

Article

## Case-Based FCTF Reasoning System

Jing Lu <sup>1,2</sup>, Xiakun Zhang <sup>3,4,\*</sup>, Peiren Li <sup>2</sup> and Yu Zhu <sup>2</sup>

<sup>1</sup> School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China; E-Mail: lujing19810629@hotmail.com

<sup>2</sup> Shanxi Meteorological Administration, Taiyuan 030006, Shanxi, China; E-Mails: lpr640221@163.com (P.L.); zhuyu0351@gmail.com (Y.Z.)

<sup>3</sup> National Meteorological Center of China Meteorological Administration, Beijing 100081, China

<sup>4</sup> Environmental Modeling Center, NOAA/NWS/National Centers for Environmental Prediction, College Park, MD 20740, USA

\* Author to whom correspondence should be addressed; E-Mail: zhangxk@cma.gov.cn; Tel.: +86-10-6840-7423; Fax: +86-10-6217-5928.

Academic Editor: Chien-Hung Liu

Received: 27 July 2015 / Accepted: 29 September 2015 / Published: 16 October 2015

---

**Abstract:** Case-based reasoning uses old information to infer the answer of new problems. In case-based reasoning, a reasoner firstly records the previous cases, then searches the previous case list that is similar to the current one and uses that to solve the new case. Case-based reasoning means adapting old solving solutions to new situations. This paper proposes a reasoning system based on the case-based reasoning method. To begin, we show the theoretical structure and algorithm of from coarse to fine (FCTF) reasoning system, and then demonstrate that it is possible to successfully learn and reason new information. Finally, we use our system to predict practical weather conditions based on previous ones and experiments show that the prediction accuracy increases with further learning of the FCTF reasoning system.

**Keywords:** case-based reasoning; from coarse to fine; reasoning system

---

## 1. Introduction

Case-based reasoning (CBR) is a problem-solving method, not only making connections between problem descriptors and conclusions, but also is able to utilize the specific previous knowledge to infer new solutions for new problem situations (cases) [1,2]. A new problem is solved by finding a similar past case, and reused in this new problem situation. The CBR field has grown rapidly in recent years, as evidenced by the emergence of several international case-based reasoning conferences in European and other CBR workshops, by the usage of available commercial tools, and successful applications in daily use [3–6].

In the early 1980s, CBR traces back to the work of Roger Schank, whose model of dynamic memory was the basis for the earliest CBR systems [7]. Other types of CBR emerged in the 1980s, aiming at topics such as legal reasoning, reasoning from massively parallel machines cases, and combinations of CBR with other machine learning techniques. CBR technology resulted in deployment of a number of successful systems, such as the system for controlling composite parts to be baked in a convection oven and the system of health application area [8].

There is one type of CBR system called Conversational Case-based Reasoning (CCBR) [9], which provides a mixed-initiative dialog for guiding users to construct their problem description incrementally through a question. For each time running a CBR, the users are confronted with a new dialog or a new query that is used to decide to terminate the whole process or just go on. This is a kind of “multistep CBR” that means applying CBR and improving the query, and reapplying CBR, *etc.* The common features of our system and CCBR is “multistep”. Our CBR system is a kind of CCBR.

The main contribution of the paper is to use previous information to infer new information based on case based “from coarse to fine” (FCTF) reasoning system. Specifically, the system starts with the cases with the least features/feature to learn and predict, if it could not get the expected accuracy of the prediction, the system would increase one feature, and then repeat the step for many times. This process stops when it reaches to the expected accuracy.

The organization of this paper is as follows. In Section 2, we briefly introduce the preliminaries of the paper: case-based reasoning and Occam’s razor theory. Section 3 is the description of the case-based FCTF reasoning system. Section 4 presents how to use the case-based FCTF reasoning system to predict the precipitation. Finally, we make the conclusion.

## 2. Preliminaries

### 2.1. Case-Based Reasoning

Case-based reasoning learns from experience. A reasoner uses old experience to suggest solutions for new problems [10]. Specifically, the reasoner solves new problems by recording and reusing the most similar one or one set [11,12]. It interprets new situations by remembering old similar situations and contrasting the new one to old ones to see which one fits best. Case-based reasoning could be used as the diagnostic assistant to help users diagnose problems on their computer systems [13]. When a user describes his problems, the closest cases in the case base will be retrieved. The diagnostic system can recommend some of these to the user, adapting each case to each user’s particular situation, such as changing recommendations based on what operating system the user has, what software the user uses,

what internet connection method they choose, and the printer brand. If none of the cases is suitable, some other problem-solving method can be done to solve the problem, perhaps by having a human help diagnose the problem. When the problem is finally fixed, the case would be added to the case base.

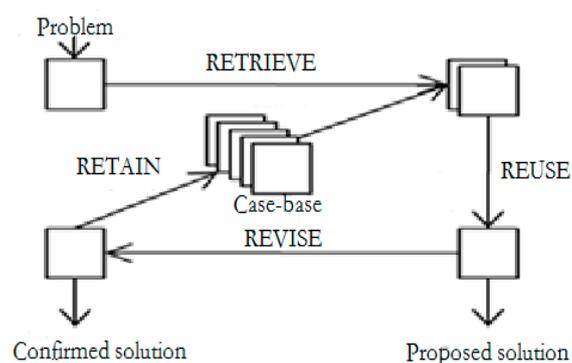
Case-based reasoning is a method that combines learning with reasoning. In case-based learning, the training samples, the cases, are stored and accessed to solve a new problem [14,15]. It is very important to find a way to represent the data. To trigger this method, a distance is required to measure the closeness of two examples. Suppose  $e(X_i)$  (shown in Table 1) is the value of feature  $X_i$  for the example  $e$ ;  $(e_1(X_i) - e_2(X_i))$  is the feature difference between example  $e_1$  and example  $e_2$ . The Euclidean metric can be used as the distance between two objects and can be calculated as the square root of the sum of the squares of the feature differences. One issue is the importance of different features; increasing the weight of one feature increases the importance of that feature. Let  $w_i$  be a non-negative real-valued parameter that specifies the weight of feature  $X_i$ . The distance between examples  $e_1$  and  $e_2$  is then  $d(e_1, e_2) = \sqrt{\sum_i w_i \times (e_1(X_i) - e_2(X_i))^2}$ . The feature weights can be provided as input.

**Table 1.** Values of feature  $X_i$  for the example  $e_1$  and  $e_2$ .

Examples/Features	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_i$	$X_n$
$e_1$	$e_1(X_1)$	$e_1(X_2)$	$e_1(X_3)$	$e_1(X_4)$	$e_1(X_5)$	$e_1(X_i)$	$e_1(X_n)$
$e_2$	$e_2(X_1)$	$e_2(X_2)$	$e_2(X_3)$	$e_2(X_4)$	$e_2(X_5)$	$e_2(X_i)$	$e_2(X_n)$

After the situations or cases are properly represented and stored, the new situation (case) will enter into the system memory that is used for reasoning for the next time. In case-based reasoning, to get a prediction for a new case, we choose the case that is similar, or close to, the new case to predict the value of the target features of the new case. In general, the second time solving the problem is easier than the first because we are more competent since we remember our mistakes and, therefore, can avoid them.

A case-based reasoning algorithm to predict whether the user will or will not read this article must determine the relative importance of the dimensions. One of the problems in case-based reasoning is accessing the relevant cases. To finish this work as quickly as possible, a *kd*-tree [16] is a way to number the training examples so that chosen examples can be found quickly. Just like a decision tree, a *kd*-tree splits on input features. In building a *kd*-tree from a set of examples, the learner tries to find a suitable feature that can split the examples into several sets that have approximately equal size, then builds sub-*kd*-trees for the cases in each partition. The exact matches will be found at the leaf. Case-based reasoning can be expressed as a four-step process shown in Figure 1 [17].



**Figure 1.** Case-based learning process.

**Retrieve:** Remembering is the process of retrieving a case or set of cases from memory. In general, it consists of two sub-steps: *recall* previous cases and *select* the best case. The goal of the first step is to retrieve suitable cases that can support reasoning that occurs in the following step. Good cases are those that have the potential ability to make a correct prediction on the new case. Given a new case, we retrieve a similar case/cases from the case base, and then select the best subset. This step selects the most promising case/cases to reason with from those generated in the *recall* step of previous cases. The purpose of this step is to narrow the set of relevant cases to a few candidates worthy of intensive consideration. Sometimes it is appropriate to choose one best case, sometimes a small set.

**Reuse:** Adapt the retrieved case or cases to match with the new case, and propose a solution.

**Revise:** Evaluate and revise the solution based on how well it works. The revision can involve other reasoning techniques, such as using the proposed solution as a basic point to come up with a more suitable solution, which can be done by a human interactive system.

**Retain:** Decide whether to keep this new case and its solution in the case base.

## 2.2. Occam's Razor

Occam's razor traces its root to the work of William Ockham, a principle of succinctness used for solving problem [18–20]. It states that the one with the fewest assumptions should be selected among a number of hypotheses. Other more complicated solutions may ultimately be proved correct, but the fewer assumptions that are made, the better, in the case of certainty absence.

Occam's razor is also called the parsimony principle [21–23]. Recently, it is usually interpreted as something like “the simpler the explanation, the better”. Occam's razor, as a principle that is frequently used, establishes the criteria for choosing among choices that have equal opportunities. For example, Occam's razor can be used to support creationism over evolution since evolution is much more complex than having God create everything, which indeed is much simpler than evolution. After all, evolution is a very complex process.

However, Occam's razor does not say that the more simple a hypothesis, the better [24]. We don't assume that the simpler theory is correct and the more complex one is false. We know from experience the theory that requires complicated machinations is often to be proved wrong. However, until proved, the more complex theory competing with a simpler explanation should still be considered. We should not safely assume that the simpler one is absolutely right. Occam's razor is useless when a rational person should make a choice between an implausible explanation and a probable one since the answer is obvious, but if the principle is minimalist, then it seems to imply that the more simplification the better. If so, then the principle of parsimony might better have been called Occam's Chainsaw, for its main use looks like a clear-cutting action [25,26].

## 3. Case-Based “FCTF” Reasoning System

As mentioned before, Occam's razor is a principle of succinctness used for solving problems, which means that the one with the fewest assumptions or features should be selected. Among the scientific methods, Occam's razor is not considered as an irrefutable existence [27–29]; the preference for simplicity is based on the falsifiability criterion [30,31]. For each accepted explanation of a phenomenon, although there is always a number of possible and more complex alternatives, simpler theories are

preferable to more complex ones since they are better testable and falsifiable. Case-based FCTF reasoning system is based on such a theory. FCTF, just as the name implies, means “from coarse to fine” and starts with the least feature or features to reason.

When performing analysis of complex data, one of the major problems is the number of involved variables; in other words, the number of features. Generally speaking, analysis with a large number of variables demands a large amount of memory, time, and computation time to generate an answer of a problem. Feature extraction is a key point for constructing the variables to solve these problems while still describing the data with sufficient accuracy.

In case-based reasoning, feature extraction is a primary step. When the input data is too complicated to be processed or suspected of having redundant information, then the input data will be transformed into a reduced representation set of features. Using the set of features to represent the input data is called feature extraction. If the features extracted are carefully chosen, it is expected that the features set will perform the desired task using this feature extraction representation.

Best results are achieved when an expert constructs a set of application-dependent features. Nevertheless, if no such expert knowledge is available, general dimensionality reduction techniques may help.

### 3.1. Case-Based Reasoning Problem Definition

Generally speaking, a case in case-based reasoning system can be represented by the three steps conceptually below:

Problem definition: the expression of the problem that needed to be solved at one time.

Solution definition: the expression of the solution to the problem specified in the problem definition step.

Outcome: the resulting state after the solution was carried out.

A new case only has the problem definition part. The similarity measurement between a query and a stored case is based on the comparison of the problem definition parts of them.

### 3.2. Case-Based “FCTF” Reasoning System

Case-based FCTF reasoning system is shown in Figure 2.

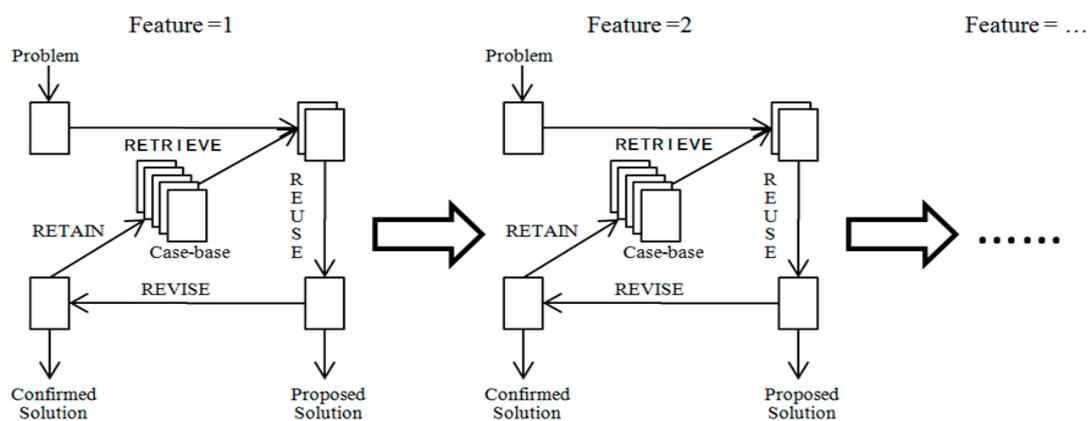


Figure 2. Case-based FCTF reasoning system.

As shown in Figure 2, the system starts with the least number of features; one, for example. If the prediction accuracy cannot be reached, we increase the number of features and do the case-based learning again.

**Retrieve:** Remembering is the process of retrieving a case or set of cases from memory. The case element may be a crisp number or a more complicated one.

**Reuse:** If we find the same case, we just set the solution of the case in the case base as the solution of the new case, if we cannot, we find the nearest case and output according to this case.

**Revise:** Evaluate the solution and revise it based on the real situation.

**Retain:** Keep this solution to the basement.

The following is the pseudocode of the reasoning algorithm.

Crisp reasoning algorithm: (1)  $n = 1$ ; (2) while(accuracy  $> \alpha$ ); (3) for  $j = 1$  to  $r$  do; (4) for  $i = 1$  to  $n$  do; (5) Computedist( $A_{(r+1)i}, A_{ji}$ ); (6) Endfor; (7) Compute  $dist(R_{(r+1)}, R_j)$ ; (8) Endfor; (9) Find the rule number with the minimum value of  $dist(R_{(r+1)}, R_j)$ :  $minj$ ; (10) Output( $y(R_{minj})$ ); (11)  $n++$ ; (12) End while.

In a case-based reasoning system, one key point is to compute the similarities between a query and old cases to decide which case is most similar to the new coming one. Normally, the similarity between a query and an old case is calculated by the accumulated similarities on all counted features.

The similarities are also influenced by the calculation methods. For example, in one method, two cases can be treated as similar, while in another method, two cases with various values on one nominal feature can possibly be considered as similar through exploring general domain knowledge. There are many methods to compute the similarities between a query and old cases.

Here, we just give an example of a calculation method, the Euclidean metric, to compare the similarity of the cases if the case is a set of features, each of which is a crisp number.

Assume that there are  $r$  rules in the rule set with the same structure, we first compute the distance between the prerequisite of the  $R_{(r+1)}$  and  $R_j$ , denote it as  $dist(A_{(r+1)i}, A_{ji})$ , which can be calculated as  $(A_{(r+1)i} - A_{ji})$ . Then, we find the rule number with the minimum value of  $dist(R_{(r+1)}, R_j)$  and denote it as  $minj$ . To calculate the Euclidean metric of the prerequisite between two rules, we compute the distance between one variable value in one rule and the corresponding variable value in the other rule, and then do the same to the next variable value and the corresponding variable value in the other rule, and so on. Finally, we calculate the  $dist(R_{(r+1)}, R_j)$  according to the equation following:

$$dist(R_{(r+1)}, R_j) = \sqrt{\sum_i (A_{(r+1)i} - A_{ji})^2} \tag{1}$$

For example, if we want to compare the Euclidean metric between  $R_1$  and  $R_2$ , we should compute the distance between the first element of the prerequisite in  $R_1$  “ $A_{11}$ ” and the first element of the prerequisite in  $R_2$  “ $A_{21}$ ” yielding  $dist(A_{11}, A_{21})$ . We then do the same for the second element of the prerequisite in  $R_1$  and the second element of the prerequisite in  $R_2$  yielding  $dist(A_{12}, A_{22})$ . At this point, we use the equation above to calculate  $dist(R_1, R_2)$ .  $\alpha$  in line 1 is a user-specified threshold to control the reasoning process. Its value determines the accuracy of the reasoning system: if  $\alpha$  is bigger, the reasoning system is more accurate.

Interval reasoning algorithm: (1)  $n = 1$ ; (2) while (accuracy  $> \alpha$ ); (3) for  $j = 1$  to  $r$  do; (4) for  $i = 1$  to  $n$  do; (5) Computedist( $avg(A_{(r+1)i}), avg(A_{ji})$ ); (6) Endfor; (7) Compute  $dist(R_{(r+1)}, R_j)$ ; (8) Endfor; (9) Find the rule number with the minimum value of  $dist(avg(A_{(r+1)i}), avg(A_{ji}))$ :  $minj$ ; (10) Output( $y(R_{minj})$ ); (11)  $n++$ ; (12) End while.

If the case of the feature is an interval, we average the values in the interval and use the averaged, crisp number to denote the interval. Afterwards, the interval reasoning algorithm is the same as the crisp reasoning algorithm.

#### 4. Experiments and Analysis

We took the most common time series predicting weather forecast for example, to test the effects of different number of input atmosphere conditions on the rainfall forecasting accuracy of our case-based FCTF Reasoning System.

Expert systems in meteorological area are expressed using rules in the following way:

Given a rule base of  $r$  rules, e.g.,  $n$ -input 1-output, the  $j^{\text{th}}$  control rule is described in the following form:  $R_j$ : IF  $x_1$  is  $A_{j1}$  AND  $x_2$  is  $A_{j2}$ , AND  $x_n$  is  $A_{jn}$ , THEN  $y$  is  $B_j$ , where  $j$  is a rule number,  $n$  is the total number of inputs. Rule-based expert systems can only deal with rules that is in the rule set.

When we want to deduce another rule that is not in the rule set above (denoted as  $R_a$ ): “ $R_a$ : IF  $x_1$  is  $A_1$  AND  $x_2$  is  $A_2$ , AND  $x_n$  is  $A_n$ , THEN  $y$  is ?”.

We can translate this rule-based expert system into a case-based reasoning system: one complete case is one complete rule; one incomplete case is one partial rule. Thus, a rule-based expert system can be transferred to a case-based reasoning system, which can deal with rules that are not in a rule set. This is somewhat like partial-rule matching algorithm [32].

In our experiments, we take Euclidean distance and cosine similarity to measure the similarities between the two instances. In our experiments, there is no difference between these two methods when we use them.

##### 4.1. Data Collection

From Shanxi Meteorological Administration, China, we got the ground surface meteorological observation data in Taiyuan, a city of Shanxi Province, China from May 2007 to October 2007, May 2008 to October 2008, and May 2009 to October 2009, which are the rainy seasons in that city. The specific meteorological elements included the day’s atmospheric pressure, relative humidity, dry and wet bulb temperature, wind speed, and accumulated precipitation at 20 o’clock (Beijing Time, equivalent to 1200 UTC), which we took into our experiments.

##### 4.2. Experimental Section

We took the day’s atmospheric pressure, relative humidity, dry and wet bulb temperature, and wind speed observations at 20 o’clock (Beijing Time) as input conditions. Before the experiments, we classified the input meteorological conditions into different levels according to their rating, which is shown in Table 2. Atmospheric pressure falls into four levels shown in the first two columns of Table 2; dry and wet bulb temperature falls into four levels shown in the third and fourth columns; relative humidity falls into five levels, shown in the fifth and sixth columns; and wind speed falls into five levels shown in the last two columns.

**Table 2.** Levels classification of input conditions.

Atmospheric Pressure		Dry and Wet Bulb Temperature		Relative Humidity		Wind Speed	
Rating	Value (hPa)	Rating	Value (°C)	Rating	Value (%)	Rating	Value (MPH)
Moderate	>940	Lowest	<-10	Dry	[0,30)	Calm	(0,2)
Lower slightly	[930, 940)	Lower	[-10,5)	Less dry	[30,50)	Light Air	[2,4)
Lower	[920,930)	Moderate	[5,30)	Less humid	[50,70)	Light Breeze	[4,7)
Lowest	<920	Higher	[30,45)	Humid	[70,100)	Gentle Breeze	[7,11)
		Highest	>45			Moderate Breeze	[11,17)

We tested our case-based FCTF reasoning systems with different numbers of input conditions in these experiments to discover the effects of number of input conditions on the case-based FCTF reasoning systems’ predicting accuracy. Table 3 shows the detailed experiment section: in experiment number one, we took only one input condition which is atmospheric pressure; in experiment number two, we took two input conditions which are atmospheric pressure and relative humidity; in experiment number three, we took three input conditions which are atmospheric pressure, relative humidity, and dry and wet bulb temperature; and in experiment number four, we took four input conditions which are atmospheric pressure, relative humidity, dry and wet bulb temperature, and wind speed. Since the same meteorological premises may lead to several different conclusions, which are different meteorological levels. In those cases, we choose the most probable level.

**Table 3.** Case-based FCTF Reasoning Systems’ experiments with different numbers of input conditions.

Experiment Number	Input Conditions			
	Atmospheric Pressure	Dry and Wet Bulb Temperature	Relative Humidity	Wind Speed
1	×	○	○	○
2	×	×	○	○
3	×	×	×	○
4	×	×	×	×

Notes: “×” denotes “including” and “○” denotes “not including”.

We also took the accumulated precipitations in the next 24 h as outputs for the weather prediction models, and used the precipitation observations to verify the prediction results of Case-based FCTF Reasoning Systems’. Numbers from zero to four inclusively is given to represent light rain, moderate rain, heavy rain, rainstorm, and heavy rainstorm, respectively. Thus, precipitation values falls into five levels shown in Table 4.

**Table 4.** Levels classification of precipitation outputs.

Precipitation Rating	Precipitation Value (mm)
Light rain	(0,10.0)
Moderate rain	[10.0,24.9)
Heavy rain	[24.9,49.9)
Rainstorm	[49.9,99.9)
Heavy rainstorm	[99.9,249.0)

Since people concern forecasting rainfalls more than forecasting sunny days [33,34], we defined *Hitrates* as:

$$Hitrates = Correctp/Totalp \tag{2}$$

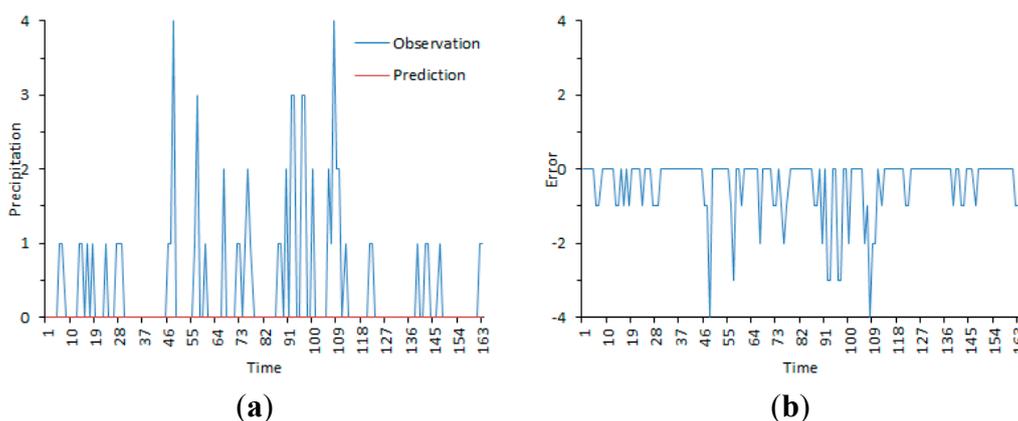
where *Correctp* is times of correct prediction in observed rainfall days and number of observed rainfall days as *Totalp*. We defined *Error* as:

$$Error = Predictp - Observp \tag{3}$$

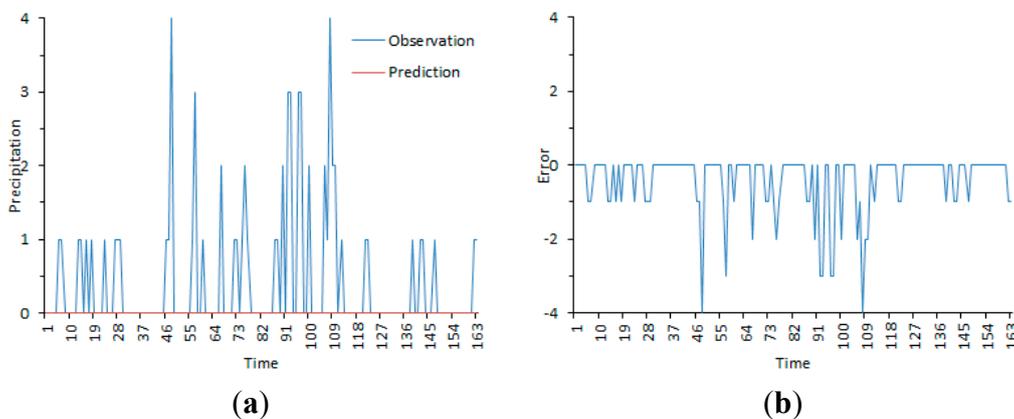
where *Predictp* is the predicted precipitation values and the observed precipitation values as *Observp*.  $|Error|$  denotes the absolute value of *Error*. For the same reason as above, we also only considered observed rainfall days in calculating average  $|Error|$ .

### 4.3. Experiment Results and Error Analysis

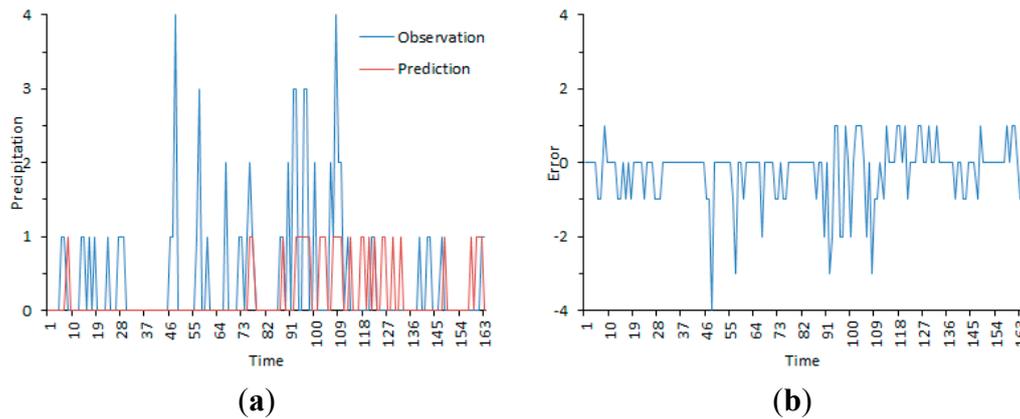
After four experiments completed, we calculated their errors and hitting rates, shown in Table 5. We also drew comparison charts between predicted values and observed values (panels (a) in Figures 3–6), and time sequence diagrams of prediction errors (panels (b) in Figures 3–6).



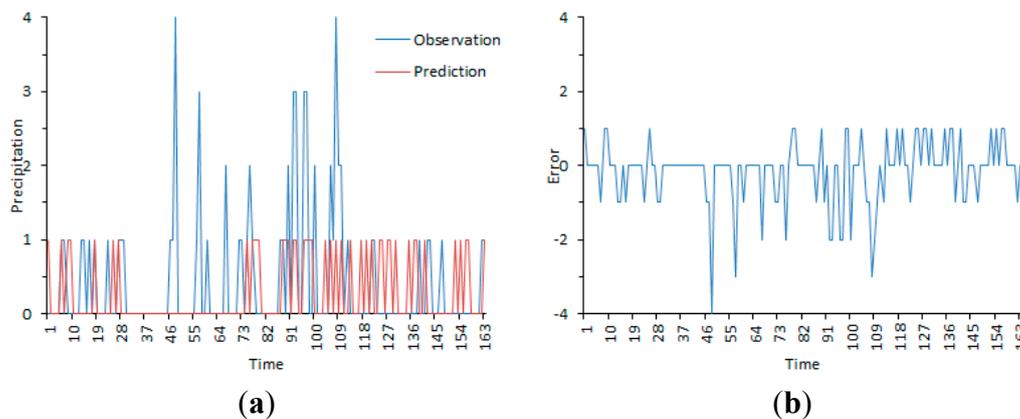
**Figure 3.** Testing results of the Case-based FCTF Reasoning System in experiment number one. (a) Comparison between Observation and Prediction; (b) Prediction error.



**Figure 4.** Testing results of the Case-based FCTF Reasoning System in experiment number two. (a) Comparison between Observation and Prediction; (b) Prediction error.



**Figure 5.** Testing results of the Case-based FCTF Reasoning System in experiment number three. (a) Comparison between Observation and Prediction; (b) Prediction error.



**Figure 6.** Testing results of the case-based FCTF reasoning system in experiment number four observed rainfall days in calculating average  $|Error|$ . (a) Comparison between Observation and Prediction; (b) Prediction error.

Firstly, the case-based FCTF reasoning systems with only one input condition and with two input conditions are used to predict the precipitation levels for weather forecasting, but lead to the non-accuracy of the prediction outputs. Table 5 shows that hitting rates in experiments number one and number two are both equal to zero, which means that the rates of prediction accuracy are zero. Figures 3a and 4a show that the case-based FCTF reasoning systems with only one input condition and with two input conditions never successfully forecasted any rainfalls during the experimental time period. However, they had forecast no rainfalls in all the days during the whole experiment, which makes the errors are always equal to or less than zero as shown in Figures 3b and 4b. Therefore, we believe that the case-based FCTF reasoning systems with only one input condition and with two input conditions are completely impossible to forecast precipitation. These results can also be explained on meteorology. Meteorologists point out that the conditions of rainfall formation are complex [35,36], and using only one or two kinds of input conditions cannot forecast precipitation.

**Table 5.** Comparisons among the case-based FCTF reasoning systems with different number of input conditions.

Experiment Number	Number of Input Conditions	Minimum Error	Maximum Error	Minimum  Error	Maximum  Error	Average  Error	Hit Rate
1	1	−4 levels	0 level	0 level	4 levels	1.52 levels	0.00
2	2	−4 levels	0 level	0 level	4 levels	1.52 levels	0.00
3	3	−4 levels	1 level	0 level	4 levels	1.25 levels	0.27
4	4	−4 levels	1 level	0 level	4 levels	1.16 levels	0.36

The case-based FCTF reasoning system with three input conditions improved heavily compared to experiments one and two, just after adding the relative humidity as the new input condition. Figure 5 shows that it can forecast some of the rainfall events. Hit rate, as shown in Table 5, which increased to 0.27, is much better than those in experiments one and two, which are both equal to zero. Average |error| plummeted from 1.52 levels to 1.25 levels, decreasing by 17.8%. This result preliminary notes that when the case-based FCTF reasoning system forecasts a complex system (such as weather system), adding a new input condition is one of the important and useful methods to increase the accuracy. This is also evident in the meteorology area. Compared with the atmospheric pressure and temperature, relative humidity has greater effect on precipitation [37–39]. However, this forecast result is still far from our expectations.

We continued with the case-based FCTF reasoning system with four input conditions, which is the improvement of our system with three input conditions. At this time, we added wind speed as a new input condition. Table 5 shows that after average |error| dropped from 1.25 levels to 1.16 levels, there are plenty of rises in hit rate, from 0.27 to 0.36. Compared with other three experiments, the case-based FCTF reasoning system with four input conditions is the best of these four experiments. Such improvement is also very easy to explain using meteorological knowledge. That is, a larger surface wind can bring a steady stream of moisture from other place, which makes a rain sustainable and precipitation increasing [40].

However, Figure 6a shows that this system often makes mistakes at predicting the rainfall levels, especially for the high magnitude of precipitation, often erroneously forecasted as a low magnitude of precipitation; this is because the entire weather system is very complicated. The purpose of these experiments was to predict precipitation by inputting atmospheric pressure, temperature, relative humidity, and wind force is only to demonstrate that adding input conditions can increase the accuracy of the case-based FCTF reasoning systems. We just added two input conditions in experiment number four compared with experiment number two, and the forecast accuracy improved by 0.36 in hit rate. We believe that, in the practical meteorological services, we can choose more suitable input conditions to improve the accuracy of precipitation forecasts dramatically.

#### 4.4. Analysis of Our System

Assuming that there are  $n$  features from  $X_1$  to  $X_n$ ,  $e_{new}$  is the new coming instance,  $e_{nearest}$  is the closest instance to the new coming one, and we take *Euclidean distance* as the similarity measure method:

$$Dist_{nearest} = \text{sqrt}(\sum iw_i \times (e_{new}(Xi) - e_{nearest}(Xi))^2) \tag{4}$$

Algorithm starts from  $X_1$ ,

$$Dist_{nearest} = \text{sqrt}((e_{new}(X_1) - e_{nearest}(X_1))^2) = |e_{new}(X_1) - e_{nearest}(X_1)| \quad (5)$$

If  $X_1$  is not related to the problem ( $X_1 \notin P_f$ , where  $P_f$  is the set of the problem features), the *hit rate* would be very low regardless whether  $Dist$  is below our threshold. Otherwise, if  $Dist_{nearest} < \alpha$ , Output  $e_{nearest}(Y)$ , And we add another  $X_2$ ,

$$Dist_{nearest} = \text{sqrt}((e_{new}(X_1) - e_{nearest}(X_1))^2 + (e_{new}(X_2) - e_{nearest}(X_2))^2) \quad (6)$$

Analysis: if  $X_i \in P_f$  and  $Dist_{nearest} < \alpha$ , *Hitting rate* would increase at the ideal conditions; if  $X_i \in P_f$  and  $Dist_{nearest} > \alpha$ , *Hitting rate* may increase through increasing the value of  $\alpha$ . If  $X_i \notin P_f$  and  $Dist_{nearest} < \alpha$ ,  $Dist_{nearest}$  is not influenced by adding the noisy attributes. However, the *hit rate* of the system would not increase; If  $X_i \notin P_f$  and  $Dist_{nearest} > \alpha$ ,  $Dist_{nearest}$  is influenced by adding the noisy attributes, and the results would be proved as a failure.

## 5. Conclusions

This paper presents a new reasoning system that can reason new information based on case-based reasoning. Our system is based on a case-based reasoning, but unlike previous case-based reasoning, it starts reasoning with the least number of features. This model is more convenient for some applications when we do not know which feature or features are highly related to the reasoning problem. As mentioned before, Occam's razor is a principle of succinctness used for solving problems. FCTF means that the one with the fewest assumptions or features should be selected. Other more complicated solutions may ultimately prove correct, but if there is less certainty, the fewer assumptions, the better.

We show that the new system is suitable for reasoning. That is, we can reason the solution of a new example based on the previous examples. Most important of all, our experiments show that increasing the number of features would result in more accurate prediction results. However, we have to confess a fact that if a feature is not a key point to a problem, adding the feature to analyze the problem would not result in any improvement of the prediction accuracy.

As we know, case-based learning is important because most real world problems are better characterized as an unknown-feature model. Thus, developing models may better describe real world systems.

Our system is an expansion of the case-based reasoning system, thus, the problem that can be handled by the case-based reasoning system can be solved by our system too. It is suitable when we do not know the features of the problem which we use the case-based reasoning system to deal with, and not suitable when we know the exact influential factors or to any problem that cannot be solved by using case-based reasoning system.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 41275116, 41375121 and 41305079) and the Scientific Research and Innovation Plan for College Graduates of Jiangsu Province, China (No. CXZZ13\_0500) and the Project of Shanxi Meteorological Administration for Leading Talented Person (No.SXKLJRY 20151005).

## Author Contributions

Jing Lu and Xiakun Zhang conceived the study. Jing Lu conducted the experiments and analyzed the results. Jing Lu and Xiakun Zhang wrote the manuscript. Peiren Li and Yu Zhu revised the manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Kolodner, J. *Case-Based Reasoning*, 14rd ed; Elsevier Science: Burlington, MA, USA, 2014; pp. 3–8.
2. Richter, M.M.; Weber, R.O. *Case-Based Reasoning: A Text Book*, 5rd ed; Springer: Heidelberg, Germany, 2013; pp. 5–10.
3. Fischer, B.; Welter, P.; Günther, R. W.; Deserno, T. M. Web-based bone age assessment by content-based image retrieval for case-based reasoning. *Int. J. Comput. Ass. Rad.* **2012**, *7*, 389–399.
4. Recio-García, G.A.; González-Calero, P.A.; Díaz-Agudo, B. Jcolibri2: A framework for building case-based reasoning systems. *Sci. Comput. Program.* **2014**, *79*, 126–145.
5. Bichindaritz, I.; Montani, S. Report on the Eighteenth International Conference on Case-Based Reasoning. *AI Mag.* **2012**, *33*, 79–82.
6. Jo, H.; Han, I.; Lee, H. Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis. *Expert Syst. Appl.* **1997**, *13*, 97–108.
7. Schank, R.C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*, 1st ed; Cambridge University Press: Cambridge, UK, 1983; pp. 1–10.
8. Chattopadhyay, S.; Banerjee, S.; Rabhi, F.A.; Acharya, U.R. A case-based reasoning system for complex medical diagnosis. *Expert Syst.* **2013**, *30*, 12–20.
9. Aha, D.W.; Breslow, L.A.; Muñoz-Avila, H. Conversational case-based reasoning. *Appl. Intell.* **2001**, *14*, 9–32.
10. Lenat, D.B.; Durlach, P.J. Reinforcing math knowledge by immersing students in a simulated learning-by-teaching experience. *Int. J. Artif. Intell. Edu.* **2014**, *24*, 216–250.
11. Qi, J.; Hu, J.; Peng, Y. A new adaptation method based on adaptability under k-nearest neighbors for case adaptation in case-based design. *Expert Syst. Appl.* **2012**, *39*, 6485–6502.
12. Begum, S.; Barua, S.; Filla, R.; Ahmed, M.U. Classification of physiological signals for wheel loader operators using Multi-scale Entropy analysis and case-based reasoning. *Expert Syst. Appl.* **2014**, *41*, 295–305.
13. Chen, X.; Wang, Z.; Sy, C.; Liu, X.; Qian, J.; Zheng, J.; Dong, Z.; Cao, Li.; Geng, X.; Xu, S.; *et al.* Computer-aided diagnosis expert system for cerebrovascular diseases. *Neurol. Res.* **2014**, *36*, 468–474.
14. Ahmed, M.U.; Begum, S.; Funk, P.; Xiong, N.; Schéeleet, B.V. A multi-module case-based biofeedback system for stress treatment. *Artif. Intell. Med.* **2011**, *51*, 107–115.
15. Mansouri, D.; Mille, A. Hamdi-Cherif, A. Adaptive delivery of trainings using ontologies and case-based reasoning. *Arabian J. Sci. Eng.* **2014**, *39*, 1849–1861.

16. Wess, S.; Althoff, K.D.; Richter, M.M. Using kd trees to improve the retrieval step in case-based reasoning. In *Topics in Case-Based Reasoning Lecture Notes in Computer Science*, 1st ed; Wess, S., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; pp. 167–181.
17. Aamodt, A.; Plaza, E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.* **1994**, *7*, 39–59.
18. Li, M.; Tromp, J.; Vitanyi, P. Sharpening Occam’s razor. *Inf. Process. Lett.* **2003**, *85*, 267–274.
19. Floridi, L. A defence of constructionism: Philosophy as conceptual engineering. *Metaphilosophy* **2011**, *42*, 282–304.
20. Gill, Z. The other edge of Occam’s Razor: The A-PR hypothesis and the origin of mind. *Biosemiotics* **2014**, *6*, 403–419.
21. Sinclair, T.R.; Muchow, R.C. Occam’s Razor, radiation-use efficiency, and vapor pressure deficit. *Field Crops Res.* **1999**, *62*, 239–243.
22. Elia, F.; Pagnozzi, B.L.; Aprà, F.; Roccatello, D. A victim of the Occam’s razor. *Int. Emerg. Med.* **2013**, *8*, 767–768.
23. Bontly, T.D. Modified Occam’s Razor: Parsimony, Pragmatics, and the Acquisition of Word Meaning. *Mind Lang.* **2005**, *20*, 288–312.
24. Riesch, H. Simple or simplistic? Scientists’ views on Occam’s Razor. *Theoria* **2010**, *25*, 75–90.
25. Graham, J.; Iyer, R. The unbearable vagueness of “essence”: Forty-four clarification questions for gray, young, and waytz. *Psychol. Inq.* **2012**, *23*, 162–165.
26. Sinnott-Armstrong, W. Does morality have an essence? *Psychol. Inq.* **2012**, *23*, 194–197.
27. Koleva, S.; Haidt, J. Let’s use Einstein’s Safety Razor, not Occam’s Swiss Army Knife or Occam’s Chainsaw. *Psychological. Inq.* **2012**, *23*, 175–178.
28. Geary, T.G.; Bourguinat, C.; Prichard, R.K. Evidence for Macrocyclic Lactone Anthelmintic Resistance in *Dirofilaria immitis*. *Top Companion Anim. Med.* **2011**, *26*, 186–192.
29. Lewowicz, L. Phlogiston, Lavoisier and the purloined referent. *Stud. Hist. Philos. Sci. Part A* **2011**, *42*, 436–444.
30. Mokady, O. Occam’s Razor, invertebrate allorecognition and Ig superfamily evolution. *Res. Immunol.* **1996**, *147*, 241–246.
31. Harta, J. Terence Hutchison and the introduction of Popper’s falsifiability criterion to economics. *J. Econ. Methodol.* **2011**, *18*, 409–426.
32. Charles, A.; Timbal, B.; Fernandez, E.; Hendon, H. Analog downscaling of seasonal rainfall forecasts in the Murray Darling Basin. *Mon. Wea. Rev.* **2013**, *141*, 1099–1117.
33. Rajagopalan, B; Molnar, P. Pacific Ocean sea-surface temperature variability and predictability of rainfall in the early and late parts of the Indian summer monsoon season. *Clim. Dyn.* **2012**, *39*, 1543–1557.
34. Refai, M.H.; Yusof, Y. Partial Rule Match for Filtering Rules in Associative Classification. *J. Comput. Sci.* **2014**, *10*, 570–577.
35. Korena, I.; Feingold, G. Aerosol-cloud-precipitation system as a predator-prey problem. *PNAS* **2011**, *108*, 12227–12232.
36. Rollenbeck, R.; Bendix, J. Rainfall distribution in the Andes of southern Ecuador derived from blending weather radar data and meteorological field observations. *Atmos. Res.* **2011**, *99*, 277–289.

37. Gultepe, I.; Milbrandt, J.A. Probabilistic parameterizations of visibility using observations of rain precipitation rate, relative humidity, and visibility. *J. Appl. Meteor. Climatol.* **2010**, *49*, 36–46.
38. Fay, P.A. Relative effects of precipitation variability and warming on tall grass prairie ecosystem function. *Biogeosciences* **2011**, *8*, 3053–3068.
39. Stahl, C. Seasonal variation in atmospheric relative humidity contributes to explaining seasonal variation in trunk circumference of tropical rain-forest trees in French Guiana. *J. Trop. Ecol.* **2010**, *26*, 393–405.
40. Zhang, W. Differences in teleconnection over the North Pacific and rainfall shift over the USA associated with two types of El Niño during Boreal autumn. *J. Meteorol. Soc. Jpn.* **2012**, *90*, 535–552.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).