelly and Davis (1991) proposed a GA-based, weighted K-NN approach to attain lower error rates than the standard K-NN approach, seldom has other study focused on the non-linear feature value distance relationship between an old case and an input case. To overcome this shortcoming in the traditional case retrieval process, this study presents a GA approach to support the determination of the most appropriate weighting values for each case feature.

3. The genetic algorithm approach

GA is an optimization technique inspired by biological evolution (Holland, 1975). Its procedure can improve the search results by constantly trying various possible solutions with the reproduction operations and mixing the elements of the superior solutions. In contrast to the traditional mathematical optimization methods that search for solutions via a blindfold, the GA works by breeding a population of new answers from the old ones using a methodology based on survival of the fittest. Based upon the natural evolution concept, the GA is computationally simple and powerful in its search for improvement and is able to rapidly converge by continuously identifying solutions that are globally optimal within a large search space. By using the random selection mechanism, the GA has been proven to be theoretically robust and empirically applicable for searching in complex spaces (Goldberg, 1989).

To determine a set of optimum weighting values, the search space is usually quite huge. This is because the search process must consider countless combinations of variant possible weighting values for each of the feature against all of the cases stored in the case base. Therefore, traditional approaches such as heuristic or enumerative search methods are lack of efficiency due to the enormous computing time.

To solve a problem, the GA randomly generates a set of solutions for the first generation. Each solution is called a chromosome that is usually in the form of a binary string. According to a fitness function, a fitness value is assigned to each solution. The fitness values of these initial solutions may be poor. However, the fitness values will rise as better solutions survive in the next generation. A new generation is produced through the following three basic operations.

1. *Reproduction*: solutions with higher fitness values will be reproduced with a higher probability. Solutions with lower fitness value will be eliminated with a higher probability.
2. *Crossover*: crossover is applied to each random mating pair of solutions. For example, consider solutions S_1 and S_2 (Fig. 2a). By randomly choosing the location for the separator (as symbol | shown in Fig. 2b), a simple crossover can be applied to yield $S_1{}'$ and $S_2{}'$ as new offspring (Fig. 2c).
3. *Mutation*: with a very small mutation rate P_m, a mutation occurs to arbitrarily change a solution that may result in a much higher fitness value.

$S_1 =$ 0 1 1 0 1 0 1 1
$S_2 =$ *0 0 1 0 1 1 0 0*

(a)

$S_1 =$ 0 1 | 1 0 1 0 1 1
$S_2 =$ *0 0* | *1 0 1 1 0 0*

(b)

$S'_1 =$ 0 1 *1 0 1 1 0 0*
$S'_2 =$ *0 0* 1 0 1 0 1 1

(c)

Fig. 2.

3.1. The fitness function

The objective of the proposed GA approach is to determine a set of weighting values that can best formalize the match between the input case and the previously stored cases. The GA is used to search for the best set of weighting values that are able to promote the association consistency among the cases. The fitness value in this study is defined as the number of old cases whose solutions match the input case(s) solution, i. e. the training case(s). In order to obtain the fitness value, many procedures have to be executed beforehand. Fig. 3 presents the overall system framework. The details of the system processes are shown in the system architecture as follows.

3.2. The system architecture

The system is composed of three major processes. One case base includes both training cases and old cases. The Similarity Process computes the similarity between an input training case and an old case. The similarity value (named Overall Similarity Degree (OSD)) is derived by summing each degree of similarity resulting from comparing each pair of corresponding case features out of the selected training case and old case. The OSD is expressed as the following equation (Eq. (2)).

$$\mathrm{OSD} = \sum_{i=1}^{n} W_i * \mathrm{S}_{i,j,k}^{e_i} \tag{2}$$

Where $i = 1,\ldots,n$, n is the total number of features in a case.

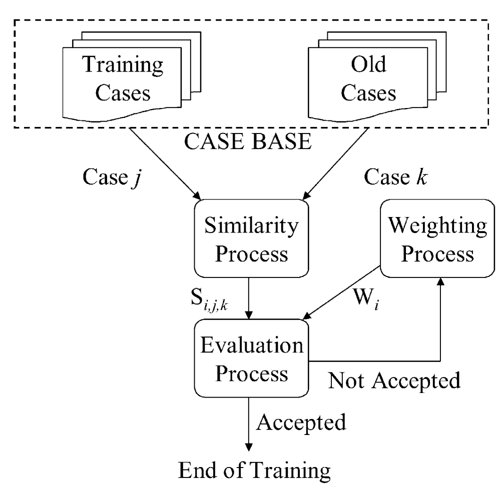

Fig. 3. The system architecture.

Table 1
The illustration of the evaluation function

Training cases	Expected outcome ($\mathbf{O}'_j$)	Real outcome ($\mathbf{O}_j$)	Matched (y_j)
$Case_1$	Yes	Yes	1
$Case_2$	No	Yes	0
⋮	⋮	⋮	⋮
$Case_j$	No	No	1
⋮	⋮	⋮	⋮
$Case_p$	⋮	⋮	⋮
Total	Evaluation function		$Y = \sum_{j=1}^{p} y_j$

W_i is the weighting value assigned to the ith feature. This value is generated from the Weighting Process by the GA. e_i represents the power of $S_{i,j,k}$, which represents the degree of similarity for the ith feature between the training case j and the old case k ($j = 1$ to p; $k = 1$ to q). p is the total number of training cases; while q is total number of old cases. $S_{i,j,k}$ is used as an index to describe the similarity level for certain case features for one training case against that of one old case. This can be expressed as the following equation (Eq. (3)).

$$S_{i,j,k} = 1 - [|\text{Feature}_{\text{Case}_j} - \text{Feature}_{\text{Case}_k}|/\text{Range}_i] \qquad (3)$$

where Range_i is the longest distance between two extreme values for the ith feature.

3.3. The evaluation process

As mentioned earlier, OSD is the key determinant for assessing the similarity between the input case and the old case. For each the Similarity Process batch executed with a specific input case, Case_j, then q OSDs are produced since q old cases have been compared with the input Case_j. The basic notation is that the higher the OSD, the more likely the retrieved old case matches the input case. Therefore the challenge becomes how to determine the most appropriate case(s). The purpose for introducing the GA is to determine the most appropriate set of weighting values that can direct a more effective search for higher OSDs to match the input case.

Usually there may exist several old cases that are inferred to be similar (either exact or near identical) to the input case. In other words, the solution (i.e. the outcome feature) of each old case could be proposed as a solution for a certain training case. To determine which cases whose outcome feature can be adopted as the outcome feature for the input case, this research proposes that for the majority of the outcome feature among the top 10% OSDs in those old cases are used to represent the final solution for each batch of the Similarity Process execution for a given training Case_j. The derived final expected outcome feature is denoted as $\mathbf{O}'_j$ as opposed to the real outcome feature $\mathbf{O}_j$ for a given training Case_j.

However for a complete run of the Similarity Process, p outcome features will be determined for each training case. Hence the Weighting Process is applied to minimize the overall difference between the original real outcome features and the expected outcome features. In other words, the more chance $\mathbf{O}'_j$ is deemed as equal to $\mathbf{O}_j$, the higher the probability that the appropriate weighting values can be produced. According to the illustration in Table 1, the evaluation function is expressed as the following equation (Eq. (4)).

Maximize

$$Y = \sum_{j=1}^{p} y_j \qquad (4)$$

where p is the total number of training cases; y_j is the matched result between the expected outcome and the real outcome, if $\mathbf{O}'_j = \mathbf{O}_j$ then y_j is 1; otherwise y_j is 0.

4. The experiment and results

4.1. Data description

Customer classification is an important issue in real world marketing. It is believed that the more understanding the corporation has about its customer behavior patterns, the greater the chance that more effective marketing strategies can be developed. This research adopts the GA–CBR method to classify potential customers into either purchasing or non-purchasing categories. Data derived from real world insurance customer was collected by the direct marketing department and separated into learning and testing groups for model construction. In order to develop a model able to effectively differentiate purchasing customers from non-purchasing customers, all possible factors such as customer demographics and other supporting information were collected. The supporting information from experienced domain experts and information from

Table 2
The description of case features

Feature (I)	Data type	Content	Range
Gender	Character	F: female; M: male	[a]
Marital status	Character	Y: married; N: single: U: unknown	[a]
Number of child	Integer	Range: [1–8]	8.0
Age	Integer	Range: [1–70]	70.0
Profession	List	Range: [1–10]	[a]
Zip code	Integer	Three-digits zip code	[a]
Area saving rate	Integer	Range: [1–27]	27
Purchasing potential (predicted outcome)	Character	Y: Yes; N: No	

[a] Indicating the range is not required. The features were matched using heuristic rules.

Table 3
The learning results of weighting values and powers

Weight values		Powers	
W_1	0.01	e_1	2
W_2	0.01	e_2	1
W_3	0.01	e_3	3
W_4	0.07	e_4	1
W_5	0.72	e_5	3
W_6	0.17	e_6	2
W_7	0.01	e_7	2

professional reports were collected to support the feature selection process (Kenneth & David, 1987; Reynolds & Wells, 1977). These factors were further investigated using statistical factor analysis to reveal the most influential factors that would have substantial impact upon the outcome.

The substantial factors derived as case features include *Gender*, *Marital Status*, *Number of Child*, *Age*, *Profession*, *Zip Code*, *Area Saving Rate*. The data model and detail explanation of these features is described in Table 2. Four hundred and forty customer profiles were averaged and mixed with purchasing and non-purchasing cases and learned by the multiple logistic regression and the GA–CBR. The 400 GA–CBR cases were used for model construction with 40 cases used for testing.

4.2. GA control parameters

The key parameters consisting of population size, crossover rate, and mutation rate needed to be defined first when developing GA computer programs. A theory that can concisely guide the assignment of these values is rarely seen (Srinivas & Patnaik, 1994). Initially, the following values were adopted in this research.

Population size	50
Crossover rate	0.5
Mutation rate	0.1

4.3. The developed model and forecasting results

After the 1000th generation in the GA training process, the best approximate weighting values and powers for the similarity values are shown in Table 3.

Once these derived values were applied to the case features, the GA–CBR system produced an accuracy or 'Best-Fit' of 77% from the targets (purchasing outcomes) over the training data, the regression model produced an accuracy of 50%. The GA–CBR demonstrated an accuracy of 65% with the test data. The regression model accuracy was 45%. These results are shown in Fig. 4.

4.4. Exploring the customers with the most potential

Basically prediction models are used to map the inputs to determine the outcome(s). However when the model is complex, is not possible to easily figure out the appropriate inputs that can best approximate the expected target. This situation applies to GA–CBR as well. In order to further explore those customer types that are most likely and most unlikely to purchase the insurance products, an additional procedure is required to aid the search process. This research adopted another GA computer program to determine the three types of customers most likely and most unlikely customers to purchase insurance. These customer types were presented in Table 4. Such information provides highly strategic values for the following campaign management.

5. Discussion and conclusion

Defining appropriate feature weighting values is a crucial issue for effective case retrieval. This paper proposed the GA-based approach to determine the fittest weighting values for improving the case identification accuracy. Compared to the regression model, the proposed method has better learning and testing performance. In this study the proposed GA-based CBR system is employed to classify potential customers in insurance direct marketing. The results show significant promise for mining customer purchasing insights that are complex, unstructured, and mixed with qualitative and quantitative information. By using the GA's rapid search strengths

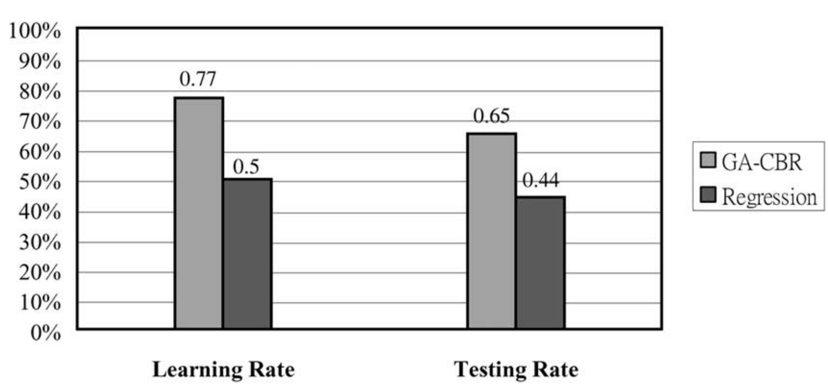

Fig. 4. Learning accuracy vs. testing accuracy for—CBR and regression model.

Table 4
The most likely and unlikely purchasing customer types

	Type	Gender	Marital status	Number of child	Age	Profession	Zip code	Area saving rate
Most likely customers	A	F	Y	1	40	3	540	27
	B	M	N	4	64	7	540	27
	C	M	Y	4	52	2	570	26
Most unlikely customers	D	F	Y	3	58	2	120	19
	E	M	N	4	60	2	120	19
	F	F	N	4	55	6	650	23

this system is able to determine the optimum customer characteristics that reflect the customer features that are most likely and unlikely to buy the insurance products. This system has not only demonstrated its better performance for prediction but also the ability to understand a model. While traditional approaches may provide many similar capabilities, other types of business data can be intensively investigated and tested to assure the GA–CBR strength in modeling classification problems. Because the similarity functions may influence the case association process, future research may work on different combinations of similarity functions between case features to examine their retrieval effectiveness.

Acknowledgements

This research was partially supported by National Science Council, Taiwan, Republic of China, under the contract number NSC85-2416-H-155-002.

References

Aha, D. W. (1992). Tolerating noisy, irrelevant and novel attribute in instanced-based learning algorithms. *International Journal of Man–Machine Studies*, *36*, 267–287.

Bradley, P. (1994). Case-based reasoning: Business applications. *Communication of the ACM*, *37* (3), 40–43.

Brieman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*, Belmont, CA: Wadsworth International Group.

Goldberg, D. (1989). *Genetic Algorithms in search, optimization, and machine learning*, Reading, MA: Addison-Wesley.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*, Ann Arbor: The University of Michigan Press.

Jeng, B. C., & Liang, T. P. (1995). Fuzzy indexing and retrieval in case-based systems. *Expert Systems with Applications*, *8* (1), 135–142.

Kelly, J.J.D., Davis, L. (1991). A hybrid genetic algorithm for classification. *International Joint Conference on Artificial Intelligence* pp. 645–650.

Kenneth, E. R., & David, W. S. (1987). *Consumer behavior and the practice of marketing*, Reading, MA: Addison-Wesley.

Kohavi, R., Langley, P., Yun, Y. (1995). *Heuristic Search for Feature Weights in Instance-Based Learning*. http://citeseer.nj.nec.com/context/324914/0, 2001/8/2.

Kolodner, J. L. (1988). Retrieving events from a case memory. *Proceedings of the Case-Based Reasoning Workshop* pp. 233–249. San Mateo, CA: Morgan Kaufmann.

Kolodner, J. L. (1992). An introduction to case-based reasoning. *Artificial Intelligence Review*, *6* (1), 3–34.

Kolodner, J. L. (1993). *Case-based reasoning*, Los Altos, CA: Morgan Kaufmann.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1* (1), 81–106.

Reynolds, F. D., & Wells, W. D. (1977). *Consumer behavior*, New York, USA/Maiderherd: McGraw-Hill.

Seifert, C.M. (1988). A retrieval model for case-based memory. *Proceedings of the 1988 AAAI Case-Based Reasoning Workshop* pp. 120–125. San Mateo, CA: Morgan Kaufmann.

Shin, K., & Han, I. (1999). Case-based reasoning supported by genetic algorithms for corporate bond rating. *Expert Systems with Applications*, *16* (2), 85–95.

Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer*, *27* (6), 17–26.

Utgoff, P. E. (1989). Incremental induction of decision trees. *Machine Learning*, *4*, 161–186.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, *11* (1–5), 273–314.