

Article

Fuzzy Case-Based Reasoning System

Jing Lu ^{1,2,3}, Dingling Bai ¹, Ning Zhang ¹, Tiantian Yu ⁴ and Xiakun Zhang ^{5,*}

¹ Shanxi Meteorological Administration, Taiyuan 030006, China; lujing19810629@hotmail.com (J.L.); baidingling85@163.com (D.B.); ozhangningo@163.com (N.Z.)

² School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

³ Computer Science Department, Oklahoma State University, Stillwater, OK 74075, USA

⁴ Training Center, Anhui Meteorological Administration, Hefei 230001, China; angelfish0325@hotmail.com

⁵ Weather Forecasting Office, National Meteorological Center of China Meteorological Administration, Beijing 100081, China

* Correspondence: zhangxk@cma.gov.cn; Tel.: +86-10-5899-4184

Academic Editor: José Santamaria

Received: 13 April 2016; Accepted: 9 June 2016; Published: 29 June 2016

Abstract: In this paper, we propose a fuzzy case-based reasoning system, using a case-based reasoning (CBR) system that learns from experience to solve problems. Different from a traditional case-based reasoning system that uses crisp cases, our system works with fuzzy ones. Specifically, we change a crisp case into a fuzzy one by fuzzifying each crisp case element (feature), according to the maximum degree principle. Thus, we add the “vague” concept into a case-based reasoning system. It is these somewhat vague inputs that make the outcomes of the prediction more meaningful and accurate, which illustrates that it is not necessarily helpful when we always create accurate predictive relations through crisp cases. Finally, we prove this and apply this model to practical weather forecasting, and experiments show that using fuzzy cases can make some prediction results more accurate than using crisp cases.

Keywords: case-based reasoning; fuzzy logic; maximum degree principal

1. Introduction

Case-based reasoning systems [1–3] are important in the machine learning field. They have been broadly used in the process of solving new problems based on the solutions of similar past problems. Case-based reasoning is a prominent kind of analogy making [4–6].

A fuzzy concept is widely understood by scientists as a concept for which the boundaries of application vary according to specific context or conditions, instead of being fixed [7,8]. This means the concept is vague in some way, lacking a fixed, precise meaning [9,10]. Some scientists have claimed that there is no fuzzy concept in the world. For example, Rudolf E. Kálmán stated in 1972 that fuzzy concepts did not exist since they were not scientific concepts [11,12]. This suggestion was based on an idea that a scientific concept should be clear and precise. However, there is no general agreement on how a scientific concept should be defined, and scientists quite often use imprecise analogies in their models to help understand an issue.

Over the last decade, a great deal of attention has been devoted to the use of case-based reasoning systems [13–15]. Scientists sometimes found that the exact inputs of a case-based reasoning system for predictions lead to inaccuracy in the prediction outputs [16–18]. In view of this, we consider the use of fuzzy cases as a point forecast for load. Traditionally, a case-based reasoning system used crisp cases as its inputs to output crisp results. In this paper, we use fuzzy cases as its inputs and predict fuzzy outputs instead. Our fuzzy case-based reasoning system could produce a fuzzy output according to

the fuzzy inputs by using the algorithm we proposed in this paper. We use this approach in weather forecasting as a test, and a proof is given.

The organization of this paper is as follows. In Section 2, we briefly introduce the preliminaries of the paper: case-based reasoning and fuzzy logic theory. Section 3 presents how to use fuzzy case-based reasoning methods to express fuzzy sets and use them in prediction problem. Section 4 is the discussion and the experiments. Finally, the conclusion (Section 5) is given.

2. Preliminaries

2.1. Case-Based Reasoning System

Machine learning is a field of study that gives computers the ability to learn from and make predictions on data without being explicitly programmed. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions expressed as outputs, rather than following strictly static program instructions. Case-based reasoning, as one of the most important machine learning algorithms, has been widely studied. A case-based reasoner uses old experiences or cases to suggest solutions to new problems, to point out potential problems, to interpret a new situation and make predictions on what might happen, or to create arguments to measure the conclusion [19,20]. Among machine learning algorithms, case-based reasoning (CBR) has a higher flexibility and requires less maintenance effort. Also, it could improve problem solving performance through reuse, improve over time and adapt to changes in the environment as well. Most important of all, it has a higher user acceptance. In total, CBR is a worthy topic to research. The CBR working process is shown in Figure 1.

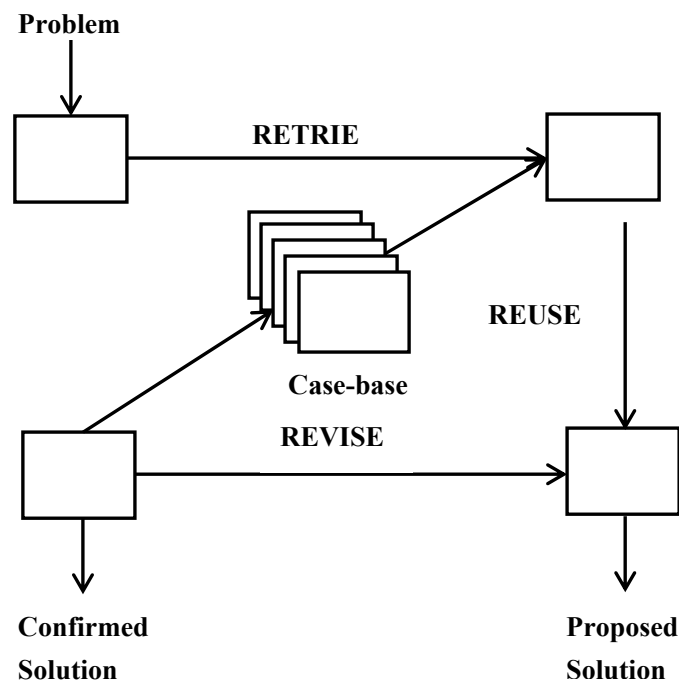


Figure 1. Case-based learning process.

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.

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

Specification on Retrieve and Reuse: Generally speaking, a case-based reasoner solves new problems by recording old cases and adapting the most similar one [21,22]. Specifically, a case is composed of several premises and a conclusion; if the premises of an old case in the case base (used to

store the previous cases) are the most similar to the premises of the new coming case, the conclusion of this old case will be chosen as the conclusion of this new coming case; the similarity of the premises of the cases is usually calculated using a *Euclidean metric* specified as follows.

Suppose $c(X_i)$ (shown in Table 1) is the value of feature X_i for the case c ; $(c_1(X_i) - c_2(X_i))$ is the feature difference between c_1 and c_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. 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 a problem is easier than the first because we are more competent and we remember our mistakes and therefore can avoid them.

Revise: Evaluate and revise the solution based on how well it works. The revision may 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.

Table 1. Values of feature X_i for the case c_1 and c_2 .

Features	X_1	X_2	X_3	X_4	X_5	X_i	X_n
c_1	$c_1(X_1)$	$c_1(X_2)$	$c_1(X_3)$	$c_1(X_4)$	$c_1(X_5)$	$c_1(X_i)$	$c_1(X_n)$
c_2	$c_2(X_1)$	$c_2(X_2)$	$c_2(X_3)$	$c_2(X_4)$	$c_2(X_5)$	$c_2(X_i)$	$c_2(X_n)$

2.2. Fuzzy Logic-Membership Function

We introduce fuzzy logic to our paper to illustrate that it is not always good when we unlimitedly describe the world more and more accurately. A case based reasoning system or other machine learning methods have the ability to learn from and make predictions about data without being explicitly programmed. “Data” here means crisp data and using these data to learn and make predictions sometimes leads to unsatisfactory results. To improve their performance we introduce fuzzy logic theory, where the membership function is a key concept.

In crisp set theory, the membership of a value in a set is a binary value; that is, a value either is a member of the set or it is not. Fuzzy set theory [23,24] introduces the idea that a value can be partially in the set and partially outside the set, simultaneously. This is accomplished by defining a membership function that describes the degree of membership [25,26] in the set for each value.

Both crisp set and fuzzy set can be represented by a membership function. Membership functions can take many forms, but there are some common examples that appear in real applications. A membership function can be chosen by the user according to their experience or can be decided by machine learning methods, and it can be represented as curves, where the x -axis represents values from the universe and the y -axis represents the membership degree. A membership function of a crisp set should be like a clock signal since its y -axis values are 0 or 1; a membership function of a fuzzy set may be triangular, trapezoidal, and so on.

3. Fuzzy Case-Based Reasoning System

We firstly give an overview of a fuzzy case-based reasoning system in Section 3.1. The first and most important step for a fuzzy case-based reasoning system is to transform the crisp cases to fuzzy ones. In Section 3.2, we mainly focus on this step. In Section 3.3, we give a total description of the working procedure of this system; in Section 3.4, we prove the correctness of this system.

3.1. Introduction

We may notice that fuzzy case-based reasoning system is a kind of fuzzy expert system, but differs from them significantly. Previous fuzzy expert systems make use of “fuzzy inference”, formulating the

mapping from a given crisp input to a crisp output using fuzzy logic. The process of fuzzy inference involves all of the pieces that are described in Membership Functions, Logical Operations, and If-Then Rules. This section describes the steps of a fuzzy expert system and uses the example of the two-input, one-output, three-rule problem. The basic structure of this example is shown in the following diagram, Figure 2a. From Figure 2a, we see that the inputs of a fuzzy expert system are crisp and the outputs are also crisp; in intermediate layers, fuzzy logic is used for predicting the outcomes given an input set based on the background knowledge—fuzzy rules and the membership functions. Our fuzzy case-based reasoning system is shown in Figure 2b. From Figure 2b, we first transfer the crisp inputs into fuzzy ones. Assume input 1 (Atmosphere Pressure) is “Lower” and input 2 (Wind Force) is “Breeze”, according to the case base created in advance, then the output (Precipitation) is “Heavy”, which is a fuzzy output.

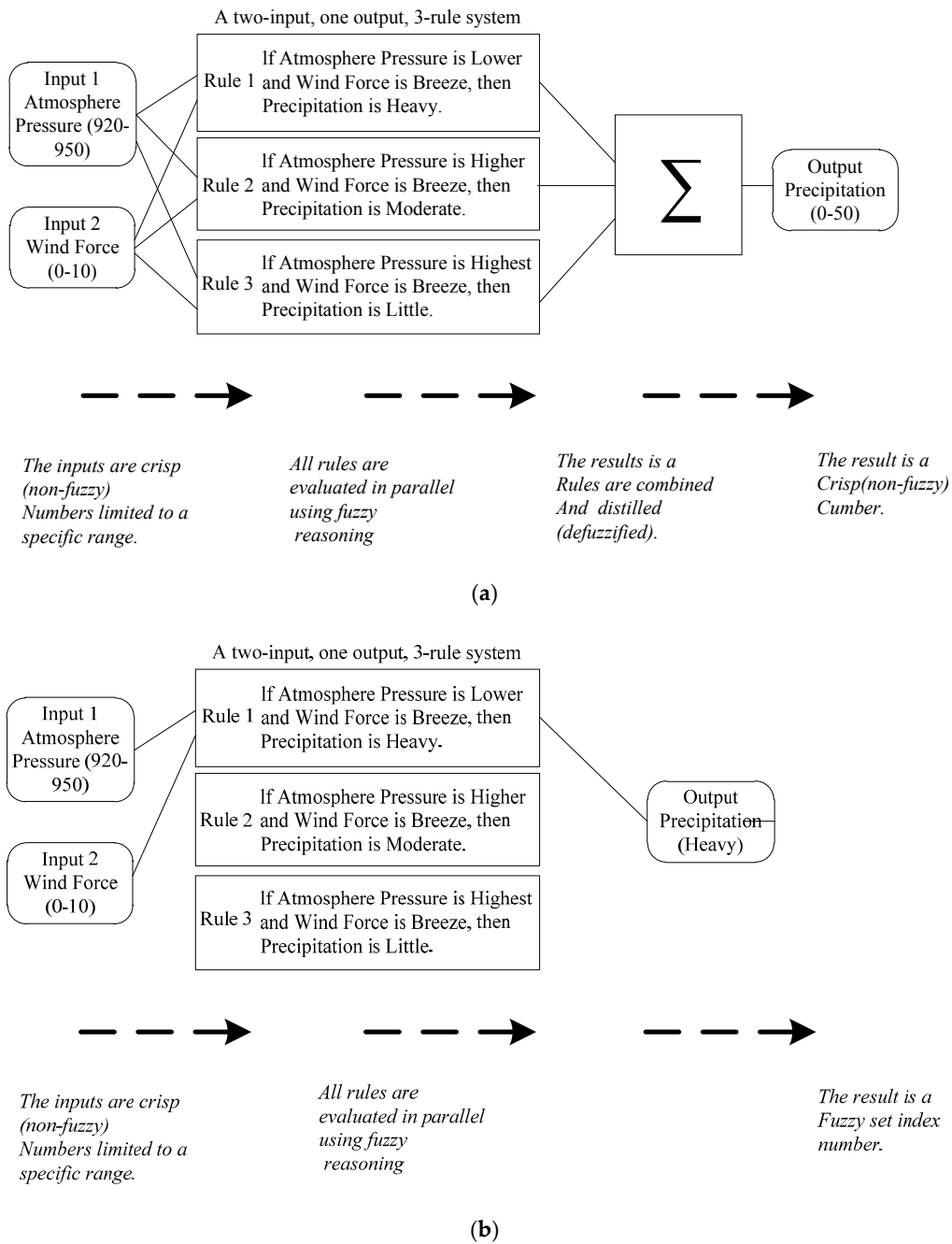


Figure 2. (a) Fuzzy expert system. (b) Fuzzy case-based reasoning system.

The most important reason for changing crisp cases into fuzzy ones is that the prediction results may get better with the fuzzification process. It is well known that the need for accurate prediction is apparent when considering the benefits that it has. However, experiments show that the accurate crisp patterns make some accurate prediction results useless and meaningless. We also add the “vague” concept into the pattern reduction problem. It is somewhat vague inputs that make the outcomes of prediction more meaningful and accurate.

Some would say that this better prediction accuracy might be achieved by sacrificing the output precision, and sometimes that is the case. The disadvantage of this system is that it can increase the prediction accuracy, but must sacrifice the precision.

Sometimes, however, this is not the case. Experiments and the proof show that with the output precision unchanged (only change the inputs fuzzification), the prediction accuracy can still be increased. These phenomena can be intuitively understood because fuzzy ones may solve a problem better than a more accurate one. Sometimes, several elements together take effect, and if we divide them then we will not be able to get desired results.

3.2. Transfer Crisp Cases to Fuzzy Ones

Susan Haack once claimed that a many-valued logic requires neither intermediate terms between true and false, nor a rejection of bivalence. Her suggestion was that the intermediate terms (i.e., the gradations of truth) can always be restated as conditional if-then statements, and by implication, that fuzzy logic is fully reducible to binary true-or-false logic. This interpretation of a long sequence of if-then statements is often enormously less efficient than membership functions, although they have the same meaning to represent a fuzzy set [27–31]. However, this point is obviously of great importance for computer programmers.

In our paper, we change Susan Haack’s idea to represent a fuzzy set by a long sequence of if-then statements. Specifically, an interval is the “if” part of a “if-then” statement, and the positive integer is the “then part” of it. An example is given in Figure 3. Usually, different fuzzy sets are represented by different membership functions shown in Figure 3, for example shown as y_1 and y_2 , correspondingly match with “Small” and “Large”. In real ideal circumstance, there is no overlapping between the original domain(x), but sometimes in real world, y_1 and y_2 share the same domain (shown in Figure 3), and given a crisp number c that partially belongs to Small and partially belongs to Large, which fuzzy set should c belong to? According to the maximum degree principle, c belongs to the fuzzy set Small since $y_2(c) > y_1(c)$. Thus, we fuzzify one premise of a case by interpreting an interval as a number. If one premise of a case falls within an interval, we set it to a numerical value. From the Figure 3, if a premise is between 0 and a , $y_2(c) > y_1(c)$, we set the interval value to Number_1; if the premise is between a and b , $y_2(c) > y_1(c)$, we set the interval value to Number_2. Now we fuzzify a case by fuzzifying each compositive element (premise) using the interval transfer method mentioned above. Thus, each element is transferred to a fuzzy set denoted by a crisp positive integer.

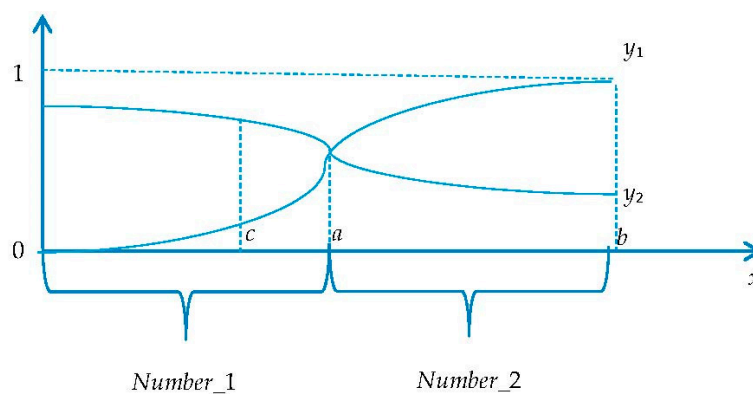


Figure 3. Interval transfer.

In real circumstance, a case in case-based reasoning is complicated and can be represented in various forms. Traditional approaches can be classified into three main categories: feature vector representations, structured representations, and textual representations. More sophisticated approaches make use of hierarchical representations. For particular tasks such as designing and planning, highly specific representations have been developed. Developments have been made in the process of refining in case-based reasoning as shown above. However, in this paper, we take the opposite direction and reduce the complexity of cases and their total number (different case elements may be transferred to the same number and correspondingly previous different cases are changed to the same). Most important of all, the prediction results may get better with the fuzzification process.

3.3. Fuzzy Case-Based Algorithm

The fuzzy case-based algorithm is the algorithm executed in the fuzzy case-based reasoning system and symbols used are shown in Table 2.

/ we assume that there are r fuzzy cases (rules) in the base with the same structure and each case is composed of n fuzzy premises (features) in each case and one fuzzy output.*/*

Table 2. Symbols used in fuzzy case-based reasoning algorithm.

Symbols	Meanings
$j; i$	the index of cases; the index of features
n	the number of features
R	the number of cases
$PreCount[]$	array to count the number of new coming case's premises that are equal to the R_j 's
$Countj$	index to test if the new coming case existed in the case base
$c_{new}; p; o; b$	c_{new} is the new coming case; p presents the premises of the new coming case; o is the outcome of the new coming case; b is the case's index number when the premises of the new coming rule is the most similar to its premises

Stage 1: In this stage, crisp numbers are mapped to corresponding fuzzy sets. We divided these fuzzy sets according to different intervals.

Stage 2:

```

for j = 1 to r
    PreCount[] = 0; // set PreCount[] to zero
    for i = 1 to n
        if ( $c_{new} \cdot p == c_{ji}$ )
            PreCount[j] ++ ;
        end
    end
    if (PreCount[j] == n) // PreCount[j] == n when the j th fuzzy rule's premises are the same to
    the new coming rule
         $c_{new}.o = c_{j(n+1)}$ ;
         $c_{new}.push()$ ;
    end
end
Countj = 0
if k = 1 to r
    if (PreCount[j] != 0)
        Countj = 1;
        break;
    end
if (Countj == 0) // Countj == 0 means the new coming case(rule) is a new one.

```

```

CasebasedReasoning();
/* "CasebasedReasoning()" is a function that could doing the case-based reasoning. */
else  $c_{new,0} = c_{b(n+1)}$ 
     $c_{new}$ .push();
end

```

Stage 3: We use CasebasedReasoning() function to simulate the inference process.

Stage 4: We return to Stage 1 but keep the original fuzzy classification on fuzzy outputs unchanged. Thus, we maintain the precision and show a change in prediction accuracy with the change in fuzzy classification of fuzzy inputs.

/ "restricted domain" means the reasonable range of the output. If we hope to restrict the range of the output to {1, 2, 3, 4, 5}, the restricted domain is {1, 2, 3, 4, 5}. β is a threshold to control the similarity degree of the output and the element of the restricted domain.**

Definition 3.2: let *Correctp* be number of correct predictions and let overall number of predictions be *Totalp*. The hitting rate *Hitrate* is defined as:

$$Hitrate = Correctp/Totalp$$

3.4. Analysis and Proof

The fuzzification process of case elements can be understood as the process of merging on adjacent intervals.

Assume that there are two fuzzy cases with only one input and one output: c_1 and c_2 are crisp numbers of two adjacent intervals, or in other words, the index numbers of two adjacent fuzzy sets; *high_pro*{ e_1, e_2, \dots, e_n } means that the events to occur with a high possibility given a specific input, for example "1" or "2"; te_1, te_2, \dots, te_n is the occurring times of events e_1 to e_n in the case base given the same input.

case(1) and case(2) can be expressed as follows:

case(1): if $x_1 = c_1, y_1 = high_pro\{e_1, e_2, \dots, e_n\}_1$

case(2): if $x_1 = c_2, y_2 = high_pro\{e_1, e_2, \dots, e_n\}_2$.

If

$max\{te_1, te_2, \dots, te_n\}_1 = te_i$ and $max\{te_1, te_2, \dots, te_n\}_2 = te_j$,

then $high_pro\{e_1, e_2, \dots, e_n\}_1 = e_i$ and $high_pro\{e_1, e_2, \dots, e_n\}_2 = e_j$.

Before merge,

case(1): if $x_1 = c_1, y_1 = high_pro\{e_1, e_2, \dots, e_n\}_1 = e_i$

case(2): if $x_1 = c_2, y_2 = high_pro\{e_1, e_2, \dots, e_n\}_2 = e_j$.

- (1) if case(1) and case(2) are fit for the new coming case(the case-based reasoning system can predict the correct value for the new coming case), the predicted value and the expected value should be:

$c_1 \rightarrow e_i$;

$c_2 \rightarrow e_j$

After merging,

$c_{12} \rightarrow e_i$ or e_j

prediction accuracy would decrease, except $e_i = e_j$ since the prediction accuracy won't change in this case.

- (2) if case(1) and case(2) are not fit for the new coming rule(the case-based reasoning system cannot predict the correct value for the new coming case), assume the expected value is shown as below:

$$c_1 \rightarrow e_w$$

$$c_2 \rightarrow e_p$$

After merging,

$$c_{12} \rightarrow high_pro\{e_1, e_2, \dots, e_n\}_{12}$$

$$max\{te_1, te_2, \dots, te_n\}_{12} = max\{(te_{11} + te_{12}), \dots, (te_{n1} + te_{n2})\}$$

if $max\{(te_{11} + te_{12}), \dots, (te_{n1} + te_{n2})\} = (te_{w1} + te_{w2})$, then $high_pro\{e_1, e_2, \dots, e_n\}_{12} = e_w$, the prediction accuracy would increase after merge;

if $max\{(te_{11} + te_{12}), \dots, (te_{n1} + te_{n2})\} = (te_{p1} + te_{p2})$, then $high_pro\{e_1, e_2, \dots, e_n\}_{12} = e_p$, the prediction accuracy would increase after merge;

if $e_w = e_p$, the prediction accuracy would also increase after merge.

4. Experiments and Discussion

Data collection: Weather data (May 2010–October 2010, May 2011–October 2011, May 2012–October 2012, May 2014–October 2014) were collected from the meteorological department of Shanxi, China, Asia. The day’s wind speed, dry wet bulb temperature, relative humidity, and pressure at 20 o’clock are taken as inputs while the cumulative precipitation quantity for the next 24 h is taken as outputs for our weather prediction model.

Atmospheric pressure falls into four fuzzy sets in Table 3; relative humidity falls into four fuzzy sets shown in Table 4; dry and wet bulb temperature falls into six fuzzy sets shown in Table 5; wind force falls into 5 levels (fuzzy sets) shown in Table 6; precipitation falls into five fuzzy sets shown in Table 7. In order to test, numbers from 0 to 4 inclusively is given to represent Light rain, Moderate rain, Heavy rain, Rainstorm and Heavy rainstorm respectively. In Section 4.2, we keep the original fuzzy classification of fuzzy outputs unchanged. Thus, we maintain the precision and show a change in prediction accuracy with the change in fuzzy classification of fuzzy inputs.

Working sample: We give a simple example to show how to work with a fuzzy case-based reasoning system. Assume the original case base contains four cases with four inputs and one output as shown below:

927.1	12.	35	0.9	0	(old case 1)
919.7	18.5	30	2	0	(old case 2)
920.3	20.6	49	1.4	1.2	(old case 3)
917	17.1	75	1.8	0	(old case 4)
915.9	18.0	37	2.6	?	(New coming case)

and we check the Tables 3–7 to fuzzify them into

3	2	2	2	0	(old case 1)
1	2	1	3	0	(old case 2)
3	2	2	2	0	(old case 3)
1	2	4	3	1	(old case 4)
1	2	2	3	?	(New coming case)

Then, the system finds the nearest case is number 2, and generates output according to it, which should be “0”.

Table 3. Fuzzy classification of atmospheric pressure.

Atmospheric Pressure Rating	Moderate	Slightly Lower	Lower	Lowest
hPa	≥940	[930, 940)	[920, 930)	<920

Table 4. Fuzzy classification of dry and wet bulb temperature.

Dry and Wet Bulb Temperature Rating	Lowest	Lower	Slightly Lower	Moderate	Higher	Highest
°C	<−10	[−10, 5)	[5, 20)	[20, 30)	[30, 45)	≥45

Table 5. Fuzzy classification of relative humidity.

Relative Humidity Rating	Dry	Less Dry	Less Humid	Humid
Humidity value(%)	[0, 30)	[30, 50)	[50, 70)	[70, 100]

Table 6. Fuzzy classification of wind force. MPH: Miles per hour.

Wind Force Rating	Calm	Light Air	Light Breeze	Gentle Breeze	Moderate Breeze
MPH	(0, 2)	[2, 4)	[4, 7)	[7, 11)	[11, 17)

Table 7. Fuzzy classification of precipitation.

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

4.1. Comparison between a Fuzzy Case-Based Reasoning System and Other Prediction Models

We conducted experiments to compare a fuzzy case-based reasoning system with a fuzzy expert system and case-based reasoning system.

According to Tables 3–7 above, we make up the membership functions for these four inputs and one output, shown in Figure 4. Fuzzy rules are created based on previous old crisp cases through the fuzzification process.

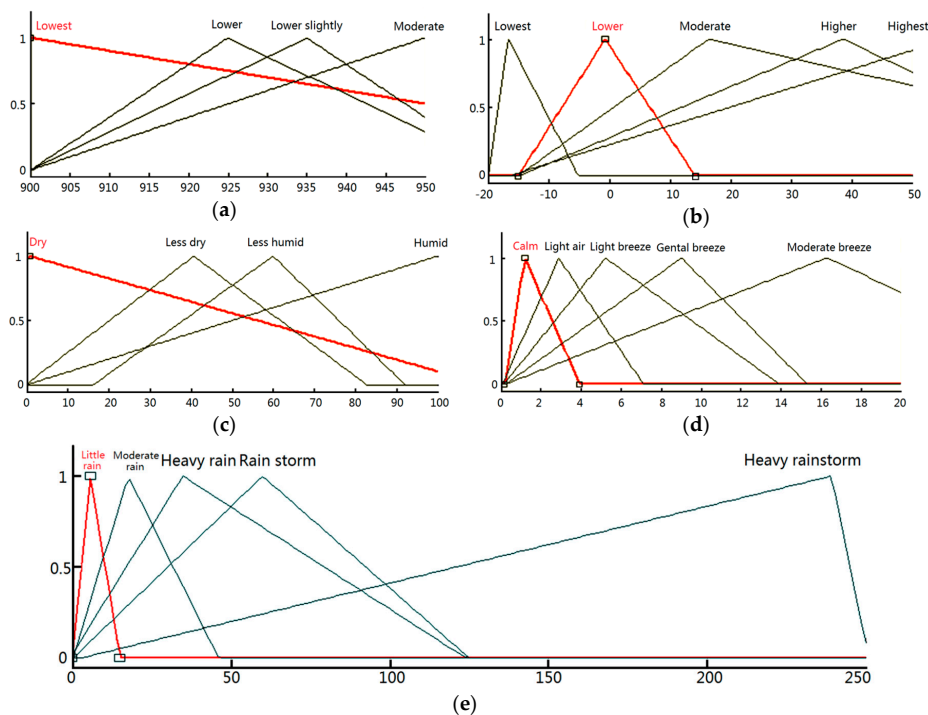


Figure 4. Membership function curves for input variables (a) atmospheric pressure, (b) temperature, (c) relative humidity, (d) wind force and (e) output variable precipitation.

Both CBR and FCBR (Fuzzy case-based reasoning system) are machine learning prediction methods: CBR works with crisp cases and FCBR works with fuzzy ones. The fuzzy expert system differs from the two above largely since it is not a machine learning prediction method but a mathematical one. Experimental results (Figures 5 and 6) show that FCBR has a higher hitting rate than its relatives CBR and the fuzzy expert system, and the fuzzy expert system shows its worst performance among these three. The main reason is shown below.

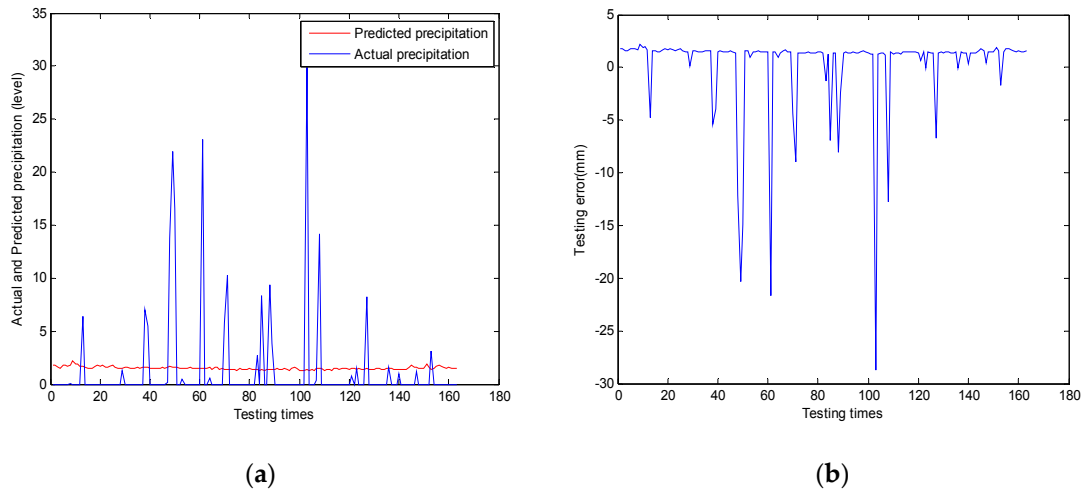


Figure 5. (a) Testing results of case-based reasoning (CBR). (b) Testing error of CBR.

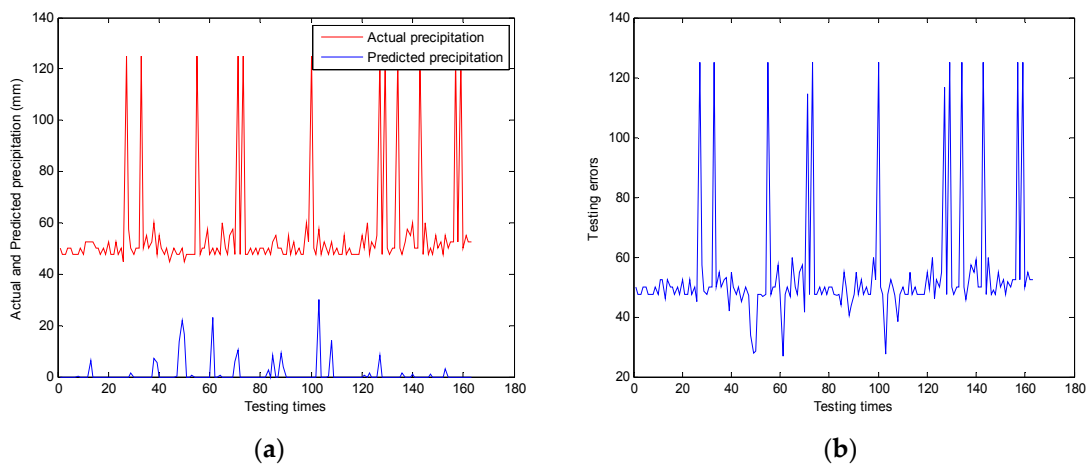


Figure 6. (a) Testing results of fuzzy expert system. (b) Testing error of fuzzy expert system.

Practical weather prediction is very complicated and the fuzzy expert system is a comparatively simpler and linear mathematical prediction model, so it cannot predict future states very well. To solve this, a lot of work has been put into improving the fuzzy expert system, such as combining it with neural networks since machine learning models are non-linear and can predict complex relationships.

4.2. Changes in Prediction with Changes in Fuzzy Classification

This section is meant to specify that the increment of prediction accuracy may or may not be achieved by sacrificing the prediction precision. Correspondingly, the advantage of this system is that it can increase prediction accuracy without sacrificing precision in an ideal situation, and the disadvantage is that it may sometimes sacrifice precision. From CBR to FCBR(1), there are both

increment and sacrifice; from FCBR(1) to FCBR(2), there is no increment and no sacrifice; from FCBR(1) to FCBR(2), there is increment and no sacrifice.

Analysis: From the context above, we change Susan Haack's idea to represent fuzzy sets as "if-then" statements. In other words, an interval can be expressed as a fuzzy set. Because of this, different intervals match with different fuzzy set and fuzzification process can be treated as the process of interval merge.

FCBR starts with the original fuzzy classification and experiment shows results in Table 8 as FCBR(1) and in Figure 7. Then, we change fuzzy classification of atmosphere pressure from Moderate, Slightly Lower, Lower and Lowest into three classes: Moderate, Lower and Lowest. In other words, we merge Moderate and Lower slightly, which show in Table 9. Next, we merge dry and wet bulb temperature Lower and Slightly Lower into Lower according to Table 10. We clearly see that merge Slightly Lower and Lower does not influence the prediction results since Slightly Lower and Lower have the same effects on the prediction results: both of them are sensitive or non-sensitive to the precipitation. However, when we change dry and wet bulb temperature, the hitting rate is increased since one of them is sensitive and the other is non-sensitive and through merging the effect of non-sensitivity is reduced.

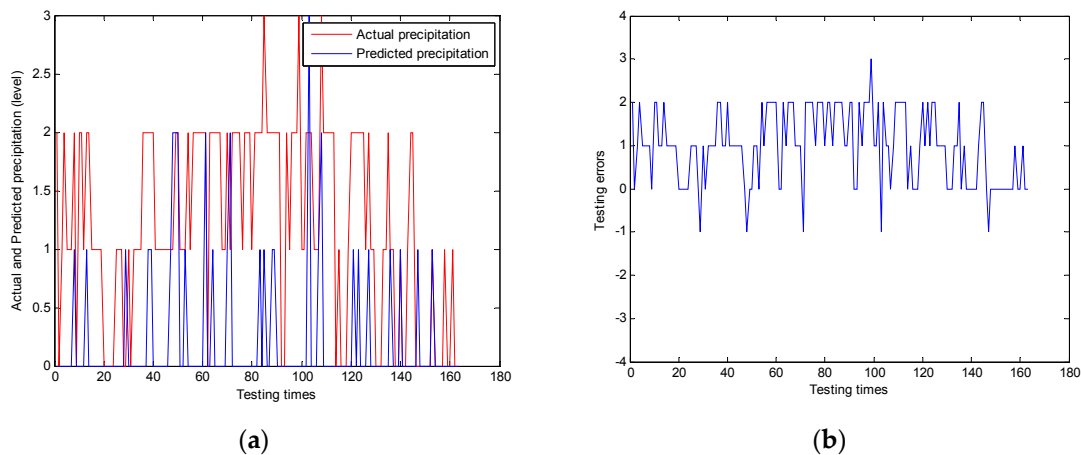


Figure 7. (a) Testing results of FCBR (Fuzzy case-based reasoning system)(1). (b) Testing error of FCBR(1).

Error Analysis

Data from Figures 7–9 and Table 8 indicates:

1. **CBR** methods are used to predict the exact accumulative precipitation for weather forecasting, but lead to the inaccuracy of the prediction outputs. Hitting rate is 0, which means that the rate of prediction accuracy is 0. From Figure 7, we see that predicted results cannot reflect the change of actual precipitation.
2. **FCBR(1)** is our fuzzified CBR system under our first classification method, which is not good enough to predict the fuzzy output but still gets an 0.28 increment on hitting rate.
3. **FCBR(2)** there is no change in results after a merging of atmospheric pressure classification. This means that increasing the degree of fuzzification may do not work.
4. **FCBR(3)** Table 8 shows that the FCBR(3) is a relative stable model since the errors fall between an interval between -3 to 3 level and its hitting rate and the average error are the best of the four methods.

Table 8. Comparison between case-based reasoning (CBR) and fuzzy case-based reasoning system.

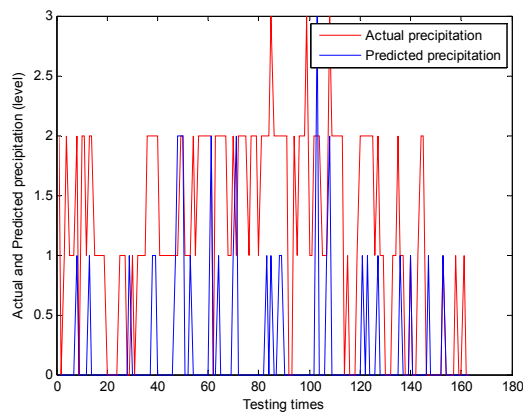
Experimental Results	Minimum Error	Maximum Error	Minimum Error	Maximum Error	Average Error	Hit Rate
CBR	−2.1 mm	−28.1 mm	0.094 mm	28.1 mm	2.49 mm	0
FCBR(1)	−1 level	3 level	0 level	3 level	1.04 level	0.28
FCBR(2)	−1 level	3 level	0 level	3 level	1.04 level	0.28
FCBR(3)	−3 level	2 level	0 level	3 level	0.57 level	0.57
Fuzzy expert system	26.9 mm	125 mm	26.9 mm	125 mm	54.6 mm	0

Table 9. Fuzzy classification of atmospheric pressure.

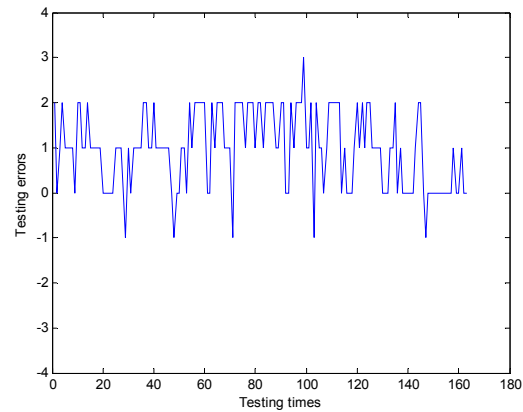
Atmospheric Pressure Rating	Moderate	Lower	Lowest
hPa	≥940	[920, 940)	<920

Table 10. Fuzzy classification of dry and wet bulb temperature.

Dry and Wet Bulb Temperature Rating	Lowest	Lower	Moderate	Higher	Highest
°C	<−10	[−10, 5)	[5, 30)	[30, 45)	>45

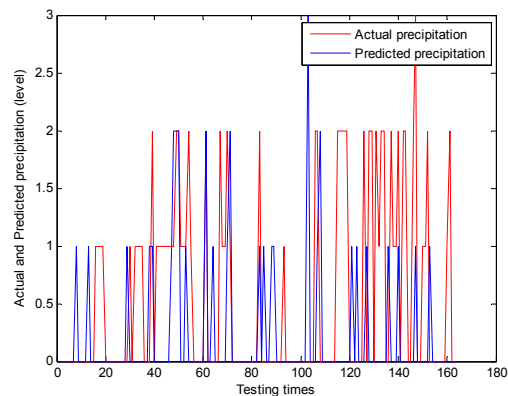


(a)

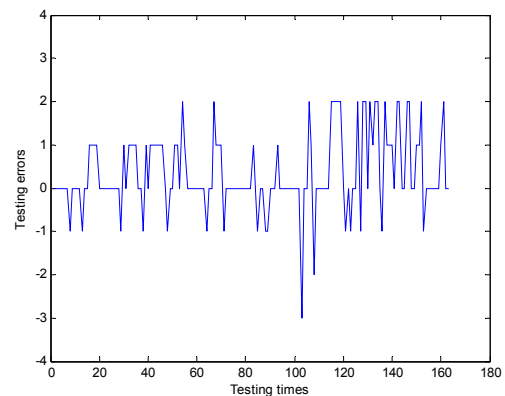


(b)

Figure 8. (a) Testing results of FCBR(2). (b) Testing error of FCBR(2).



(a)



(b)

Figure 9. (a) Testing results of FCBR(3). (b) Testing error of FCBR(3).

5. Conclusions

A case-based reasoning system is a kind of supervised learning: The computer is presented with example inputs and desired outputs (given by a “teacher”), and the goal is to learn a general rule that maps inputs to outputs. When we use case-based reasoning system to predict future conditions, the first task is to record the old cases, which we call the learning process. Although we do not know the exact relationship, such as “ $y = ax + b$ ”, we still can get prediction results according to the old cases, in other words, the maps, after the process of training. In our paper, we change the crisp inputs of the case-based reasoning system into fuzzy ones and establish a proof to show the correctness of this method. Finally, we make use of this fuzzy case-based reasoning system to predict weather conditions and get a better result than the traditional case-based reasoning.

A case-based reasoning system can reduce the prediction error, increase the hitting rate greatly, and decrease complexity and running time. As we know, weather forecasting methods are often comprehensive since a single method or model may not be able to get good results, but only our model could reduce the prediction error by a large amount. Not only may our models combine other prediction methods together to produce better prediction results, but they can also be used in forecasting other meteorological elements. However, space is lacking for a detailed description of it.

Acknowledgments: This work was supported by the Shanxi Province Overseas Students Management Office Foundation (No. 2015-121).

Author Contributions: Jing Lu and Xiakun Zhang conceived the study; Jing Lu and Dingling Bai conducted the experiments and analyzed the results; Jing Lu wrote the manuscript; Ning Zhang and Tiantian Yu helped revise the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Richter, M.M.; Weber, R.O. *Case-Based Reasoning: A Text Book*, 5rd ed.; Springer: Heidelberg, Germany, 2013; pp. 5–10.
2. Kolodner, J. *Case-Based Reasoning*, 14rd ed.; Elsevier Science: Burlington, MA, USA, 2014; pp. 3–8.
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. Assist. Radiol. Surg.* **2012**, *7*, 389–399. [[CrossRef](#)] [[PubMed](#)]
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. [[CrossRef](#)]
5. Bichindaritz, I.; Montani, S. Report on the eighteenth international conference on case-based reasoning. *AI Mag.* **2012**, *33*, 79–82.
6. 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. [[CrossRef](#)]
7. Stotts, L.; Kleiner, B.H. New developments in fuzzy logic computers. *Ind. Manag. Data Syst.* **1995**, *95*, 13–17. [[CrossRef](#)]
8. Roman-Flores, H.; Barros, L.C.; Bassanezi, R.C. A Note on Zadeh’s Extensions. *Fuzzy Sets Syst.* **2001**, *117*, 327–331. [[CrossRef](#)]
9. Andrew, A.M. Artificial neural nets and BCL. *Kybernetes* **2005**, *34*, 33–39. [[CrossRef](#)]
10. González, J.M.B.; DesJardins, S.L. Artificial neural networks: A new approach to predicting application behavior. *Res. High. Educ.* **2002**, *43*, 235–258. [[CrossRef](#)]
11. Li, S.G.; Wu, Z. Business performance forecasting of convenience store based on enhanced fuzzy neural network. *Neural Comput. Appl.* **2008**, *17*, 569–578. [[CrossRef](#)]
12. Rajasekaran, S.; Febin, M.F.; Ramasamy, J.V. Artificial fuzzy neural network in civil engineering. *Comput. Struct.* **1996**, *61*, 291–302. [[CrossRef](#)]
13. 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. [[CrossRef](#)]
14. 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. [[CrossRef](#)]

15. Chen, X.; Wang, Z.; Sy, C.; Liu, X.; Qian, J.; Zheng, J.; Dong, Z.; Cao, L.; Geng, X.; Xu, S.; et al. Computer-aided diagnosis expert system for cerebrovascular diseases. *Neurol. Res.* **2014**, *36*, 468–474. [[CrossRef](#)] [[PubMed](#)]
16. Lu, J.; Zhang, X.; Li, P.; Zhu, Y. Case-based FCTF reasoning system. *Appl. Sci.* **2015**, *5*, 825–839. [[CrossRef](#)]
17. Lu, J.; Xue, S.; Zhang, X.; Zhang, S.; Lu, W.; Zhu, Y. Neural fuzzy inference system-based weather prediction model and its precipitation predicting experiment. *Atmosphere* **2014**, *5*, 788–805. [[CrossRef](#)]
18. Lu, J. A neural fuzzy inference system. *J. Electron. China* **2013**, *30*, 401–410. [[CrossRef](#)]
19. 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. [[CrossRef](#)] [[PubMed](#)]
20. 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. [[CrossRef](#)]
21. Aamodt, A.; Plaza, E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.* **1994**, *7*, 39–59.
22. Jo, H.; Han, I.; Lee, H. Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis. *Expert Syst. Appl.* **1997**, *13*, 97–108. [[CrossRef](#)]
23. Matbouli, Y.T.; Hipel, K.W.; Kilgour, D.M.; Karray, F. A fuzzy logic approach to assess, manage, and communicate carcinogenic risk. *Hum. Ecol. Risk Assess.* **2014**, *20*, 1687–1707. [[CrossRef](#)]
24. Precup, R.E.; Tomescu, M.L.; Dragos, C.A. Stabilization of Rössler chaotic dynamical system using fuzzy logic control algorithm. *Int. J. Gen. Syst.* **2014**, *43*, 413–433. [[CrossRef](#)]
25. Kissi, M.; Ramdani, M.; Tollabi, M.; Zakarya, D. Determination of fuzzy logic membership functions using genetic algorithms: Application to structure-odor modeling. *J. Mol. Model.* **2004**, *10*, 335–341. [[CrossRef](#)] [[PubMed](#)]
26. Grauel, A.; Ludwig, L.A. Construction of differentiable membership functions. *Fuzzy Sets Syst.* **1999**, *101*, 219–225. [[CrossRef](#)]
27. Loetamonphong, J.; Fang, S.C.; Young, R.E. Multi-objective optimization problems with fuzzy relation equation constraints. *Fuzzy Sets Syst.* **2002**, *127*, 141–164. [[CrossRef](#)]
28. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning—I. *Inf. Sci.* **1975**, *8*, 199–249. [[CrossRef](#)]
29. Potts, L.V.; Miima, J.B.; Acar, M. Course evaluation using fuzzy linguistic rules. *Surv. Land Inf. Sci.* **2012**, *72*, 119–128.
30. Ghatee, M.; Hashemi, S.M. Application of fuzzy minimum cost flow problems to network design under uncertainty. *Fuzzy Sets Syst.* **2009**, *160*, 3263–3289. [[CrossRef](#)]
31. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *SMC-15*, 116–132. [[CrossRef](#)]



© 2016 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 (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).