FuzzyRULES-II: A new approach to fuzzy rule induction from numerical data

Ashraf A. Afify*
Industrial Engineering Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt

Abstract. Data mining is a broad area that integrates research efforts from several fields with the aim of processing large volumes of data into knowledge bases for better decision making. Since numerical and nominal data are equally important in practical data mining applications, dealing with different types of data items are among the most important problems in data mining research and development. This paper introduces a new fuzzy rule induction algorithm, able to deal properly with either numerical or nominal attributes, for the creation of classification and predictive models. To better handle numerical data, fuzzy sets are used to represent intervals in the domains of numerical attributes. Experimental results have shown that the proposed algorithm produces robust and general models that can be used for prediction as well as for classification.

Keywords: data mining, rule induction, numerical output prediction, discretisation, fuzzy sets

1. Introduction

Recent advances in information technology have made it easy to collect and store enormous amounts of data. The new research field of data mining [1] offers new intelligent tools for discovering knowledge from such data. There are many different types of data mining algorithms. These algorithms allow the creation of different models that describe the patterns found in the data. The obtained models can then be used to clarify these particular patterns or as predictive models.

This paper focuses on a particular type of data mining algorithms, namely, rule induction algorithms. An important feature of rule induction algorithms is that their model structure (in the form of if-then rules) is readily understood by humans. Because of this structure, rule induction became popular for classification problems. However, rule induction algorithms are still weak at handling numerical values and in particular when dealing with numerical outputs. Other learning methods, such as regression and neural networks, are widely used to develop models for this type of application. However, their more complex mathematical structure makes them difficult to interpret, whilst in many applications experts might need to be able easily to verify a decision given by a model.

Few fuzzy rule induction methods have been developed for handling attributes with numerical values. Reference [2] proposed a fuzzy rule induction method that can generate fuzzy rules from numerical data. However, this method is computationally expensive and creates complex fuzzy rule sets. Reference [3] introduced another method called RULES-F Plus. When the data is noisy, this method generates large number of rules with low predictive accuracy. Reference [4] presented FuzzySRI-II that can effectively handle noise in the data.

The principal aim of this paper is to propose a new fuzzy rule induction algorithm called FuzzyRULES-II, able to deal properly with either numerical or nominal attributes, for the creation of predictive and classification models. FuzzyRULES-II employs a different learning and search strategy from that of FuzzySRI-II. Also, it introduces new discretisation methods for deriving fuzzy intervals for classification learning problems and for handling numerical outputs. This results in the efficient extraction of accurate and compact fuzzy rules from large and noisy data.

The paper is organised as follows. Section 2 reviews the existing discretisation methods. Section 3 describes the new FuzzyRULES-II algorithm. The

*Corresponding author. E-mail: ash_afify@yahoo.com

performance of the proposed algorithm is evaluated in section 4. Section 5 concludes the paper and discusses further research.

2. Discretisation methods

Discretisation of a numerical attribute in the context of rule induction involves partitioning the domain of that attribute into a certain number of intervals. An important aspect of discretisation is to find an appropriate set of cutting points to set up interval borders.

Current discretisation methods are categorised based on predefined properties, such as supervised vs. unsupervised, multivariate vs. univariate, off-line vs. on-line, and parametric vs. non-parametric [5]. *Equal-width interval* discretisation [6] is possibly the simplest unsupervised discretisation procedure. In this method, the range of a numerical attribute is divided into a number of equal intervals. A problem with the equal-width method is that it does not take the relationship between the class label and the numerical attribute, which results in possible loss of classification information. Reference [7] presented a simple supervised discretisation method called *1R Discretiser*. This method divides the values of a numerical attribute into several disjoint intervals each containing a strong majority of one particular class with the constraint that each interval must include at least a prespecified number of examples in the same class. Reference [8] developed a recursive entropy-based discretisation method that is motivated by the C4.5 decision tree induction algorithm [9]. To improve the discretisation efficiency, the method examines only the boundary points of each numerical attribute rather than all its distinct values. It also uses the minimum description length (MDL) principle to control the number of intervals produced over the search space. Reference [10] introduced an *optimal* discretisation method that obtains optimal number of intervals through searching only the boundary points of each numerical attribute and selecting those points that optimise an MDL-based evaluation function.

The aforementioned discretisation methods partition the ranges of numerical attributes into "crisp" intervals, where a numerical attribute value must belong to only one interval. This causes substantial information loss. To overcome this problem, this paper uses the techniques of fuzzy sets to construct "fuzzy" intervals. This allows overlapping in the adjacent intervals, and thus attribute values can belong to more than one interval.

3. Description of the proposed FuzzyRULES-II algorithm

FuzzyRULES-II is based on the learning strategy employed in the FuzzyRULES classification rule induction algorithm [11]. This section describes the proposed FuzzyRULES-II algorithm and highlights its distinctive features.

3.1. Representation

FuzzyRULES-II creates fuzzy if-then rules directly from a set of examples called the training set *T*. Each example *E* is described by a vector of n_a input attributes ($A^1, .., A^i, ..., A^{n_a}$) and an output attribute A^o . Each input attribute value v_E^i and the output attribute value v_E^o in example *E* are either nominal or numerical. An example *E* can thus be formally described as follows:

$$E = (A^1 = v_E^1 , ..., A^i = v_E^i , ..., A^{n_a} = v_E^{n_a} , Class = v_E^o) \quad (1)$$

A fuzzy rule *R* is described by a conjunction of conditions on each input attribute ($Cond_R^i$) and an output condition ($Cond_R^o$) of the class to be learned. It can be defined as follows:

$$R = Cond_R^1 \wedge, ..., \wedge Cond_R^i \wedge, ..., \wedge Cond_R^{n_a} \rightarrow Cond_R^o \quad (2)$$

For the i^{th} input attribute A^i , a fuzzy condition has the form [A^i is L_R^i], where L_R^i is the linguistic value of the i^{th} attribute in rule *R*. Similarly, the output fuzzy condition has the form [A^o is L_R^i], where L_R^o is the linguistic value of the output attribute in rule *R*. Each linguistic value represents a fuzzy set for which every example in the training set has a corresponding membership value. For numerical attributes, linguistic values can be obtained by defining membership functions on the domains of the attributes. Typical shapes of membership functions are triangular, trapezoidal, and bell-shaped. FuzzyRULES-II adopts triangular forms in this study as

they are simple and often used in fuzzy sets. A triangular membership function can be described as $Tr(a,b,c)$, where ac is the base of the triangle and b the location of its apex. The values of nominal attributes can be converted to linguistic values by assigning crisp functions with membership degrees of either 0 or 1. A rule set is a disjunction of a number of rules $\{R_1, ..., R_l, ..., R_{n_r}\}$, and it is defined as $RS = \{R_1 \vee ... \vee R_l \vee ... \vee R_{n_r}\}$.

3.2. Handling numerical input attributes

When there are both numerical and nominal attributes in a data set, most rule induction techniques discretise numerical attributes into intervals and the discretised intervals are treated similarly to nominal values during induction. In the crisp case, discretisation results in "crisp" intervals and an attribute value either belongs to a certain interval or not. In the fuzzy case, however, an attribute value belongs to an interval to a certain degree. Discretised crisp intervals, therefore, should be fuzzily interpreted.

The FuzzyRULES algorithm uses an intuitive way to obtain fuzzy intervals for numerical attributes. First, it discretises the domain of each attribute into several crisp intervals using an efficient off-line discretisation method. Second, it derives fuzzy intervals from the crisp ones by defining membership functions on the domains of the attributes.

Various studies [12] have pointed out that the selection of a discretisation method depends on both the learning algorithm and on the data to be discretised. Therefore, an empirical evaluation was carried on a large number of data sets to assess the four discretisation methods mentioned in section 2 when used with the FuzzyRULES-II algorithm. Each of the discretisation methods was first employed to create crisp intervals for all the numerical attributes. The fuzzification method of the FuzzyRULES algorithm was then employed to fuzzify these intervals. The fuzzy intervals were finally used to generate fuzzy classification rules by the FuzzyRULES-II algorithm. The results (see section 4) demonstrated that the performance of the FuzzyRULES-II algorithm considerably improved when the entropy method was used to discretise the numerical attributes. Thus, this discretisation method is employed by FuzzyRULES-II.

3.3. Handling numerical output attributes

The rule forming process of the FuzzyRULES algorithm is mainly designed to handle examples with nominal classes. Therefore, to enable the use of this process, the numerical output values of all examples in the training set T need to be fuzzified. It is proposed to split the numerical output range of each example ($v^o_{\min}$, $v^o_{\max}$) into a user-defined number (n_l) of linguistic values, where $v^o_{\min}$ and $v^o_{\max}$ are respectively the minimum and maximum known values for the output attribute.

Given the numerical output attribute range and the number n_l of required linguistic values, the FuzzyRULES-II algorithm decomposes the output range into n_l triangular linguistic values (L^o_1 , .. , L^o_k , .. , $L^o_{n_l}$) defined as: $L^o_k = Tr(a(k), b(k), c(k))$, where k is an integer included in [1, n_l],

$b(k) = [(v^o_{\max} - v^o_{\min})/(n_l - 1)].(k-1)$,

$a(k) = b(k) - (v^o_{\max} - v^o_{\min})/(n_l - 1)$, and

$c(k) = b(k) + (v^o_{\max} - v^o_{\min})/(n_l - 1)$.

Based on this decomposition, there could be two possible linguistic values with respect to which the output attribute value (v^o_E) in a given example E can be assigned. The output linguistic value will be the value L^o_k where the membership degree $\mu_{L^o_k}(v^o_E)$ is maximum, $v^o_E \in]\big(b(k) - a(k)\big)/2, \big(c(k) - b(k)\big)/2[$, and $\mu_{L^o_k}(v^o_E) > 0.5$. In the particular case where the membership degree of v^o_E is equal for two linguistic values (L^o_k and L^o_{k+1}), i.e. $v^o_E = \big(b(k+1) - a(k+1)\big)/2 = \big(c(k) - b(k)\big)/2$, and $\mu_{L^o_k}(v^o_E) = 0.5$, one is selected randomly.

Now that the output linguistic values are defined, FuzzyRULES-II can select an example with its corresponding output linguistic value to form a fuzzy rule. The rule forming process of FuzzyRULES is then used unchanged. At the end of the rule formation process, a rule R is obtained to cover as many positive examples and as few negative examples as possible.

FuzzyRULES-II adopts the following defuzzification strategy to predict the output value of a particular example E. First, the membership degree of the example E with each rule R, $\mu_R(E)$, is computed, namely:

$$\mu_R(E) = \prod_{i=1}^{n_a} \mu_{L_R^i}(v_E^i) \quad (3)$$

where $\mu_{L_R^i}(v_E^i)$ is the membership degree of each attribute value (v_E^i) in the example E with regard to the corresponding linguistic value L_R^i in the rule R. Then, the weighted average method [13] is used to compute the defuzzified output:

$$output = \sum_{l=1}^{n_r} \mu_{R_l}(E) \cdot C_{out} \,/\, \sum_{l=1}^{n_r} \mu_{R_l}(E) \quad (4)$$

where E is the new example, C_{out} the centre of the output fuzzy set of the rule being considered and n_r the total number of rules.

4. Experimental results

A series of tests was conducted to assess the performance of four different well-known off-line discretisation methods when used in the FuzzyRULES-II algorithm. Then, the performance of the FuzzyRULES-II algorithm with the best performing discretisation method was compared against that of RULES-F Plus [3].

4.1. Empirical evaluation of fuzzy discretisation methods

The evaluation involves the four discretisation methods described in section 2, namely, the *equal-width* method, the *1R Discretiser*, the *entropy-based discretisation* method and the *"optimal"* discretisation method. These methods are commonly used in other comparative studies. Each of the discretisation methods was first employed to create crisp intervals for numerical attributes in all the data sets. The crisp intervals were then fuzzified and used to generate classification rules by the FuzzyRULES-II algorithm. Three criteria were used to evaluate the quality of the discretisation, namely, the accuracy and complexity of the generated rules, as well as the time of execution measured by the total CPU time in seconds and the number of rules evaluated during the search process. All running times were obtained on an Intel Core i5 computer with a 3.33 GHz processor, 4 GB of memory and the Windows NT 4.0 operating system.

Table 1 summarises the data sets used in this experiment [14]. The test method of FuzzyRULES was used [11]. FuzzyRULES-II and discretisation methods each has a number of parameters whose values determine the quality of their induced rule sets. For FuzzyRULES-II, the default parameters of FuzzyRULES were used [11]. For the equal-width discretisation method, the number of intervals was set to 6. For the 1R Discretiser, the number of examples in one interval was set to 6 for large data sets, while the number was set to 3 for small data sets as recommended in [7].

Table 2 gives the results of the considered discretisation methods when used in FuzzyRULES-II. As can be seen in the table, the entropy-based discretisation method obtained total accuracy higher than that achieved with the other methods. In addition, it produced notably fewer rules in total than the other discretisation methods. Also, the total CPU time and the number of rules explored by the entropy-based method were much better compared to the other methods. It could therefore be concluded that the entropy-based discretisation method outperforms the four methods tested. Consequently, this method is utilised by FuzzyRULES-II.

4.2. Comparison with RULES-F Plus

FuzzyRULES-II is compared with RULES-F Plus using two practical problems that aim to develop models for the control of a robotic arm and a truck.

Table 1

Summary of the data sets used in the experiments

Data Set Name	No. of Instances	No. of Nominal Attributes	No. of Numerical Attributes	No. of Classes
Abalone	4177	1	7	29
Adult	48842	8	6	2
Anneal	898	32	6	6
Australian	690	8	6	2
Auto	205	10	15	6
Balance-scale	625	0	4	3
Breast	699	0	10	2
Cleve	303	7	6	2
Crx	690	9	6	2
Diabetes	768	0	8	2
German	1000	13	7	2
German-organisation	1000	12	12	2
Glass	214	0	9	7
Glass2	163	0	9	2
Heart-disease	270	0	13	2
Heart-Hungarian	294	5	8	2
Hepatitis	155	13	6	2
Horse-colic	368	15	7	2
Hypothyroid	3163	18	7	2
Ionosphere	351	0	34	2
Iris	150	0	4	3
Letter	20000	0	16	26
Lymphography	148	15	3	4
Satimage	6435	0	36	6
Segment	2310	0	19	7
Shuttle	58000	0	9	7
Sick-euthyroid	3163	18	7	2
Sonar	208	0	60	2
Tokyo	961	0	46	2
Vehicle	699	0	18	4

Table 2

Performance of discretisation methods when used in Fuzzy-RULES-II

	Entropy				Optimal				1RD				Equal-width			
Data Set Name	Acc. (%)	# Rules	# Rules explored	Time (s)	Acc. (%)	# Rules	# Rules explored	Time (s)	Acc. (%)	# Rules	# Rules explored	Time (s)	Acc. (%)	# Rules	# Rules explored	Time (s)
Abalone	**25.3**	5	627	1	24.7	5	774	1	24.7	**4**	**117**	**0**	24.8	5	765	1
Adult	83.9	47	14734	290	**84.2**	114	42466	830	84.1	48	**13600**	**275**	79.7	**44**	14683	303
Anneal	97.7	**9**	1597	**1**	**98.2**	**12**	**2087**	**1**	**98.1**	**10**	**1453**	**1**	96.3	11	2315	2
Australian	87.5	**5**	2492	**0**	**83.5**	**6**	**2970**	**0**	**91.5**	**6**	**1606**	**0**	**88.0**	**8**	**2739**	**0**
Auto	**62.8**	6	922	**0**	**57.0**	**5**	**430**	**0**	**59.9**	**7**	**651**	**0**	**49.7**	**8**	**714**	**0**
Balance-scale	73.2	**6**	**133**	**0**	84.8	12	224	1	74.3	7	138	**0**	**90.6**	21	311	1
Breast	96.1	**5**	**232**	**0**	**96.5**	**7**	**236**	**0**	**98.5**	**6**	**343**	**0**	**97.5**	**9**	**251**	**0**
Cleve	**79.2**	5	816	**0**	**74.1**	**5**	**749**	**0**	**79.2**	**6**	**921**	**0**	**76.9**	**3**	**436**	**0**
Crx	83.0	9	3268	1	**85.3**	8	2915	1	83.5	**7**	**1913**	**0**	**80.9**	**7**	2216	1
Diabetes	**76.6**	**6**	**315**	**0**	**70.3**	**7**	**577**	**0**	**69.9**	**19**	**937**	**0**	73.2	21	1837	1
German	**72.0**	**10**	**5472**	**2**	**70.0**	**12**	**6772**	**2**	68.6	14	6923	3	67.3	13	9399	4
German-org.	**74.7**	**6**	4102	3	72.7	7	5153	4	74.1	7	**4043**	**2**	70.8	10	7744	6
Glass	**69.4**	**6**	**208**	**0**	**65.3**	**8**	**422**	**0**	**64.6**	**22**	**773**	**0**	**64.6**	**20**	**1184**	**0**
Glass2	**83.6**	**4**	**43**	**0**	**79.7**	**5**	**67**	**0**	**78.8**	**11**	**248**	**0**	**77.8**	**10**	**380**	**0**
Heart-disease	77.8	**5**	**480**	**0**	**72.6**	**6**	**483**	**0**	**78.8**	**6**	**661**	**0**	**76.7**	**7**	**826**	**0**
Heart-Hungarian	**77.6**	**3**	**417**	**0**	**77.6**	**3**	**501**	**0**	**76.6**	**4**	**636**	**0**	**75.6**	**6**	**1086**	**0**
Hepatitis	82.7	3	**339**	**0**	**84.6**	**3**	**372**	**0**	**84.6**	**3**	**363**	**0**	**76.9**	**2**	**379**	**0**
Horse-colic	**85.3**	7	1968	**1**	**77.2**	**6**	**1633**	**1**	**79.1**	**8**	**1519**	**1**	**74.4**	**9**	**1937**	**1**
Hypothyroid	98.7	**11**	**1064**	**1**	**98.9**	**11**	1337	2	**99.8**	15	1661	2	99.0	23	4486	6
Ionosphere	**94.0**	**7**	966	**0**	**92.3**	**9**	**1155**	**0**	**93.2**	**12**	**647**	**0**	92.3	14	1690	1
Iris	**96.0**	**3**	15	**0**	**96.0**	**3**	**13**	**0**	**96.0**	**3**	**11**	**0**	**94.0**	**5**	**17**	**0**
Letter	**68.2**	**241**	**55965**	42	61.4	346	73549	49	56.4	308	61943	56	52.2	357	81357	44
Lymphography	**82.0**	5	638	**0**	**70.6**	**4**	**486**	**0**	**70.6**	**4**	**478**	**0**	**80.1**	**6**	**639**	**0**
Satimage	82.2	69	**38005**	**100**	**85.4**	**68**	52213	152	83.7	73	42743	132	82.5	77	71218	190
Segment	91.1	27	2947	**2**	**94.7**	**26**	**3316**	**2**	**80.9**	**58**	**2198**	**2**	88.7	32	4616	**4**
Shuttle	**99.6**	36	1521	48	99.1	16	1147	32	94.6	18	1262	39	89.8	**9**	**577**	**16**
Sick-euthyroid	**97.4**	12	**2005**	**2**	**95.8**	**11**	**2122**	**2**	**95.1**	**15**	**3668**	**2**	90.1	16	4581	6
Sonar	75.7	**4**	**1121**	0	78.8	4	1121	0	72.9	10	3972	1	**81.9**	9	5622	1
Tokyo	93.3	**5**	3239	1	94.8	6	3234	1	90.9	7	**1133**	**0**	**95.9**	8	3276	1
Vehicle	67.4	**15**	2839	**1**	**69.9**	**19**	**3522**	**1**	**61.1**	**33**	**2497**	**1**	63.2	29	4682	2
Total	**2434.1**	**582**	**148490**	**496**	2395.9	754	212046	1082	2364.1	751	159058	517	2351.4	799	231963	591

These two problems were previously employed to evaluate the RULES-F Plus algorithm [3]. Three measures were used to evaluate the performance of the tested algorithms. These are the number of rules produced, the maximum absolute error, max E_{abs}, and the mean absolute error, mean E_{abs}.

4.2.1. Robot arm control problem

The problem involved the building of a fuzzy model for the control of a PUMA 560 robot [15]. The training set T contains 27,825 examples, each of which is composed of six input attributes (the joint angles $\theta 1_t$, $\theta 2_t$, $\theta 3_t$, $\theta 1_{t-1}$, $\theta 2_{t-1}$, and $\theta 3_{t-1}$ at times t, and t-1) and three outputs X_{t+1}, Y_{t+1}, and Z_{t+1} representing the resulting spatial positions.

Models were created using RULES-F Plus and FuzzyRULES-II. For both algorithms, the three outputs were decomposed into ten linguistic values. In addition, a training set was created for each output as RULES-F Plus and FuzzyRULES-II can only handle one output at a time. Finally, the parameters of RULES-F Plus and FuzzyRULES-II were set as follows. For RULES-F Plus, the PRSET_size was set to 2 and no pruning process was employed as recommended in [3]. For FuzzyRULES-II, the default parameters of FuzzyRULES were used [11].

Results are illustrated in Table 3. Considering all performance measures, FuzzyRULES-II clearly outperforms the RULES-F Plus algorithm. FuzzyRULES-II produced a much smaller model than that generated by RULES-F Plus, with smaller max Eabs and mean Eabs values. To demonstrate its performance, the predictions of the model created by FuzzyRULES-II for the first 10000 examples compared with the actual outputs X, Y, and Z are shown in Figures 2, 3, and 4 respectively.

Table 3

Fuzzy induction results for the robot arm control problem

	Robot arm control											
	Combined results			*X* position			*Y* position			*Z* position		
	Total # of rules	Aver max	Aver mean	# of rules	Max E_{abs}	Mean E_{abs}	# of rules	Max E_{abs}	Mean E_{abs}	# of rules	Max E_{abs}	Mean E_{abs}
RULES-F Plus	55	0.0754	0.0129	13	0.1025	0.0222	14	0.0889	0.0121	28	0.0438	0.0044
FuzzyRULES-II	37	0.0478	0.0079	8	0.0721	0.0151	10	0.0413	0.0064	19	0.0301	0.0021

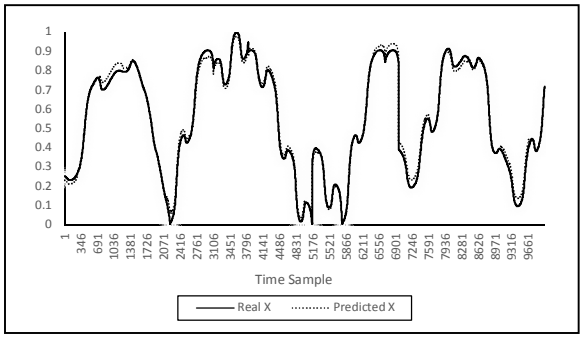

Fig. 2. Prediction of output X.

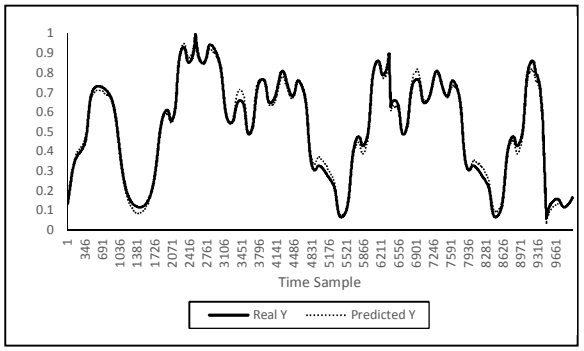

Fig. 3. Prediction of output Y.

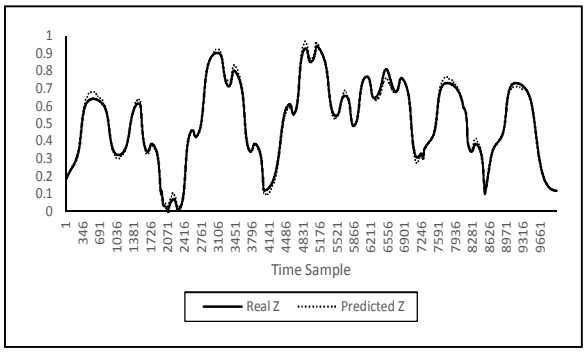

Fig. 4. Prediction of output Z.

4.2.2. Truck control problem

This problem aimed at constructing a model for the control of a truck reversing to a specified loading dock (Figure 5). The training set T contains 238 examples, each of which is composed of 2 input attributes φ (the angle of the truck relative to the horizontal) and x (the location of the truck on the horizontal axis), and one output θ (the required steering angle).

For both RULES-F Plus and FuzzyRULES-II, the output was decomposed into seven linguistic values. Two scenarios were considered when applying RULES-F Plus. In the first scenario, the noise level (*NL*) was set to 0.25, and in the second the IPP pruning process was turned off [3]. For both cases, *PRSET_size* was set to 2.

Results are given in Table 4. FuzzyRULES-II once again clearly outperforms RULES-F Plus. When using the IPP pruning process, RULES-F Plus created a more compact rule set but with a reduction in accuracy. However, for the case illustrated, the rule set created by FuzzyRULES-II was more compact and still more accurate than the model created by the RULES-F Plus algorithm.

5. Conclusions and future work

In the induction of classification rules from numerical attributes, the key point is to generate an appropriate discretisation of a numerical attribute. The crisp discretisation methods used in classical rule induction are noise sensitive, and thus prone to misclassification. This paper has presented a new fuzzy rule induction algorithm called FuzzyRULES-II in order to better handle numerical attributes. FuzzyRULES-II employs fuzzy sets to derive fuzzy intervals from those generated by four commonly used crisp discretisation methods. Also, it introduces a new approach for handling numerical outputs. The result is a powerful algorithm that can be used for classification or prediction of numerical as well as nominal values.

Additional tests could be performed to compare FuzzyRULES-II with other fuzzy algorithms. A method to automate the creation of output membership functions could also be considered to increase the robustness of the learning algorithm further.

Acknowledgment

The author wishes to thank the Industrial Engineering Department at Zagazig University, Egypt for providing a good environment, facilities and financial means to complete this paper.

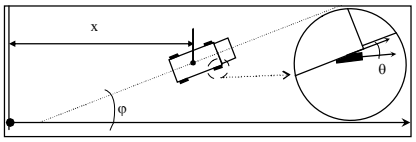

Fig. 5. Truck control problem.

Table 4

Fuzzy induction results for the truck control problem

	Truck control		
	# of rules	Max E_{abs}	Mean Eabs
RULES-F Plus	46	11.4	3.46
RULES-F Plus, $NL = 0.25$	12	42.3	6.24
FuzzyRULES-II	9	9.2	2.41

References

[1] I.H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 3rd ed., Morgan Kaufmann Publishers, 2011.

[2] L.X. Wang and J.M. Mendel, Generating fuzzy rules from numerical data, with applications, Signal and Image Processing Institute, University of Southern California, 1991.

[3] S. Bigot, A new rule space representation scheme for rule induction in classification and control applications, Proc. IMechE, Part I: J. Syst and Cont Eng, 225 (2011), 1018-1038.

[4] Afify, A.A, FuzzySRI-II: A fuzzy rule induction algorithm for numerical output prediction, Proc. World Congress on Engineering, London, UK, 2014.

[5] H. Liu, F. Hussain, C.L. Tan and M. Dash, Discretisation: An enabling technique, Data Min Knowl Disc, 6 (2002), 393-423.

[6] A.K.C. Wong and D.K.Y. Chiu, Synthesizing statistical knowledge from incomplete mixed-mode data, IEEE Trans. Pattern Analysis Mach. Intell., 9 (1987), 796-805.

[7] R.C. Holte, Very simple classification rules perform well on most commonly used data sets, Mach Learn, 11 (1993), 63-90.

[8] U.M. Fayyad and K.B. Irani, Multi-interval discretisation of continuous-valued attributes for classification, Proc. 13th Int. Joint Conf. Artificial Intell., France, 1993, pp. 1022-1027.

[9] J.R. Quinlan, C4.5: Programs for Machine Learning, 1993.

[10] Z. Cai, Technical Aspects of Data Mining, Ph.D. Dissertation, University of Wales Cardiff, Cardiff, UK, 2001.

[11] A.A. Afify, FuzzyRULES: A fuzzy rule induction algorithm for mining classification knowledge, Proc. 7th IEEE Conf. Industrial Informatics, Cardiff, UK, 2009, pp. 837-842.

[12] J. Dougherty, R. Kohavi and M. Sahami, Supervised and unsupervised discretisation of continuous features, Proc. 12th Int. Conf. Machine Learning, California, 1995, pp. 194-202.

[13] L. Rondeau, R. Ruelas, L. Levrat and M. Lamotte, A defuzzification method respecting the fuzzification, Fuzzy Sets and Systems, 86 (1997), 311-320.

[14] C.L. Blake and C.J. Merz, UCI Repository of Machine Learning Databases, 1998. Available from: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[15] B. Armstrong and O. Khatib, The explicit dynamic model and inertial parameters of the PUMA 560 Robot arm, Proc. Int. Conf. Robotics and Aut., San Francisco, 1986, pp. 510-518.