Generating and Applying Rules for Interval Valued Fuzzy Observations

Andre de Korvin
Department of Computer and Mathematical Sciences, University of Houston-Downtown,Houston, Texas 77002
Chenyi Hu
Department of Computer Science, University of Central Arkansas,Conway, AR 72035.
Ping Chen
Department of Computer and Mathematical Sciences, University of Houston-Downtown,Houston, Texas 77002

Abstract. One of the objectives of intelligent data engineering and automated learning is to develop algorithms that learn the environment, generate rules, and take possible courses of actions. In this paper, we report our work on how to generate and apply such rules with a rule matrix model. Since the environments can be interval valued and rules often fuzzy, we further study how to obtain and apply rules for interval valued fuzzy observations.

1 Introduction

In artificial intelligence, knowledge-based agents are designed and implemented to observe the environments and to reason about their possible courses of actions [17]. In such automated decision making systems, decisions are usually made through matching input data (relevance of each environment feature) with a certain set of rules. Examples of such systems are widely available in the literatures of fuzzy systems [3, 9, 12, 15, 19] and neuro-fuzzy systems [13, 16].

1.1 The rule matrix model

Assume that an environment e contains m features, and n possible different decisions, $d_1, d_2, \cdots, d_n$, that could be made based on the presences of the environment features. Let $e = (e_1, e_2, \cdots, e_m)^T$ be an observation of the environment, i.e., e denotes the degree to which certain features of an environment are present. Then, a knowledge-based agent may select a specific decision according to the m by n matrix below:

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ & \cdots & \cdots & \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix} \tag{1}$$

by matching the input e with column vectors of P. If the observation vector e matches P_j, the j^{th} column of P, then the j^{th} decision d_j should be selected. We call P the rule matrix. The decision making process is called the rule matrix model. In this paper, our study is focused on the rule matrix model.

1.2 Problems to be addressed in this paper

In this paper, we mainly address the following two problems:

1. How to establish and adjust a rule matrix P; and
2. How to make a reasonable decision from a general observation that does not match any column of P?

We address the above two questions in section 2 and 3 respectively. Then, we extend the results to interval valued training set, fuzzy rule matrix, and observations in section 4. We conclude our study in section 5.

2 Establishing and adjusting the rule matrix

The main purpose of this section is to sketch a method that estimates the rule matrix P. Without losing of generality, we normalize the environment observation vector e such that $0 \leq e_i \leq 1 \ \forall i \in \{1, 2, \cdots, m\}$ and $\Sigma_{i=1}^{m} e_i = 1$. We also assume that each column vector of P is also normalized which means that $0 \leq p_{ij} \leq 1 \ \forall j \in \{1, 2, \cdots, n\}$ and $\forall i \in \{1, 2, \cdots, m\}$; and $\Sigma_{i=1}^{m} p_{ij} = 1$ for any given $j \in \{1, 2, \cdots, n\}$.

Let E be a known data set that contains N environment-decision pairs. Since we have used a subscript e_k to indicate the k^{th} feature of an environment, we use a superscript e^k to denote the k^{th} observation of the environment. Then, an environment-decision pair, $[e^k, d_{k^*}]$ in E represents the desired decision d_{k^*} under a given environment e^k where, $1 \leq k \leq N$ and $1 \leq k^* \leq n$. A naive way to determine P_j, the j^{th} column of P, is to let $P_j = e^k$ if $j = k^*$. This certainly ensures that the j^{th} decision will be selected if the environment is e^k. However, this simple method will not work appropriately since the same decision d_j may be taken for different environment observations and very likely, in most cases, $n << N$.

Reasonable properties of P_j should include the following: It should be close enough to all of these e^ks such that $k^* = j$ and far away from those e^ks that $k^* \neq j$. Since observation vectors and columns of rule matrices are normalized, to determine P_j, we need to solve the problem below:

$$\text{To find a } P_j \text{ that minimizes: } W_j = \sum_{k^*=j} ||e^k - P_j|| + \sum_{k^* \neq j} (1 - ||e^k - P_j)||) \quad (2)$$

It is assumed that a feature extraction has been previously performed, thus the feature vectors generating a decision are presumably well separated. An alternate method would be to do a least square fit for each decision k^*. If we use the 2-norm in equation (2), then the problem we need to solve is to find a P_j that

$$\text{Minimizes: } W_j = \sum_{k^*=j} (\sum_{i=1}^{m} (e_i^k - p_{ij})^2) + \sum_{k^* \neq j} (1 - (\sum_{i=1}^{m} (e_i^k - p_{ij})^2)) \qquad (3)$$

Since all environment-decision pairs, $[e^k, d_{k^*}]$s, are given, we can solve (3) numerically. Hence, we have an algorithm to establish the rule matrix P:
Algorithm 2.1:

```
for k = 1 to N
  input environment-decision pairs
  normalize the environment vectors
for j = 1 to n
  minimize (3) subject to the normalization condition
```

The matrix obtained by Algorithm 2.1 is from the training set E. Since new data may be obtained from time to time, we may adjust the matrix P "online" dynamically with newly available data.

3 Making decision from an observation

3.1 The problem

After obtaining the rule matrix P, for a new observation e, one may pick the decision d_{j^*} if $||e - P_{j^*}|| = \min_{j \in \{1,2,\cdots,n\}} ||e - P_j||$. However, there are still questions that need to be answered. For a general observation $e \neq P_j, \forall j \in \{1, 2, \cdots, n\}$, there could be possibly multiple output of the j^*. Multiple j^* may come from calculating j^* with different norms, or from the fact that e almost equally close to several columns of P. In addition, the measurement of e may not be exact due to noise. Therefore, we need to further study decision making with fuzzy systems.

3.2 Decision making with fuzzy logic

Let us first recall a few basic definitions and facts about inference in a fuzzy system.

- If U denotes a set, a fuzzy subset of U is characterized by a function φ from U into $[0, 1]$. The function φ is called the membership function of the set.
- If we denote a fuzzy subset of U by A and if φ is its membership function, then for $u \in U$, $\varphi(u)$ denotes the membership of u in A. Of course, if A is a standard subset (i.e., a "crisp subset") of U, $\varphi(u)$ is either 1 (i.e., $u \in A$) or 0 (i.e., $u \notin A$).

– Let p_{ij} and d_j be fuzzy sets for $1 \leq i \leq m$, and $1 \leq j \leq n$. A fuzzy rule R_j for an m-dimensional vector $e = (e_1, e_2, \cdots, e_m)$, is defined as if e_1 is p_{1j}, e_2 is p_{2j}, $\cdots$, e_m is p_{mj}, then d is d_j;
– For an input $e = (e_1, e_2, \cdots, e_m)$, $\varphi_j(e) = \dfrac{\Pi_{k=1}^{m} \varphi_{kj}(e_k)}{\Sigma_{l=1}^{n} \Pi_{k=1}^{m} \varphi_{kl}(e_k)}$ denotes the strength of the rule R_j relative to e. That is to what extent rule R_j should be counted (on a scale of 0 to 1) when input e is applied.

For additional information on fuzzy systems and inference, we refer readers to [3, 9, 15, 19].

We now build the matching process between e and the j^{th} column of P in terms of fuzzy systems. Here after, we replace the entries p_{ij} of P as fuzzy sets whose membership functions are triangular functions, having value 1 at p_{ij} obtained from Algorithm 2.1. More specifically, if we let σ be a positive real number less than 1, we can define the triangular function φ_{ij} as the follow:

$$\varphi_{ij}(x) = \begin{cases} 1 & \text{if } x = p_{ij} \\ \dfrac{x}{\sigma p_{ij}} + 1 - \dfrac{1}{\sigma} & \text{if } (1-\sigma)p_{ij} < x < p_{ij} \\ \dfrac{-x}{\sigma p_{ij}} + 1 + \dfrac{1}{\sigma} & \text{if } p_{ij} < x < (1+\sigma)p_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Then, the strength of the j^{th} rule for given input e can be defined as

$$\varphi_j(e) = \frac{\Pi_{k=1}^{m} \varphi_{kj}(e_k)}{\Sigma_{j=1}^{n} \Pi_{k=1}^{m} \varphi_{kj}(e_k)} \tag{5}$$

Through using fuzzy rules, the $\varphi_j(e)$ above provides a degree of matching between e and the antecedent of each rule/decision d_j. One could select the d_{j^*} with the strongest strength, which means that $\varphi_{j^*}(e) = \max_{j \in \{1,2,\cdots,n\}} \varphi_j(e)$, as the decision. Any t-norm could be used instead of the product. We thus obtain the normalized strengths of the rules.

3.3 An index approach for decision making

Instead of applying a rule with the strongest strength, one may want to consider all rules whose strength are above a given threshold $\tau \in [0,1]$. In order to do this, we define the center of gravity of these rules as

$$I(e) = \Sigma_{j=1}^{n} j\varphi_j(e) \tag{6}$$

Note that the antecedents of the rules are fuzzy with membership functions φ_j while the consequents are indices of appropriate decisions. $I(e)$ is "the index of the most relevant rule given the feature input e". Of course, $I(e)$ is not a whole number in general. But we could select the rule whose output is "the closest" to $I(e)$. This would be a reasonable way to pick the right rule provided the decisions are "clustered" in the right way which means that d_s is close to d_k if the column vectors of P, P_s and P_k are close.

The above reasoning follows the use of the center of gravity of the output when we have a set of fuzzy rules. See [15] and [19] for example.

4 Interval valued observations

4.1 Why intervals?

In the above discussion, we implicitly assume that training data set are point valued. However, in real world applications, it would be most appropriate to study interval valued observations [2, 18]. This is because of that (a) Environment observations usually contain errors. Using intervals to represent them is more appropriate than using points; (b) With interval valued input data, one will obtain interval valued rule matrix; and (c) Decisions need to be made with interval valued observation based on an interval rule matrix.

In this section, we use boldface letters to denote interval variables. For example, we use $\mathbf{x}$ to denote an interval, and $\underline{x}$ and $\overline{x}$ for its greatest lower bound and least upper bound respectively.

4.2 Using interval valued training set

We now consider an interval valued training set $(\mathbf{e}^1, j_1), \cdots, (\mathbf{e}^N, j_N)$, where $\mathbf{e}^i$ is a vector with interval valued components and $j_1, ..., j_N$ are integers in $\{1, 2, ..., n\}$.

With interval arithmetic [6–8, 10, 11], one may apply Algorithm 2.1 to obtain an interval valued rule matrix $\mathbf{P}$ such that $[\underline{p}_{ij}, \overline{p}_{ij}] \subseteq [0,1]$. To make a decision for an interval valued environment observation $\mathbf{e}$, an exact match would be $\forall k \in \{1, 2, \cdots, m\}$, $\mathbf{e}_k \subseteq [\underline{p}_{kj}, \overline{p}_{kj}]$ for a fixed j. This implies that decision d_j is the right one when features are presented as indicated by the interval vector $\mathbf{e}$. We now have introduced an uncertainty on the presence of features by specifying the lower and upper bound of that presence.

If for any fixed j the input $\mathbf{e}$ may not have the property that $\mathbf{e}_k \subseteq [\underline{p}_{kj}, \overline{p}_{kj}]$ for all k. There could be some overlap between intervals $\mathbf{e}_k$ and $[\underline{p}_{kj}, \overline{p}_{kj}]$. We need to extend the concept of strength of a rule defined previously in section 3. In order to define the strength of rule j relative to an interval valued observation vector $\mathbf{e} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_m]^T$, let us first define a trapezoidal function $\varphi_{ij}(x)$ for a real x similar to (4) to fuzzify the rule matrix:

$$\varphi_{ij}(x) = \begin{cases} 1 & \text{if } x \in \mathbf{p}_{ij} \\ \dfrac{x}{\sigma \underline{p}_{ij}} + 1 - \dfrac{1}{\sigma} & \text{if } (1-\sigma)\underline{p}_{ij} < x < \underline{p}_{ij} \\ \dfrac{-x}{\sigma \overline{p}_{ij}} + 1 + \dfrac{1}{\sigma} & \text{if } \overline{p}_{ij} < x < (1+\sigma)\overline{p}_{ij} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

To apply (7) for an interval $\mathbf{x}$, we get an interval

$$\phi_{ij}(\mathbf{x}) = [\min\{\varphi_{ij}(\underline{x}), \varphi_{ij}(\overline{x})\}, \max\{\varphi_{ij}(\underline{x}), \varphi_{ij}(\overline{x})\}] \tag{8}$$

For the i^{th} entry of an interval valued environment vector $\mathbf{e}$, denoted as $\mathbf{e}_i$, the largest intersection of $\phi_{i,j}$ and $\mathbf{e}_i$ is defined as the possibility function below

$$\text{Poss}[\phi_{i,j} | \mathbf{e}_i] = \sup\{\phi_{i,j}(v) \wedge \mathbf{e}_i(v)\} \tag{9}$$

where $\mathbf{e}_i(v) = \begin{cases} 1, & \text{if } v \in \mathbf{e}_i \\ 0, & \text{otherwise} \end{cases}$.

We define the strength of a specific rule/decision, say j, with respect to an interval observation $\mathbf{e}$ through the use of possibility functions as

$$\phi_j(\mathbf{e}) = \frac{\Pi_{k=1}^m \text{Poss}[\phi_{k,j} | \mathbf{e}_k]}{\Sigma_{t=1}^n \Pi_{k=1}^m \text{Poss}[\phi_{k,t} | \mathbf{e}_k]} \tag{10}$$

To make a decision for an input $\mathbf{e}$ based on interval valued fuzzy rule matrix, we may pick the strongest rule j^*, where $\phi_{j^*}(\mathbf{e}) > \phi_j(\mathbf{e})$ for $j \in \{1, 2, \cdots, n\}$. Then, d_{j^*} is the decision. We could also take the index approach (6) similarly provided that the decisions are arranged so that the distance between the columns $d_1, d_2, ...d_n$ reflects the distance between the interval vectors $d_1, d_2, ...d_n$.

Another approach for decision making with interval valued fuzzy rule matrix is based on the idea of the necessity function. A necessity function of $\phi_{i,j}$ with $\mathbf{e}_i$ is defined as the follow:

$$\text{Nec}[\phi_{i,j}|\mathbf{e}_i] = \sup_v\{[1 - \mathbf{e}_i(v)] \vee \phi_{i,j}(v)\} \tag{11}$$

We then have

$$\psi_j(\mathbf{e}) = \frac{\Pi_{k=1}^m \text{Nec}[\phi_{k,j}|\mathbf{e}_k]}{\Sigma_{t=1}^n \Pi_{k=1}^m \text{Nec}[\phi_{k,t}|\mathbf{e}_k]} \tag{12}$$

For more properties of possibilities and necessities, readers may refer [14] and [19]. One may select the decision according to the highest necessity as well. The term $|\psi_j(\mathbf{e}) - \phi_j(\mathbf{e})|$ reflects to some extend the uncertainty surrounding the input $\mathbf{e}$, since it represents the difference generalized by taking the possibility versus the necessity function.

Once the strength of rule R_j is defined relative to the interval input $\mathbf{e}$ as $\phi_j(\mathbf{e})$ if we use the possibility function, or as $\psi_j(\mathbf{e})$ if we use necessity function, there are many ways to obtain the fuzzy output generalized by $\mathbf{e}$ and then obtaining the defuzzification. We refer the reader to [9] to see these different approaches to defuzzification.

5 Summary

In this paper, we have studied the rule matrix model for an intelligent agent. To obtain a rule matrix for an agent, we use a training environment-decision data set. Since the training data set and the input observation may not be exact in real world applications, we have fuzzified the rule matrix and then developed methods of decision selection by calculating strength of a rule.

In addition of using point valued training data to obtain a rule matrix, we have applied interval arithmetic which makes us able to obtain interval rule matrix from interval valued training data set. By doing this, we have allowed uncertainty on the input. Through applying possibility and necessity functions, we yield an interval-valued functions ϕ_j and ψ_j for an input $\mathbf{e}$. Making a decision becomes a defuzzification problem. In case the input is a degenerated interval vector i.e. a point, then $\phi_j(e)$ will be the same as $\psi_j(e)$.

References

1. M. Beheshti, A. Berrashed, A. de Korvin, C. Hu, and O. Sirisaengtaksin, *On Interval Weighted Three-layer Neural Networks*, Proceedings of the 31 Annual Simulation Symposium, IEEE Computer Society Press, pp. 188-194, 1998.
2. D. Berleant and *et al*, *Dependable Handling of Uncertainty*, Reliable Computing, pp. 407-418, Volume 9, 2003.
3. A. de Korvin, C. Hu, and O. Sirisaengtaksin, *On Firing Rules of Fuzzy Sets of Type II*, J. Applied Mathematics, pp. 151-159, Volume 3, No.2, 2000.

4. A. de Korvin, R. Kleyle, and P. Lea, *An evidential approach to problem solving when a large number of knowledge systems are available*, The International Journal of Intelligent Systems, pp. 293-306, Volume 5, 1990.
5. A. de Korvin, R. Kleyle, and McKeegan, *Knowledge acquisition using rough sets when membership values are fuzzy sets*, The Journal of Intelligent and Fuzzy Systems, pp. 237-244, Volume 6, 1998.
6. E. Hansen, *Global optimization using interval analysis*, Marcel Dekker, New York, 1992.
7. B. Kearfott, G. Corliss, C. Hu, M. Schulte, and M. Stadtherr *Global Solutions: Entry Page*, http://www.mscs.mu.edu/~globsol.
8. B. Kearfott, M. Dawande, K. Du, and C. Hu, *Algorithm 737: INTLIB: a Portable Fortran-77 Interval Standard Function Library*, ACM, Trans. on Math. Software, Vol. 20, No.4, pp. 447-459, 1994.
9. J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice Hall, 2000
10. R. Moore, *Methods and Applications of Interval Analysis*, Society for Industrial and Applied Mathematics, 1979.
11. A. Neumaier, *Interval methods for systems of equations*, Cambridge University Press, Cambridge, 1990.
12. Z. Pawlak, *Rough sets and fuzzy sets and systems*, pp. 99-102 1985
13. L. Parel, M. Chelaru, *Neural Fuzzy Architecture for Adaptive Control*, Proc. of IEEE International Conf. on Fuzzy Systems 1115-1122, March 1992
14. W. Pedrycz, F. Gomide, *An Introduction To Fuzzy Sets Analysis and Design*, MIT Press, 1998
15. H. Rasiowa, *Towards fuzzy logic infuzzy logic for the management of uncertainty*, Ed L. Zadh and J. Kacprzyk, pp. 121-139, New York. Wiley Interscience.
16. J. S. Roger, *ANFIS: Adaptive Network-based Fuzzy Inference System*, IEEE Transactions on Systems, Man and Cybernetics, 23(03) 665-685, May 1993
17. S. Russell, P. Norvig, *Artificial Inteligence: A Modern Approach*, second edition, Prentice Hall, 2003.
18. S. Shary, *A New Technique in Systems Analysis Under Interval Uncertainty and Ambiguity*, Reliable Computing, pp. 321-418, Volume 8, 2002.
19. A. L. Zadeh, *Outline of a new approach to the analysis of complex systems and decision processes*, IEEE Transactions on Systems, Man and Cybernetics 3(1), pp. 28-44, 1973