International Journal of Computational Intelligence Systems, Vol.3, No. 3 (September, 2010), 256-265

Improving the Reliability of Case-Based Reasoning Systems

Xu Xu , Ke Wang*, Weimin Ma and Jie Lin

School of Economics and Management, Tongji University, 1239 Siping Road
Shanghai, 200092, China

E-mail: xuxusuccess5616@yahoo.com, mr.wangkk@gmail.com,
mawm@tongji.edu.cn, jielinfd@263.net

Received: 21-03-2009
Accepted: 09-09-2009

Abstract

Case-based reasoning (CBR) infers a solution to a new problem by searching a collection of previously-solved problems for cases which are similar to the new problem. The collection of previous problems and their associated solutions represents the CBR system's realm of expertise. A CBR system helps to exploit data so that smarter decisions can be made in less time and/or at lower cost. A key issue is that can we always trust the solutions suggested by a case-based reasoning system? This paper studies the reliability of CBR systems based on previous study results, factors affecting the reliability of a CBR system are also discussed in this paper, especially the property that whether inter-feature of case exists redundancy. After that, the reliability of an individual suggested solution is studied. To illustrate these ideas, some experiments and their results are discussed in this paper. The results of experiments show a new route concerning on how to improve the reliability of a CBR system at an overall level.

Keywords: Case-based reasoning, Reliability, Redundancy, Compatibility.

1. Introduction

Case-based reasoning (CBR), broadly construed, is the process of solving new problems based on the solutions of similar past problems. CBR is a computer technique, which combines the knowledge-based support philosophy with a simulation of human reasoning when past experience is used, i.e. mentally searching for similar situations happened in the past and reusing the experience gained in those situations[1]. The concept of case-based reasoning is founded on the idea of using explicit, documented experiences to solve new problems. The decision-maker uses previous explicit experiences, called cases, to help him solve a present problem.

A CBR system helps to exploit data so that smarter decisions can be made in less time and/or at lower cost. It has been applied in many fields in real life. For example, Choy et al. utilize[2] CBR to develop an intelligent customer-supplier relationship management system, this is increasingly important for corporations to establish better relationships with customers and suppliers; Mansar and Marir employ[3] CBR as a technique for knowledge Management in business process redesign (BPR); Augusto et al. present[4] a framework for decision-making in relation to disaster management with a focus on CBR.

One of the issues in using an automated problem solver is how to obtain a measure of "trustworthiness" that allows the user to decide whether to

*Corresponding author.

use the solution suggested by the solver[5]. So the trust of the solutions suggested by a case-based reasoning should be considered and studied, i.e. how about the reliability of a CBR system and its suggested solutions? This is a key issue in this methodology, and it is particularly important in critical applications where a wrong solution can have serious consequences.

In IEEE Standard Computer Dictionary[6], the term reliability is defined as the ability of a system or component to perform its required functions under stated conditions for a specified period of time. For a case-based reasoning system, its required function is to provide effective solutions to problems to be solved via employing the case-based reasoning methodology. This paper discusses the factors affecting the reliability of a CBR system first, i.e. the factors which affect the ability of the system to perform its function as an intelligent problem-solving assistant.

Based on previous study, we accentuate the attribute selection-oriented factors in this paper. This sort of factors affects the representation of cases in a library, and somewhat overlaps with the case-oriented factors. However, since its significance to the reliability of case based reasoning, it is picked out as a separate sort of factors here. In other words, there are mainly four sorts of factors, i.e. attribute selection-oriented factors, case-oriented factors, algorithm-oriented factors, and human- oriented factors, which affect the reliability of CBR systems. Within the case-oriented factors, one property is studied in detail, that whether a case library is compatible with the foundational assumption of case-based reasoning that “similar problems have similar solutions”. To discuss how this property affects the reliability of a CBR system, a measure named *compatibility* of case library, which describes to what extent a case library supports this assumption at a general level, is proposed.

After the *compatibility* is presented, comparing with the related work on the trustworthiness or confidence of a case-based reasoning solution, the reliability of an individual suggested solution is studied. In contrast with the reliability of a CBR system at an overall level, the reliability estimated for individual solution provides additional information varying from case to case. As it is useful to answer the question that when a solution can be trusted or when a solution is not so reliable, this could be beneficial for consequential actions based on this suggested solution in practice.

In summary, this paper organized as follows. Firstly, the factors affecting CBR systems are presented and supplemented. Secondly, by employing the proposed measure, how to estimate that, whether case-based reasoning is a suitable methodology for the domain of application, is discussed in detail. Finally, it is concluded that the reliability of a CBR system in overall level can be improved by identifying the reliable solutions and reducing the number of noise cases.

2. Factors Affecting the Reliability of CBR Systems

2.1. Attribute selection-oriented factors

Every case has many attributes usually, attribute (feature) selection will be important to a CBR system. Its merit lies in the fact that each attribute selected corresponds uniquely to a instance which could become a target of interest itself. There are four strategies for the ranking of attribute selection: Individual ranking, group ranking, forward selection, and backward elimination (see Ref. 7 for more details).

The individual ranking scheme is the simplest to processing. It only considers the individual feature rankings. This happens to be one of the most popular as it is computationally most efficient. However, it fails to take into account the inter-feature redundancy that abound in data. For example, it is very possible that the two highest-rank individual features share a great degree of similarity. As a result, the selection of both features would amount to a wastage of resources.

This problem can be solved from Ref. 8 that if the exhaustive group ranking is adopted. The relevance/redundancy of all the features in a group is examined collectively. This of course leads to a truly optimal decision. While the group ranking strategy is the only means to yield a truly optimal selection, it is computationally prohibitive, if not impossible. So, it is rarely adopted in practice because of its demanding computational cost.

It is necessary to strike a compromise between these two extreme approaches. In this sense, One effective solution is the consecutive search approach, which falls into two main categories: Forward selection and backward elimination (see Ref. 9).

- Forward selection (FS): Such a search usually begins at an empty feature set, and iteratively adds features based on how much added value it brings to the existing subset. A key drawback is that once a feature is admitted, it cannot be discarded. The selection pool is recursively updated as follows: $U \longrightarrow U^+$, where U and U^+denote the subsets before and after the update. It recursively evaluates every candidate feature to be added into the current set on a one-by-one basis. The evaluation considers fully the inter-feature dependence with respect to the current subset.
- Backward elimination (BE): This approach begins with the full feature set and then eliminate those that do not make a significant contribution, i.e. those whose absence causes the smallest negative impact in terms of the net information loss. A key drawback is that once a feature is eliminated, it cannot be reselected in the future. The selection pool is recursively updated as follows: $U \longrightarrow U^-$, where U and U^- denote the subsets before and after the update. It recursively evaluate every features in the current set to be eliminated on a one-by-one basis. Such an approach takes into account the inter-feature dependence existing in the current subset.

FS is usually effective when a relative small number of features (from a much larger feature set) are to be selected and retained. BE is usually preferred when relative few features are to be eliminated from the original feature set.

Under the Gaussian distribution, there exists an optimal solution which maximizes the mutual information given in Eq. (1)[10],

$$H(y) = \frac{1}{2}log_2(|R_y|) + \frac{G}{2}log_2(2\pi e) \quad (1)$$

where $H(y)$ denotes entropy, y comes from the G highest principal components, and R_y denotes the covariance matrix, with a fixed subspace dimension G.

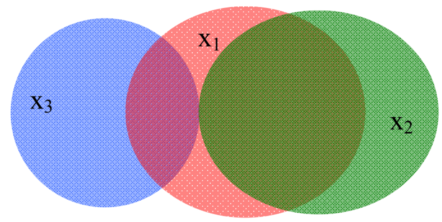

Fig. 1. With inter-feature redundancy.

For illustration purpose, we assume that the dimension of the feature is $G = 3$ and our objective is to select one or two features from the original three features (A, B and C) which represented by x_1, x_2, and x_3 (see Fig. 1). We take FS as an example. Moreover, the distribution of data is assumed to be Gaussian, assuming Gaussian distribution with a different covariance matrix:

$$\begin{pmatrix} 500 & 420 & 200 \\ 420 & 450 & 0 \\ 200 & 0 & 400 \end{pmatrix}$$

Mathematically, the inter-feature redundancy can be effectively dealt with by the entropy formula in Eq. (1). For this example, y=[x_1,x_2]^T. Thus,

$$R_y = \begin{pmatrix} 500 & 420 \\ 420 & 450 \end{pmatrix}$$

This leads to

$$H(x_1,x_2) = log_2(|R_y|) + \frac{2}{2}log_2(2\pi e)$$

The area-A represents the entropy of x_1 (= 6.53); the area-B represents the entropy of x_2 (= 6.45) and the area-C represents the entropy of x_3 (= 6.37).† It follows that the entropy of $[x_1,x_2]$ = 11.88, entropy of

†Entropy of A, B and C are under the condition of no redundancy.

$[x_1, x_3] = 12.74$, and entropy of $[x_2, x_3] = 12.82$. The total entropy for $[x_1, x_2, x_3]$ is 16.37.

A has substantial overlaps with B and C, as depicted by Fig. 1. A will be chosen in the first round. In the second round of selection, C will be a better choice to pair up with A. So, the pair to be selected is {A, C}. In this paper, we will choose FS as the method to select and optimize the attributes of case. This is a basic step to study the reliability of CBR systems.

2.2. Case-oriented factors

With the advantage that case-based reasoning systems usually require significantly less domain knowledge, case-based reasoning is often applied to solve problems where no explicit domain model exists. The solution to a new problem is based on the previously-solved problems and their associated solutions, and the case library is the main source of knowledge in a CBR system. Therefore, the performance of a CBR system relies strongly on the quality of its case library.

As case library is the main source of knowledge in a CBR system, and to propose a solution by case-based reasoning can also be treated as a process to learn knowledge from the case library, the data-oriented factors[11] affecting the reliability of knowledge discovery are also the main factors, concerning the case quality, affecting the reliability of CBR system. Both small sample size and biased sample will lead to low case quality. Hence, the case library is not a true and complete reflection of the real world. In this case, some problems to be solved may be beyond the CBR systems power, or be associated with unreliable solutions.

2.3. Algorithm-oriented factors

The algorithm-oriented factors are also an important aspect with respect to the reliability of CBR systems, especially the measurement of similarity and strategy for suggesting solution.

Case-based reasoning solves a new problem by reusing or adapting its past experience on similar previously-solved problems. Hence, the measurement of similarity is the core of all important problems concerned in a CBR system and plays a prominent role to the reliability. If the similarity of problems cannot be well measured, the solutions proposed by the system must be not so reliable and trustworthy.

However, any two things which are from one point of view similar maybe dissimilar from another point of view. Similarity always depends on the underlying contexts, and it is often used as an attempt to handle deeper or hidden relationship between objects from a more shallow level[12]. For case-based reasoning, as it is often applied to the domain with no explicit knowledge, it is more difficult to design a good measurement of similarity for different domain.

The strategy for how to propose a solution from the matched cases also plays an important role to reliability of CBR systems. A solution only referring strictly to the most similar case (determined by the measurement of similarity) may lead to over-fitting and neglect the difference between a new problem and the matched case. Some tolerance for matching can be allowed to determine what cases are similar enough. But there is not a pervasive principle to determine to what extent the tolerance should be and how to propose solutions from these similar enough cases for all applications.

2.4. Human-oriented factors

A case-based reasoning system usually performs as an intelligent problem-solving assistant to provide solutions for the decision maker. Just as any decision support system cannot perform independently with the decision maker, the reliability of a CBR system is affected greatly by the factors concerning human being.

In the four phases of case-based reasoning, retrieve, reuse, revise and retain, the human-oriented factors play a significant role to the reliability of a CBR system, especially in the process of reuse. In this phase, how to reuse or adapt the solutions with the matched cases to a new problem depends more on the subjective experience of a decision maker. Hence, the reliability of a CBR system is affected by human-oriented factors to an important extent.

3. The Reliability of a CBR System

3.1. Compatibility of case library

Case-based reasoning is a problem solving paradigm that in many respects is fundamentally different from other major artificial intelligence (AI) approaches. Instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, CBR is able to utilize the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. It bases on the assumption that "similar problems have similar solutions". However, in some real environments, this assumption is not always supported by the cases in the case library. In the case-based reasoning systems for classification tasks, especially, it is challenged at class boundaries, where the solution changes abruptly as the location of a target case crosses a decision boundary in the problem space[13]. What is more, it is not only at the boundary regions this assumption does not hold true, but also the cases themselves fail to support the assumption in some applications. Say, in some situation, some of the same cases may have different solutions. It may be because that the really critical factors affecting the solution are not involved in the case library.

As adopting case-based reasoning implies that the assumption "similar problems have similar solutions" holds true in the domain, it is of great significance to the reliability of a CBR system to evaluate the trustworthiness of the assumption in this domain. If, at a general level, the solutions of similar problems do not apply to new target problems, then case-based reasoning is not a suitable problem-solving approach for the domain. Therefore, we mainly consider how this property affects the reliability of a CBR system in this section. A measure named *compatibility* of case library, which describes to what extent a case library supports this assumption "similar problems have similar solutions" at a general level, is proposed to evaluate the trustworthiness of the assumption in a constructed CBR system. It is defined as follows.

Let $CL=\{C,S,F\}$ be a case library, where C is a finite nonempty set of M cases $\{C_1,C_2,\cdots,C_M\}$; S is a finite nonempty set of N solutions $\{S_1,S_2,\cdots,S_N\}$; F: $C_i \longrightarrow S_j$ is the rule of correspondence established between C and S, where $1 \leqslant i \leqslant M$ and $1 \leqslant j \leqslant N$.

Denoted by $D(\oplus)$ a distance function determines the similarity ranking. For a case C_i, let $K(\mathrm{C}_i)$ denote the set of cases which have the same solution with C_i, namely

$$K(C_i) = \{x|F(x) = F(C_i), x \in C, x \neq C_i\} \tag{2}$$

Hence, for a given case C_i, its closest match with same solution can be formulated by

$$C_i^* = arg \min_{x\in K(C_i)} D(C_i, x) \tag{3}$$

An indicator is denoted by $H(C_i)$ to measure the heterogeneity of solutions to C_i's neighbors in a certain region.

$$H(C_i) = \sum_{j=1}^{N} P_j^2(C_i) \tag{4}$$

where $P_j(C_i)$ is the proportion of cases with solutions S_j in a certain region for C_i and $1 \leqslant j \leqslant N$. It is easy to see that indicator $H(C_i)$ is analogous to the Gini coefficient (see Ref. 14), which is a measure of statistical dispersion most prominently used as a measure of inequality of income distribution or inequality of wealth distribution. From the formulation, we can see that $0<H(C_i)\leqslant 1$ holds. The closer $H(C_i)$ approaches to 1, the more possible that cases in the given region have the same solution. If $H(C_i)$=1 then all cases in the region have the same solution. The closer $H(C_i)$ approaches to 0, the more equal that cases in the region fall into classes with different solutions.

Here we define the region in which $H(C_i)$ is calculated through a hypersphere centered on C_i with radius $D(C_i, C_i^*)$. Thus, $P_j(C_i)$ can be illustrated as $\forall\ x \in C$,

$$P_j(C_i) = \frac{\#\{x|D(C_i, x) \leqslant D(C_i, C_i^*), F(x) = S_j\}}{\#\{x|D(C_i, x) \leqslant D(C_i, C_i^*)\}} \tag{5}$$

The *compatibility* of a case library is the average value of $H(C_i)$ with respect to all cases in the case library. That is

$$Compatibility = \frac{1}{M}\sum_{i=1}^{M} H(C_i) \qquad (6)$$

It reflects to what extent, at a general level, the case library is compatible with the assumption"similar problems have similar solutions". A high *compatibility* indicates more compatible with the assumption, hence case-based reasoning is a suitable methodology for this domain; while a low *compatibility* indicates more incompatible between the case library and the assumption, which may lead to unsatisfactory results with CBR.

In Fig. 2, one case library being simplified as an example is shown to illustrate how the *compatibility* is calculated. Suppose that the data are 2-dimensional vectors (after attribute selection), and the similarity between cases is determined by the Euclidean distance. Three different symbols in the figure stand for three classes, i.e. case C_1 and C_5, C_2 and C_3, C_4 and C_6 belong to the same class respectively.

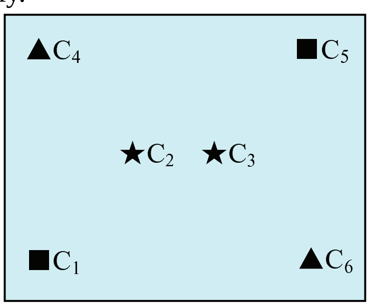

Fig. 2. Simplified illustrative case library and its compatibility.

It is easy to see that, in Fig. 2, for C_1, its closet match with same class is C_5. In the same time, C_2, C_3, C_4 and C_6 fall into the hypersphere centered on C_1 with radius $D(C_1, C_5)$, hence $H(C_1)=(1/3)^2+(1/3)^2+(1/3)^2=0.33$. Similarly, we have $H(C_5)=0.33$, $H(C_2)=H(C_3)=1$, and $H(C_4)=H(C_6)=0.33$, then the *compatibility* of case library in Fig. 2 is 0.56.

3.2. Noise cases

It is obvious that when the *compatibility* of a case library equals to 1, it implies, that the assumption is supported by the case library perfectly. For a case library with high *compatibility* approaching to 1, the little flaw that it does not equal to 1 so perfectly may be induced by some noisy cases. Some *k*-NN-based methods for case selection can be employed to remove noisy cases effectively. These *k*-NN-based methods, Condensed Nearest Neighbor Rule, Wilson Editing method and their variations (see Refs. 15-16 for more details), have been shown to be very useful for identifying and removing noisy cases because they closely examine the *k*-nearest neighbors of each case [17].

However, if the *compatibility* is somewhat prominently far away from 1, we should be more careful to remove these "noise cases", as these "noise cases" may not be noise, but a true reflection of the real world. Concerning this situation, the *compatibility* can be modified to check whether a much looser assumption that "same problems have same solutions" hold true. For this purpose, the region in which $H(C_i)$ is calculated is restricted just at the point where C_i lies. We denote this modified *compatibility* as *compatibility*′. Related work of *compatibility*′ can be seen from Ref.18.

It should be noted that the *compatibility* of case library formulated above is influenced by the distance function $D(\cdot)$. If a case library attains a high *compatibility* with respect to a certain distance function, we can expect a satisfactory result by treating this distance function as the similarity measure in CBR. If a high *compatibility* could not be attained with a possible distance function, then we can go back to the *compatibility*′ to verify a much looser assumption. When *compatibility*′=1 holds, it implies that the same problems strictly have same solutions in the case library. However, what is more important is that more attention should be paid to the case quality when *compatibility*′ ≠1, as it implies that the same problems described in the case library may not have same solutions. This situation may be caused by a few noisy cases or that the constructed case library is not a good representation of the problems and some critical features to the solution are not in-

Published by Atlantis Press

volved in the case library.

3.3. Reliability of an individual suggested solution

We focus on the reliability of an individual suggested solution in this section, i.e. to what extent we can trust it for a single solution given by the system.

In contrast with the reliability of a CBR system at an overall level, the reliability of an individual solution provides additional information varying from case to case. Say, for different problems, a CBR system may provide solutions with different reliability. As the solution to an individual problem is not only affected by the property of a CBR system, but also depends much on the problem itself. It may be reliable for some problems, but not so reliable for some other problems. The reliability of an individual solution just provides information on that when a solution can be trusted or when a solution is not so reliable, while the reliability of a CBR system at an overall level only estimates the global or average performance of the system. Hence, the reliability of an individual solution could be beneficial for consequential actions based on its suggested solution in practice.

Considerable research effort has been focused on this issue, some of which are concerned with the term "confidence" or "trustworthiness" (see Refs. 19-20). However, we would like to use the more general term "reliability" in this paper for the purpose to provide information on what extent we can trust a solution, as the term "confidence" is usually used as a statistical term in knowledge discovery. It can be seen as synonymous with trustworthiness. For a new problem to be solved (denoted by C_i), the reliability of an individual suggested solution (denoted by S_j) to the problem is formally defined as follows.

$$Reliability(S_j) = \mathrm{P}(F(C_i) = S_j) \tag{7}$$

where $\mathrm{P}(F(C_i)=S_j)$ denotes the probability that $F(C_i)=S_j$ holds true. Say, the reliability of the solution is an estimated probability that the solution is correct.

With the perspective of reliability, some existing approaches concerning confidence and trustworthiness, and some other measures can be employed to estimate the reliability of a solution. For example, Enemy(F: E) ratio[12] can be considered for the purpose of estimating the reliability of a solution; a criterion for determining the trustworthiness of a solution is based on the relative similarities between cases in the case library (see Ref. 19).

$H(C_i)$ is also a useful indicator to estimate the reliability. The criterion employing $H(C_i)$ to determine whether a solution is reliable can be simply presented as follows: If C_i is the best matched case and $H(C_i)$=1 holds, then the solution suggested referring to C_i is reliable, or else it is unreliable.

For example, as shown in Fig. 3, the hollow square and five-pointed star stand for the test case, in the left and right of the figure, respectively. Figure in left, C_1 is the best matched case. However, $H(C_1)$=0.33. Hence, the solution suggested referring to C_1 is thought unreliable. Figure in right, C_2 is the best matched case and $H(C_2)$=1 holds, thus the solution suggested referring to C_2 is considered as a reliable one. Some experiments (see Ref. 18) have shown that, in some applications, criterion based on $H(C_i)$ performs much better than criteria based on existing approaches.

3.4. Experiments and results

To illustrate the *compatibility* of case library, some experiments and their results are discussed in this subsection. An experiment is based on the famous "iris" data (see http://archive.ics.uci.edu/ml/datasets/Iris). The iris data set contains 150 instances, with each instance belonging to one of three classes (Setosa, Versicolor, Virginica). Each of the three classes contains 50 instances. The attributes of each instance consist of 4 numeric, predictive attributes and its corresponding class. For testing, 75 instances (25 instances in each class) are selected randomly and treated as training data to generate a case library. The remaining 75 instances are treated as the test data.

A distance function $D(x,y)$:

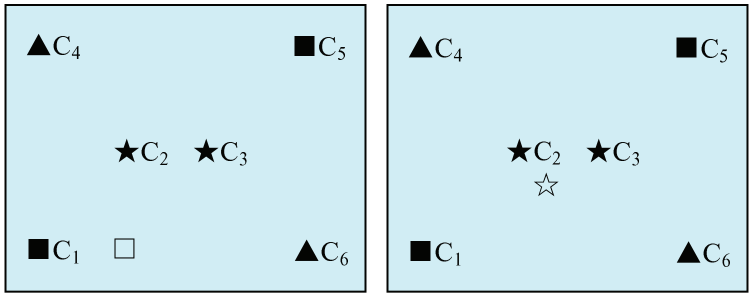

Fig. 3. H(C_i) is considered as an indicator to estimate the reliability(left: unreliable, right: reliable).

$$D(x,y) = \sum_{l=1}^{m} \frac{(x^l - y^l)^2}{\sigma_l^2} \tag{8}$$

is employed as the similarity measure, and is also used to calculate the *compatibility* of case library, where x^l denotes the value of l-th attribute of instance x, where y^l denotes the value of l-th attribute of instance y. σ_l is the standard deviation of the l-th attribute values in the case library, m is the number of predictive attributes and here m=4 holds. The experiment is repeated 5 times. As the training data is selected randomly, it has different case library every time. The performance of *compatibility* in each test can be seen in Table 1.

Table 1. Results for the iris data set.

Case library	Compatibility
Test 1	97.60%
Test 2	98.45%
Test 3	98.07%
Test 4	97.51%
Test 5	96.82%
Average	97.69%

Another experiment, on the data of Western North Pacific tropical cyclone is also considered by us. The raw data is available from an archive of tropical cyclone track data, maintained by the Joint Typhoon Warning Center (JTWC), commonly referred to as "best -tracks" (see http://metocph.nmci.navy.mil/jtwc/best_tracks/).

Some attributes, such as tropical cyclone center locations and intensities, at six-hour intervals are contained in each best-track life. The cases in each year from 2001 to 2005 are treated as training data to generate a case library respectively. The (723) cases in the year 2006 are treated as the test data, i.e. the same cases are used in each test. A same distance function as used in the experiment on iris data set is also employed in the test on tropical cyclone data. Here FS is employed to select the attributes of each case, i.e. the attributes followed the order intensity change, lasting time, intensity, pressure and moving speed‡ that describe each case. Table 2 shows the results of experiments.

As shown in Table 2, the *compatibility* of each case library is obviously improved through feature selection. Likewise, Fig. 4 shows the average value of *compatibility* that has an upward tendency, in the meantime, Fig. 4 presents the average value of conflicting cases that has a downward trend. However, in order to improve the *compatibility* of CBR, attribute selection is scientific and feasible as one of the factors affecting CBR.

‡Intensity change denotes the change of intensity in 6 hours (the difference of intensity between current observation and its previous observation). Lasting time denotes the time of the current intensity has lasted since it was observed (the time of current intensity being observed with no change). Moving speed is the averaged translation speed of the tropical cyclone in the past 6 hours. Since every observation is made with the same interval, it is denoted by the distance between the two locations of the current observation and its previous observation.

Table 2. Compatibility with different attributes.

Case library	Total cases	2 attributes	3 attributes	4 attributes	5 attributes
2001	840	47.92%	59.66%	61.84%	76.96%
2002	819	47.08%	59.99%	60.24%	74.90%
2003	836	48.05%	59.86%	60.16%	73.83%
2004	1003	46.25%	57.56%	57.73%	75.34%
2005	687	48.17%	60.94%	61.48%	72.87%
Average	837	47.49%	59.60%	60.29%	74.78%

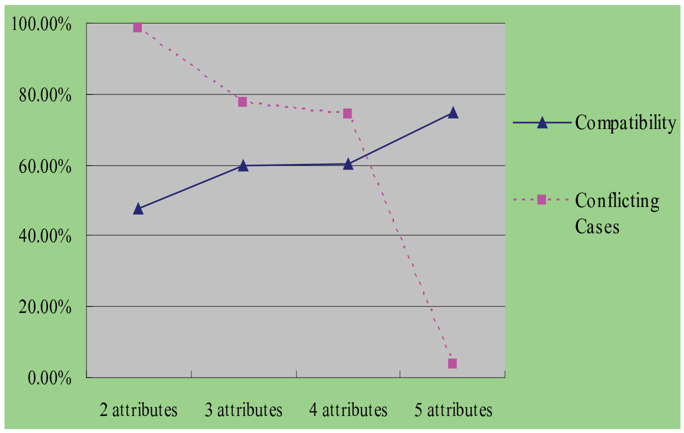

Fig. 4. Comparing of average values (Compatibility and Conflicting Cases) in each test with different attributes.

4. Conclusions

Concerning on the reliability of CBR systems at an overall level, there are mainly four sorts of factors affecting it, i.e. attribute selection-oriented factors, case-oriented factors, algorithm-oriented factors and human-oriented factors. The reliability of an individual suggested solution evaluates to what extent we can trust it for a single solution. Some existing approaches can be employed to estimate it. Well, the accuracy of reliable solutions through our proposed method is consistently higher than that of total solutions.

With respect to case quality, FS is employed, in the meantime, a measure named *compatibility* of case library is proposed to estimate whether a case library is compatible with the foundational assumption of CBR that “similar problems have similar solutions”. Experiments show if a case library will attains a high *compatibility*[§] with respect to a certain distance function, a satisfactory result by treating this distance function as the similarity measure in case-based reasoning can be expected.

Generally speaking, this paper summarizes some existing methods that concern the reliability of CBR systems. The main novel points and merits of this paper are in twofold: First, since FS is significant to the reliability of CBR, FS is employed to improve the *compatibility* of case library. Second, the discussions in this paper are mainly focused on the performance of a CBR system dealing with classification task. In addition, the experiments are based on several data sets only, therefore, a much more thorough study still should be done further. In a sense, this paper shows a new route concerning on how to improve the reliability of a CBR system at an overall level.

[§]The attributes of case are considered first.

Acknowledgements

This study is supported by the National High-Tech. R&D Program for CIMS, China (No.2007AA04Z151), Program for New Century Excellent Talents in University, China (No.NCET-06- 0172 and No. NCET-06-0377), and the National Natural Science Foundation, China (No.70521001, No.70531020 and No.70671004).

References

1. D. B. Leake, *Case-based Reasoning: Experiences, Lessons and Future Directions*, (MIT Press, Cambridge, MA, 1996).
2. K. L. Choy, K. K. H. Fan and V. Lo, " Development of an intelligent customer-supplier relationship management system: The application of case-based reasoning,"*Industry Management and Data Systems*, **103**, 263–274 (2003).
3. S. L. Mansar and F. Marir, "Case-based reasoning as a technique for knowledge management in business process redesign," *Electronic Jour. on Knowledge Management*, **1**, 113–124 (2003).
4. J. C. Augusto, H. Wang and J. Liu, "Situation assessment in disaster management,"*Intl. Jour. of Comput. Intelligence Systems*, **1**, 237–247 (2008).
5. A. G. David, "Knowing what you don't know: Roles for confidence measures in automatic speech recognition," *PhD Thesis of University of Sheffield*, (1999).
6. Institute of Electrical and Electronics Engineers, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*, (IEEE, New York, NY, 1991).
7. D. koller and M. Sahami, "Toward optimal feature selection," *Proc. Intl. Conf. on Machine Learning*, 284–292 (1996).
8. I. Guyon and A. Elisseeff,"An introduction to variable and feature selection," *The Jour. of Machine Learning Research*, **3**, 1157–1182 (2003).
9. R. Caruana and D. Freitag, "Greedy attribute selection," *Proc. Intl. Conf. on Machine Learning*, 28–36 (1994).
10. Q. Lu and X. Yao,"Clustering and learning Gaussian distribution for continuous optimization," *IEEE Trans. on Systems, Man, and Cybernetics*, **35**, 195–204 (2005).
11. H. Dai, "A study on reliability in graph discovery," *Proc. IEEE Intl. Conf. on Data Mining Workshops*, 775–779 (2006).
12. H. D. Burkhard, "Similarity and distance in case-based reasoning," *Foundamenta Informaticae*, **47**, 201–215 (2001).
13. S. Massie, S. Craw and N. Wiratungs, "When similar problems don't have similar solutions," *Proc. Intl. Conf. on Case-Based Reasoning*, 92–106 (2007).
14. C. Gini, "Measurement of inequality and incomes," *The Economic Jour.*, **31**, 124–126 (1921).
15. P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. on Information Theory*, **14**, 515–516 (1968).
16. D. R. Wilson and L. Dennis, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. on Systems, Man, and Cybernetics*, **3**, 408–421 (1972).
17. Y. Li, S. C. K. Shiu, and S. K. Pal, "Combining feature reduction and case selection in building CBR classifiers," *IEEE Trans. on Knowledge and Data Engineering*, **18**, 415–429 (2006).
18. K. Wang, J. N. K. Liu, and W. M. Ma, "A study on the reliability of case-based reasoning systems," *Proc. IEEE Intl. Conf. on Data Mining Workshops*, 60–68 (2008).
19. W. Cheetham, "Case-based reasoning with confidence," *Proc. European Workshop on Case-based Reasoning*, 15–25 (2000).
20. J. J. Chua and P. E. Tischer, "Determing the trustworthiness of a case-based reasoning solution," *Proc. Intl. Conf. on Comput. Intelligence for Modeling*, 952–962 (2004).