



Spring 1-1-2019

Uncertain Knowledge Reasoning Based on the Fuzzy Multi-Entity Bayesian Network

Jinzhu Gao

University of the Pacific, jgao@pacific.edu

Dun Li

Zhengzhou University

Hong Wu

Zhengzhou University

Zhuoyun Liu

Zhengzhou University

Lun Li

Zhengzhou University

See next page for additional authors

Follow this and additional works at: <https://scholarlycommons.pacific.edu/soecs-facarticles>



Part of the [Engineering Commons](#)

Recommended Citation

Gao, J., Li, D., Wu, H., Liu, Z., Li, L., & Zheng, Z. (2019). Uncertain Knowledge Reasoning Based on the Fuzzy Multi-Entity Bayesian Network. *Computers, Materials & Continua*, 61(1), 301–321. DOI: [10.32604/cmc.2019.05953](https://doi.org/10.32604/cmc.2019.05953)
<https://scholarlycommons.pacific.edu/soecs-facarticles/139>

This Article is brought to you for free and open access by the All Faculty Scholarship at Scholarly Commons. It has been accepted for inclusion in All Faculty Articles - School of Engineering and Computer Science by an authorized administrator of Scholarly Commons. For more information, please contact mgibney@pacific.edu.

Authors

Jinzhu Gao, Dun Li, Hong Wu, Zhuoyun Liu, Lun Li, and Zhiyun Zheng

Uncertain Knowledge Reasoning Based on the Fuzzy Multi Entity Bayesian Networks

Dun Li¹, Hong Wu¹, Jinzhu Gao², Zhuoyun Liu¹, Lun Li¹ and Zhiyun Zheng^{1,*}

Abstract: With the rapid development of the semantic web and the ever-growing size of uncertain data, representing and reasoning uncertain information has become a great challenge for the semantic web application developers. In this paper, we present a novel reasoning framework based on the representation of fuzzy PR-OWL. Firstly, the paper gives an overview of the previous research work on uncertainty knowledge representation and reasoning, incorporates Ontology into the fuzzy Multi Entity Bayesian Networks theory, and introduces fuzzy PR-OWL, an Ontology language based on OWL2. Fuzzy PR-OWL describes fuzzy semantics and uncertain relations and gives grammatical definition and semantic interpretation. Secondly, the paper explains the integration of the Fuzzy Probability theory and the Belief Propagation algorithm. The influencing factors of fuzzy rules are added to the belief that is propagated between the nodes to create a reasoning framework based on fuzzy PR-OWL. After that, the reasoning process, including the SSFBN structure algorithm, data fuzzification, reasoning of fuzzy rules, and fuzzy belief propagation, is scheduled. Finally, compared with the classical algorithm from the aspect of accuracy and time complexity, our uncertain data representation and reasoning method has higher accuracy without significantly increasing time complexity, which proves the feasibility and validity of our solution to represent and reason uncertain information.

Keywords: Ontology language, uncertainty representation, uncertainty reasoning, fuzzy multi entity Bayesian networks, belief propagation algorithm, fuzzy PR-OWL.

1 Introduction

Data acquisition technology, storage technology and computer performance have gained rapid development in the information era, and the global data size is doubled every 20 months. Moreover, data not only are big in volume, but also involve increasingly diversified types as well as complicated and changeable data relations. In order to make the computer automatically process and integrate valuable data in network, the semantic web is proposed.

The increasing popularity of semantic web services makes it a challenging task to represent and reason uncertain data. However, the existing theory and practices do not provide

¹ Zhengzhou University, 100 Science Avenue, Zhengzhou, 450001, China.

² University of the Pacific, 3601 Pacific Avenue, Stockton, 95211, USA.

* Corresponding Author: Zhiyun Zheng. Email: iezyzheng@zzu.edu.cn.

enough feasible solutions to deal with the uncertainty of semantic web. To some extent, semantic markup languages, such as OWL (Ontology Web Language) or RDF (Resource Described Frame), can be used to signify the quality and quantity information of uncertainty, but they still have limitations. Previously proposed solutions include Probabilistic and Dempster-Shafer Models and Fuzzy and Possibilistic Models. BN and MEBN are two representative models based on probability and the corresponding ontology languages are OntoBayes based on BN [Yang (2007); Yang and Calmet (2006)], BayesOWL [Ding, Peng and Pan (2006)], and PR-OWL2 based on MEBN [Costa Laskey and Laskey (2008); Carvalho, Laskey and Costa (2013)], major ontology languages based on fuzzy model such as Fuzzy OWL [Calegari and Ciucci (2007)], etc. The existing uncertainty ontology languages cannot represent and reason probability and fuzzy information at the same time.

Fuzzy Multi Entity Bayesian Networks (MEBN) [Golestan, Karray and Kamel (2013, 2014)] was proposed to represent the fuzzy semantic and uncertainty relation between knowledge entities. This paper aims to extend the expression and reasoning ability of ontology for fuzzy probability knowledge and propose a method to denote and reason uncertainty.

Based on the ontology language, Fuzzy PR-OWL of Multi Entity Bayesian Networks, in this paper, we propose a novel reasoning method of fuzzy probability according to Fuzzy MEBN. Our main contributions are as follows:

- (1) The fuzzy probability ontology of Fuzzy PR-OWL is created;
- (2) Through integrating the fuzzy probability theory and Bayesian network, the reasoning framework of Fuzzy PR-OWL is created and the reasoning method of fuzzy Belief Propagation are proposed;
- (3) Experimental results prove the feasibility and validity of our proposed reasoning method.

2 Relevant theories

2.1 Fuzzy set and fuzzy probability system

Fuzzy set is a kind of description about fuzzy concept, which originates from the uncertainty and inaccuracy essence of abstract thought and concept, rather than the randomness of set elements. The logical value of fuzzy logic can be expressed with a real value in $[0, 1]$, which is the multivalued logic of fuzzy set. The fuzzy set theory transfers logical reasoning from two-valued logic into multivalued logic for predictive reasoning.

Zadeh [Zadeh (1968)] defined the probability of fuzzy event by extending the classical theory of probability. Suppose that x is the random variable in the sample space X and A is the subset that defines discrete events in X . The non-conditional probability of A can be calculated by Eq. (1):

$$P(A) = \int_{x \in A} f(x) dx = \int_{-\infty}^{\infty} X_A(x) f(x) dx \quad (1)$$

where $X_A(x)$ is the binary characteristic function, that is, $X_A(x) = 1 \leftrightarrow x \in A$ and $X_A(x) = 0 \leftrightarrow x \notin A$.

In fuzzy events, the characteristic function is the membership degree function. By replacing $X_A(x)$ in Eq. (3) with the membership degree function $\mu_A(x): x \in [0, 1]$, we can calculate the

probability of fuzzy events. Berg et al. [Berg, Kaymak and Rui (2013)] calculated the probability of fuzzy event A using Eq. (2):

$$P(A) = \int_{-\infty}^{+\infty} \mu_A f(x) dx \quad (2)$$

And the probability of fuzzy discrete event A can be calculated by Eq. (3):

$$P(A) = \sum_n p(x) \mu_A(x) \quad (3)$$

where $f(x)$ is the probability density function of variable x and $\mu(x)$ is its membership degree function. Eq. (3) assumes that the conditional probability density function is correlated with $\mu(x)$.

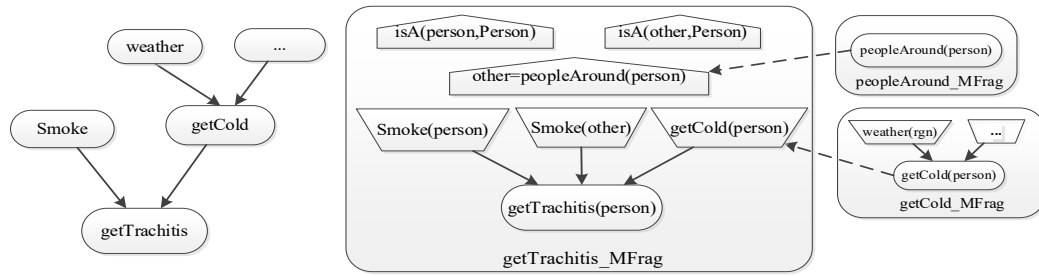
Suppose that $A_c (c=1, \dots, C)$ is an event in sample space X , and each discrete sample belongs to one or zero event. If a sample x has a group of membership degrees $\mu_{A_1}, \mu_{A_2}, \dots, \mu_{A_C}$, x is considered a fuzzy sample. To ensure the validity of probability theory, a good sample space is expected to meet the following condition:

$$\forall x: \sum_{A_c} \mu_{A_c}(x) = 1$$

2.2 BN and MEBN

Bayesian Networks (BN) [Heckerman (1997)] can deal with uncertain and probabilistic events according to the causal relationship of events or other relationships, and has the ability to process incomplete datasets, but BN has limitations when representing entity relationships. Fig. 1(a) shows relevant knowledge of tracheitis represented by BN. Tracheitis can be triggered by either smoking or getting cold can in turn be triggered by weather factors. The causal relationship can be clearly seen in the Fig. 1, but BN cannot show the influence of harmful gases produced by others' smoking behavior on the patients. Multi Entity Bayesian Networks (MEBN) [Chantas, Kitsikidis and Nikolopoulos (2014)] adopts the idea of "processing by division" by representing knowledge blocks with multi-entity fragments and then using multi-entity rules to link these fragments together. Fig. 1(b) shows relevant information of tracheitis represented by MEBN. Inherent nodes, input nodes, and context nodes are indicated by ovals, trapezoids, and pentagons, respectively. *other* and *person* are both examples of *Person* type. *other=peopleAround (person)* means that *other* is the neighbor to *person*. In this way, MEBN can show the relationship between the entities as well as the influence of others' smoking on the patient's bronchitis by adding the node *Smoke (other)*.

However, human experience or knowledge is fuzzy by nature and therefore cannot be handled by MEBN. As shown in the above example, severeness of cold may affect the probability of suffering tracheitis. But MEBN can only use 0 or 1 to represent the possibility of suffering cold. In addition, when an inherent node has a nominal value, MEBN would assign equal probability to all nominal scales. For example, suppose that the weather today has two possible nominal scales, *clear or cloudy*, MEBN would assign the probability value 0.5 to both scales. It fails when the weather is partially cloudy.



a. BN representation

b. MEBN representation

Figure 1: BN and MEBN Representation graph of bronchitis etiology

2.3 Fuzzy MEBN and fuzzy PR-OWL

2.3.1 Fuzzy MEBN

Fuzzy Multi Entity Bayesian Networks (Fuzzy MEBN) redefines the grammar specification by combining the first-order logic and the fuzzy MEBN theory, allowing the context constraints of MEBN to express the fuzziness of incomplete semantic information [Zheng, Liu and Li (2016)]. Hence, the expression and reasoning ability of inexact knowledge is improved. Therefore, fuzzy MEBN can represent fuzzy information that cannot be expressed with MEBN in Subsection 1.2, such as *Mild Cold* $\{true_{0.3} 1, false_{0.7} 0\}$, in which subscripts denote severeness of cold. Partially *cloud* can then be represented by $\{clear_{0.6} 0.5, cloudy_{0.4} 0.5\}$, in which subscripts denote the membership degree.

Fuzzy MEBN conducts modeling for specific domain ontology with the predefined attribute entity, semantics and their relationships. The attribute entity is represented by constant elements of a specific domain, and the variables are associated. The entities and relationships between entities are represented by logical and random variables of specific domain. As for the biggest difference between fuzzy MEBN and MEBN in entity and random variable symbol, the membership degree of a real number between 0 and 1 is added to the constant element symbol and entity identifier as subscript, such as $Vehicle_{0.85}$ and $!V_{428_{0.75}}$. The logic or value taking of random variable is the logical value in logical value chain $L = \langle l_1, l_2, l_3, \dots, l_n \rangle$ predefined by the language or within the scope of $[0, 1]$, rather than simple T or F .

The basic model of fuzzy MEBN is similar to that of MEBN. In fuzzy MEBN theory, the probability distribution is denoted by Fuzzy MEBN Fragments (*FMFrag*). The inherent random variable or input random variable value is set as the condition, and one *FMFrag* will define one probability distribution or several fuzzy rules for inherent random variable. The formalized definition of Fuzzy *MFrag* (*FMFrag*) is $F = (C, I, R, G, D, S)$, where C is the infinite set of context random variables, I represents the infinite set of input random variables, R indicates the infinite set of inherent random variables, G represents the directed loop-free fragment graph, comprised of variable nodes in $I \cup R$, D means the local distribution of every inherent random variable in R , and S denotes the set of fuzzy if-then rule used by fuzzy reasoning system. Context variables contain the semantic structure of knowledge that adopts fuzzy first-order logic. As the bridge with other inherent variables

of *FMFlags*, input variables will send relevant information to the current *FMFrag*. Inherent variables is random variables that set the values of context and input variables as conditions. The sets *C*, *I* and *R* have no intersection, and the random variable in *I* is the root node of fragment graph *G*. The context assignment item in *C* is used to strengthen the constraint of local distribution.

In *FMFrag*s, the context constraint is endowed with a true value, representing the degree of satisfying constraint. The consistency constraint degree of *FMFrag*s is determined by the fuzzy explanation of the item defined by *FMFrag* and built-in *FMFrag*s. When a common random variable is associated with an internal node, the aggregation function and the merging rule would be used.

2.3.2 Fuzzy PR-OWL

Fuzzy PR-OWL ontology adopted OWL2 DL language to define ontology class and attribute. In other words, OWL2 language is used to describe elements like classes and sub-classes obtained from fuzzy MEBN concept abstraction. For instance, declaration node class (*Node*) can be expressed as function Declaration (*Class (:Node)*), and it represents the entity and random variable of fuzzy MEBN. Fig. 2 shows Fuzzy PR-OWL ontology language (relation between classes and sub-classes) created with Protégé. Fuzzy PR-OWL ontology is based on fuzzy MEBN, and relevant classes and attributes of concepts like fuzzy random variable, fuzzy state, membership degree, and fuzzy rule are added based on PR-OWL2, to enhance the expressive force of uncertainty information.

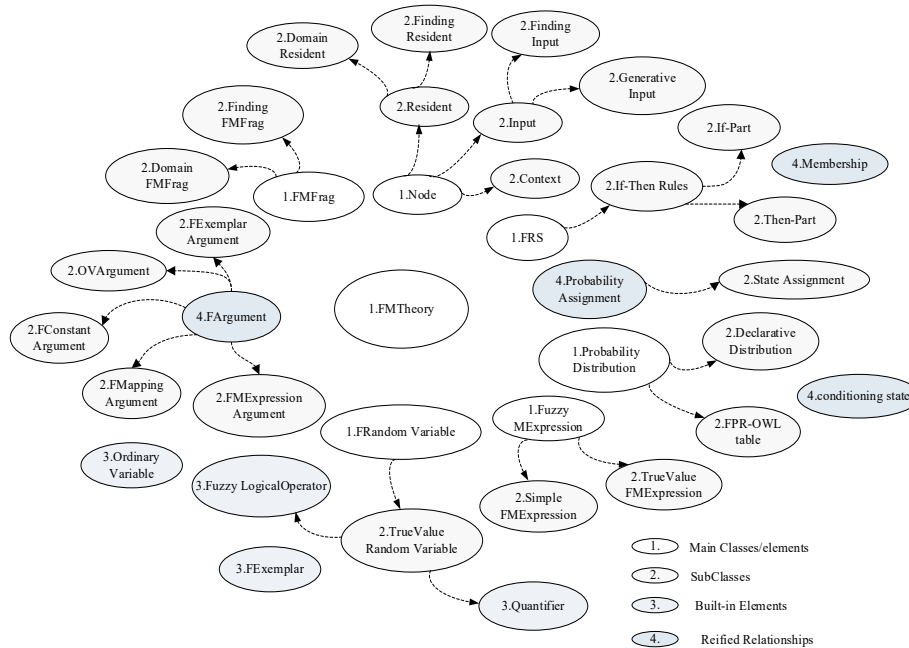


Figure 2: Fuzzy PR-OWL ontology elements

The basic model of Fuzzy PR-OWL is presented in Fig. 3, in which each piece of probability ontology has at least one *FMTheory* type, namely a label. It connects one group of *FMFrag*s to form an effective *FMTheory*. In Fuzzy PR-OWL grammar, the object attribute *hasFMFrag* is used to represent the connection. *FMFrag* is composed of multiple nodes, and each node type is a random variable. The biggest difference between Fuzzy PR-OWL and PR-OWL2 is, when the fuzzy state of random variable node is defined, the set of fuzzy rule reflecting the state of domain expert is used to define the membership degree of fuzzy state, and the object attribute *hasFRS* connects each node to one or several fuzzy rule sets. The non-conditional or conditional probability distribution is represented with the class *ProbabilityDistribution*, and is connected to their own nodes through the object attribute *hasProbDist*. The object attribute *hasFMExpression* connects every node to a logic expression or simple expression based on fuzzy first-order logic.

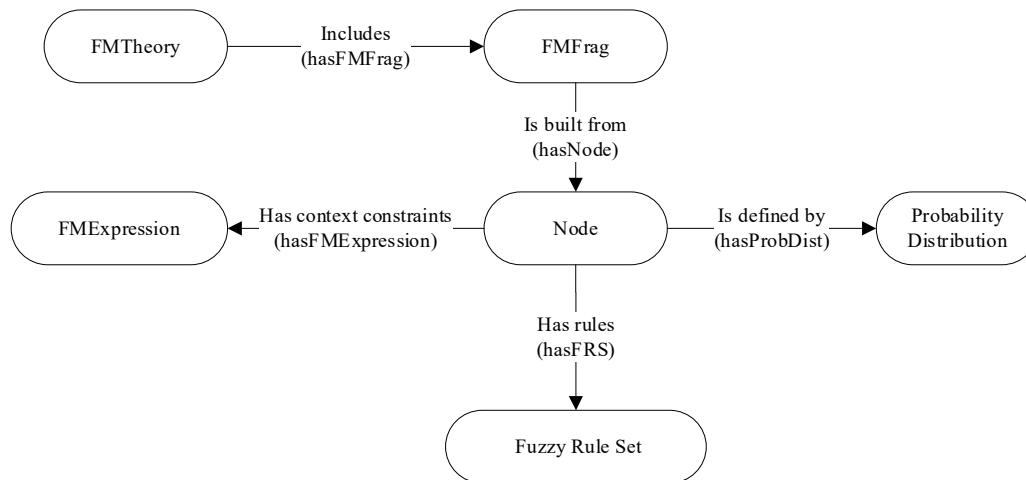


Figure 3: Basic model of Fuzzy PR-OWL

3 Reasoning based on fuzzy PR-OWL ontology

3.1 Reasoning framework

This section presents the fuzzy probability reasoning method based on Fuzzy PR-OWL ontology of fuzzy MEBN. Data fuzzification is first conducted and Situation-Specific Fuzzy Bayesian Network (SSFBN) of fuzzy MEBN is constructed. Then, fuzzy belief propagation based on fuzzy rule reasoning is conducted on the constructed network. The reasoning framework of Fuzzy PR-OWL ontology is mainly composed of four basic modules (as shown in Fig. 4), including ontology analysis and memory module, data manipulation module, reasoning module, and representation module.

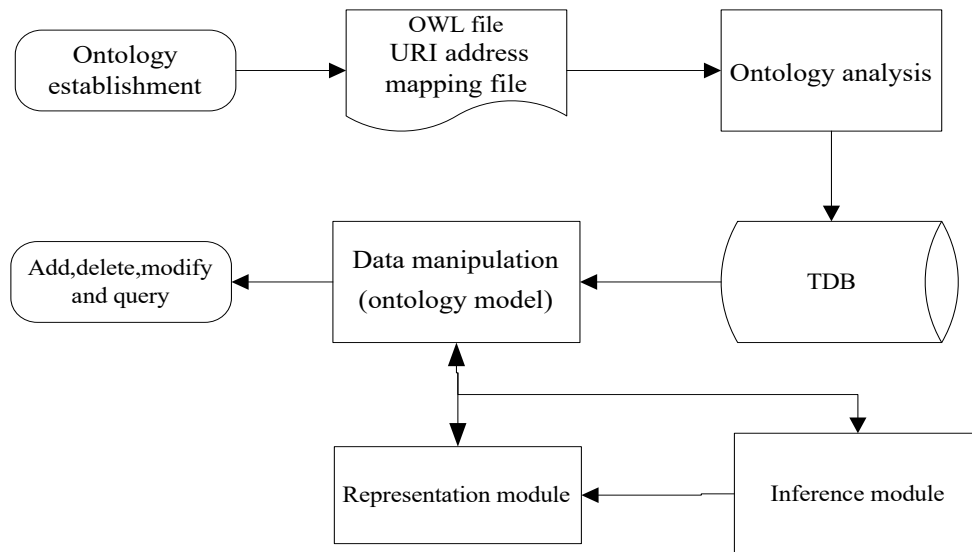


Figure 4: Reasoning framework diagram of Fuzzy PR-OWL

(1) Ontology analysis and TDB memory module

Ontology analysis and TDB memory module is realized based on Jena API. It reads, analyzes the ontology constructed by Protégé, and stores it in the RDF database of Jena [Noy, Ferguson and Musen (2000)]. The program first realizes the mapping between ontology URI and the corresponding ontology module file system address through URI address mapping file, and then reads various ontology module files via Java program or configuration files of Jena. In addition, data persistence is conducted in the RDF dataset. A built-in storage mode TDB officially recommended by Jena is used by memory module [Owens, Seaborne and Gibbins (2008)]. Characterized by fast reading speed and simple operation, this module can store billions of records at most, and support hundreds of parallel queries. Hence, it is quite suitable for the demand of ontology reasoning.

(2) Data manipulation module

Data manipulation module implements the adding, deletion, modification and querying operations by utilizing the query operation interface of TDB. The module maps the loaded TDB model to corresponding Java ontology or examples. Moreover, by integrating adding, deletion, modification and querying operations through the ontology statement of the TDB dataset, Java ontology implements RDF database operations through TDB operation interface.

(3) Reasoning module

Reasoning module contains an implementation of the Belief Propagation reasoning algorithm [Shafer and Shenoy (1990)] based on fuzzy probability. Both FuzzyOWL reasoning engine and API of UnBbayes [Carvalho, Laskey and Costa (2010)] framework is used, and reasoning and querying can be conducted on specific nodes.

Specifically speaking, it can handle knowledge base loading, fuzzy processing of data, SSFBN construction, and fuzzy probability reasoning functions.

(4) Representation module

Representation module handles GUI. It can manipulate the database through the graphical user interface, and display data variation in the database and reasoning results produced by the reasoning module.

3.2 Belief propagation algorithm based on fuzzy probability

In order to construct a Situation-Specific Bayesian Network (SSBN) based on knowledge and produce the minimum Bayesian network responding to query, common MEBN first sets prior distribution of target random variable and evidence random variable (also known as discovery random variable) as the condition. Then, posteriori distribution of target variable is reasoned through running the classic Bayesian network algorithm.

3.2.1 Data fuzzification

The algorithm proposed in this paper requires preprocessing the fuzzy concept or natural language variable gained from expert knowledge before establishing SSBN, so as to obtain the membership degree of fuzzy state of the variable. Some symbols and connotations involved in the algorithm need to be defined first.

Definition 1: the set of entity identifiers. In FMFrag F , the local node (or random variable) $\psi(\theta)$ has a set N_ψ composed of $|N_\psi|$ fuzzy states, and the corresponding fuzzy set is A . Then:

(1) $f(\mu_A, \alpha) = (\mu_A)^\alpha$, in which μ_A is the membership degree function of set A , that is $\mu_A: U \rightarrow [0,1]$; $\alpha \in [0,1]$ is the fuzzy factor. The function f is a general form of the fuzzy theory.

(2) The fuzzy state of ψ can be denoted by $\gamma_i^{\alpha_i}: i = 1, \dots, |N_\psi|$, in which $\gamma_i^{\alpha_i} \in N_\psi$, $\gamma_i \in \varepsilon$, and α_i represents the membership degree of state γ_i ; $\tilde{A} = \langle \alpha_1, \alpha_2, \dots, \alpha_{|N_\psi|} \rangle$ is the membership degree vector of ψ .

(3) The probability distribution vector corresponding to CPT of ψ is denoted by $\pi_\psi = \langle p_1, p_2, \dots, p_{|N_\psi|} \rangle$.

In FMFrag F , the local node ψ has multiple fuzzy states. For instance, if “height” is set as a random variable, there are three states (“short, medium, high”). “The person is relatively high” can be denoted by the fuzzy state $\langle \text{short}^0, \text{medium}^{0.2}, \text{high}^{0.8} \rangle$.

Random variables are discrete variables, and the membership degree of fuzzy state of these random variables should be obtained through fuzzification and discretization. The method to calculate the membership degree of fuzzy state of random variables mainly depends on the type of expert data (as shown in Fig. 5):

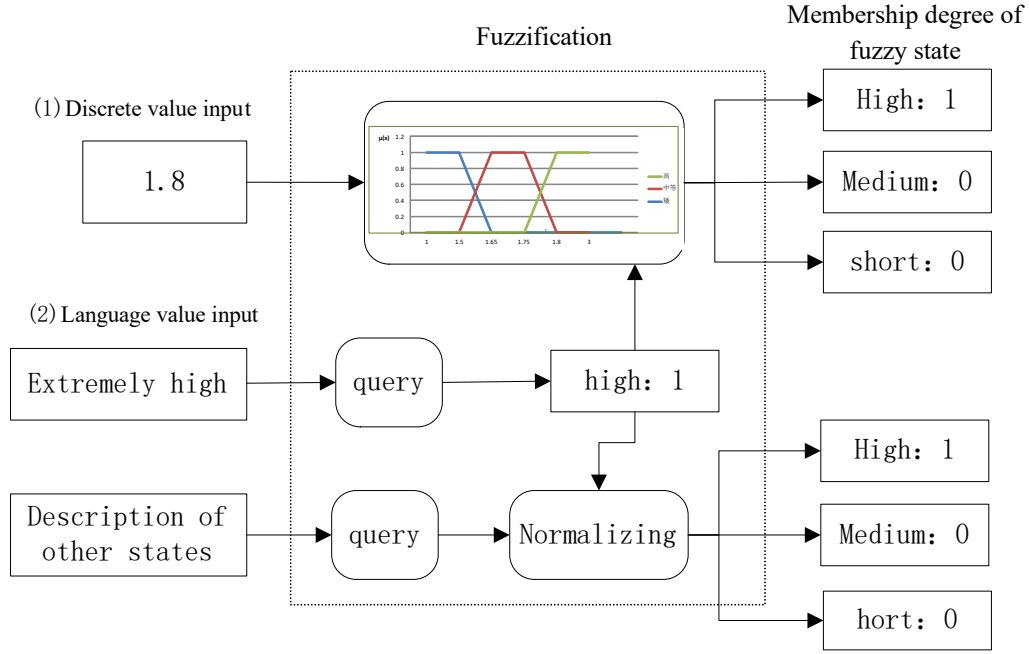


Figure 5: Schematic diagram of data fuzzification

(1) Discrete value input

As the specific data $x \in U$ is known, the membership degree can be calculated through the membership degree function $\mu_{A_i}(x)$ (retain one digit after the decimal place). For instance, the height 1.8 can be substituted into membership degree function $\mu_{short}(x)$, $\mu_{medium}(x)$, $\mu_{high}(x)$ defined in the domain of discourse $[1.0, 3.0]$, to determine the membership degree.

(2) Language value input

As the language's descriptor is known, the membership degree interval of corresponding extent description language variable can be determined using Tab. 1. Created from researches on degree adverbs of modern Chinese language and the division of Vasilios et al. for degree in English [Carvalho, Laskey and Costa (2010); Moura and Roisenberg (2015)], Tab. 1 gives the corresponding relation between language variable and membership degree suitable for Chinese context. In order to reduce the complexity, the average value was taken and the middle value in the interval was selected. Except the extreme case where 0 and 1 are taken in extremely low and high situations, the membership degree $\alpha_{A_{j:j \neq i}}$ of other states $\psi(\theta)$ can be obtained using one of the following methods:

- 1) If $A_{j:j \neq i}$ has language description only and no membership degree function, α_{A_j} can be obtained through the corresponding relation between language variable and membership degree;

2) If $A_{j:j \neq i}$ has membership degree function μ_{A_j} , x can be calculated through inverse function $x = \mu_{A_i}^{-1}(\alpha_{A_i})$, and the membership degree α_{A_j} can be obtained by substituting x into $\mu_{A_j}(x)$ and retaining one digit after the decimal place.

Table 1: Corresponding table of extent description language and fuzzy membership degree

Language tag	Similar expression	Corresponding membership degree	Value
Extremely low (weak)	The degree is extremely weak (hardly exists), incredibly low, and extremely low	0.0~0.1	0.0
Very low (weak)	The degree is very weak, awfully low, and especially low	0~0.2	0.1
Quite low (weak)	The degree is comparatively low, quite low, and not too low	0.1~0.3	0.2
Low (weak)	The degree is low	0.2~0.4	0.3
Slightly low (weak)	The degree is slightly low, a little low, and somewhat low	0.3~0.5	0.4
Medium	The degree is medium	0.4~0.6	0.5
Slightly high (strong)	The degree is slightly high, a little high, and somewhat high	0.5~0.7	0.6
High (strong)	The degree is high	0.6~0.8	0.7
Quite high (strong)	The degree is comparatively high and quite high	0.7~0.9	0.8
Very high (strong)	The degree is very high, awfully high, especially high, and not too high	0.8~1	0.9
Extremely high (strong)	The degree is incredibly high and extremely high	0.9~1	1

To further explain 2), two examples are given in the following. Suppose the random variable “weather” has two states {sunny, cloudy}, and the current description is comparatively cloudy. According to Tab. 1, the membership degree should be (0.2, 0.8). If the sum of membership degrees of states is not equal to 1, standardization should be done to make the sum equal to 1. By taking “height” as an example, if we just know that someone is “very high”, the fuzzy membership degree of state “high” can be 0.9, and the rough height value x can be calculated through the inverse function. The membership degrees of “short” and “medium” can then be calculated through $\mu_{short}(x)$ and $\mu_{medium}(x)$, that is, if $\alpha_{A_{high}} = 0.9$, then $\alpha_{A_{short}} = 0$, $\alpha_{A_{medium}} = 0.1$.

3.2.2 Fuzzy IF-THEN rule

In formal knowledge representation of reasoning system, IF-THEN rule is one of the basic approaches to represent the knowledge. This section will further expound the reasoning process of IF-THEN rule with the fuzzification.

The general format of fuzzy IF-THEN rule is “IF x is A_i , THEN y is B_j ”, which decides the causal relationship of language variable [Zadeh (1968)]. In this rule, x is the input variable, y indicates the output variable, and A_i and B_j represent the natural language values corresponding to the variables. They are represented with the fuzzy set determined by the model input.

In IF-THEN rule of fuzzy MEBN, y and x represent the random variable $\psi(\theta)$ and the language variable of its parent node $\pi_o(\psi(\theta))$ respectively. Suppose that $A_o^{(i)}$ and B_o are the natural language values of variables; $\widetilde{A_o^{(i)}}$ and $\widetilde{B_o}$ represent the domain of discourse $X_o^{(i)}$ of rule o and the fuzzy set in Y_o respectively. The IF-THEN rule can be simply represented by the following if-then rule statements:

$$\begin{aligned} \text{IF } x_1 \text{ is } A_1^{(1)} \text{ AND } \dots \text{AND } x_N \text{ is } A_1^{(N)} \text{ THEN } y \text{ is } B_1 \dots \\ \text{IF } x_1 \text{ is } A_o^{(1)} \text{ AND } \dots \text{AND } x_N \text{ is } A_o^{(N)} \text{ THEN } y \text{ is } B_o \dots \\ \text{IF } x_1 \text{ is } A_o^{(1)} \text{ AND } \dots \text{AND } x_N \text{ is } A_o^{(N)} \text{ THEN } y \text{ is } B_o \dots \end{aligned} \quad (4)$$

where O is the number of rules; N is the number of input variables; $A_o^{(1)}, \dots, A_o^{(N)}$ and B_o indicate the language values of input random variables x_1, \dots, x_N and output random variable y of IF-THEN rule o ; the fuzzy sets can be obtained through the method of the previous section, and denoted by $\widetilde{A_o^{(i)}} = \{\alpha_i^{(i)} | i = 1, \dots, |N_{\pi_i(\psi)}|\}$ and $\widetilde{B_o} = \{\beta_j | j = 1, \dots, |N_\psi|\}$ respectively, in which $\alpha_i^{(i)}$ and β_j are the membership degrees of fuzzy states in fuzzy sets $\widetilde{A_o^{(i)}}$ and $\widetilde{B_o}$ separately, $|N_{\pi_i(\psi)}|$ is the number of states of input variable x_i , and $|N_\psi|$ is the number of states of output variable y .

Table 2: IF-THEN reasoning

Major premise (rule): IF x_1 is $A_1^{(1)}$ AND ...AND x_N is $A_1^{(N)}$ THEN y is $B_1 \dots$
IF x_1 is $A_o^{(1)}$ AND ...AND x_N is $A_o^{(N)}$ THEN y is $B_o \dots$
Minor premise (fact): x_1 is $A^{(1)*}$ and AND ...AND x_N is $A^{(N)*}$
Consequent (conclusion): y is B^*

As shown in Tab. 2, under the condition that $\widetilde{A_o^{(i)*}}$ is given, according to Eq. (2), conclusion $\widetilde{B^*}$ can be calculated by Eq. (5):

$$\begin{aligned} \mu_{\widetilde{B^*}}(y) \\ = \bigvee_{x_i \in X_i, y \in Y} \left\{ \left[\bigwedge_N \mu_{\widetilde{A_o^{(i)*}}}(x_i) \right] \wedge \left[\bigvee_O \mu_{\widetilde{B_o}}(x_1, \dots, x_N, y) \right] \right\} \bigvee \left\{ \bigvee_{x_i \in X_i, y \in Y} \left[\bigwedge_N \mu_{\widetilde{A_o^{(i)*}}}(x_i) \wedge \mu_{\widetilde{B_o}}(x_1, \dots, x_N, y) \right] \right\} \\ = \mu_{\widetilde{B_1}}(y) \vee \dots \vee \mu_{\widetilde{B_o}}(y) \end{aligned} \quad (5)$$

where $\mu_{\widetilde{R}_o}(x_1, \dots, x_N, y) = (\bigwedge_N \mu_{\widetilde{A}_o^{(i)}}(x)) \vee \mu_{\widetilde{B}_o}(y)$ is the membership of \widetilde{R}_o , the minimization of Cartesian product of membership degrees of major premises and conclusions in the fuzzy rule o .

Each fuzzy MEBN node contains fuzzy and probabilistic information at the same time. Besides the IF-THEN rule, the Fuzzy PR-OWL ontology can also show the conditional probability distribution. When the conditional probability distribution is given, the fuzzy probability can be calculated through the following reasoning.

The joint probability of $A^{(1)}, \dots, A^{(N)}$ is the probability of fuzzy event $A = A^{(1)} \times \dots \times A^{(N)}$, and can be calculated by Eq. (6):

$$P(X \text{ is } A) = P((x_1 = A^{(1)} \cap \dots \cap (x_N = A^{(N)})), X = [x_1, \dots, x_N] \quad (6)$$

The probability of fuzzy event A_o can be calculated by Eq. (7):

$$P(A_o) = \sum_{x_1, \dots, x_N \in X_1 \times X_2 \times \dots \times X_N} p(x_1, \dots, x_N) \cdot \cup_o [T(\widetilde{A}_o^{(1)} \times \dots \times \widetilde{A}_o^{(N)})] \quad (7)$$

where T is the t-norm operator of weak conjunction calculated by the minimization process, \times is the Cartesian product, and \cup is the maximization operator.

When the fact value $A^* = A_o^{(1)*}, \dots, A^{(N)*}$ of fuzzy event $A = A^{(1)} \times \dots \times A^{(N)}$ and the prior probability are given, the marginal probability of child node B can be calculated by Eq. (8):

$$\begin{aligned} P(B) &= \left[\sum_{x_1 \in X_1} \dots \sum_{x_N \in X_N} p(x_1, \dots, x_N, y) \right] \cdot \widetilde{B}^*(y) \\ &= \{ \sum_{x_1 \in X_1} \dots \sum_{x_N \in X_N} [\prod_N p(x_i) p(y|x_i)] \} \cdot \mu_{\widetilde{B}^*}(y) \end{aligned} \quad (8)$$

3.2.3 Fuzzy belief propagation algorithm

In the BN reasoning algorithm, if Bayesian Network is small, simple marginal summation can meet the requirement, but the practical problem often involves a large scale. The overall operation complexity and data will show exponential growth. If the Belief Propagation algorithm is used to solve such a network problem, the operation complexity will only have a linear correlation with the number of nodes. Therefore, the Belief Propagation algorithm plays an increasingly important role when dealing with large-scale Bayesian network reasoning problems.

The Clique Tree algorithm, also known as the Junction Tree algorithm, was proposed by Lauritzen et al. [Lauritzen and Spiegelhalter (1988)]. The Clique Tree algorithm is the fastest Bayesian reasoning algorithm at present. The algorithm first converts the Bayesian network into a clique tree, and then performs calculation through message passing. The message will be broadcasted to every node of the clique tree to ensure consistency.

In the Clique Tree Belief Propagation algorithm, every clique tree is set as a node to send message to neighbor nodes. Let clique tree T be composed of clique C_1, C_2, \dots, C_k . Each $\phi \in \Phi$ is assigned to clique $\alpha(\phi)$, and the initial potential energy of C_j is defined by Eq. (9):

$$\psi_j(C_j) = \prod_{\phi: \alpha(\phi)=j} \phi \quad (9)$$

In the equation, local evidence is corresponding to the prior probability or conditional probability of every node, i.e., $P(x_i|\pi(x_i)) = \phi_{x_i}(x_i, \pi(x_i))$. For instance, $P(a) = \phi_a(a)$, $P(d|b) = \phi_d(d, b)$, $P(c|a, b) = \phi_c(c, a, b)$.

For clique C_i , β_i is initialized as ψ_i according to Eqs. (9) and (3):

$$\widetilde{\psi}_j(C_j) = \prod_{\phi:\alpha(\phi)=j} \phi \cdot \bigcup_{\mu:\alpha(\mu)=j} [T(\Gamma_\mu^* \times \widetilde{R}_\mu)] \quad (10)$$

where $\Gamma_\mu^* = \bigcup_{x_i \in X_i} [T(\widetilde{\mu}_{x_1}^* \times \dots \times \widetilde{\mu}_{x_N}^*)]$, $\widetilde{R}_\chi = U[T(\widetilde{\mu}_{x_1} \times \dots \times \widetilde{\mu}_{x_N}), \widetilde{\mu}_y]$.

Clique C_i multiplies all other neighbor messages with the initial clique potential energy, adds up all variables except variables in the cut set of C_i and C_j , and sends the result to C_i . Clique C_i node transmits the updated messages to other nodes after receiving messages of its neighbor nodes. The ultimate potential energy of C_i can be obtained by Eq. (11):

$$\widetilde{\beta}_i = \widetilde{\psi}_i \cdot \prod_{k \in Nb_i} \widetilde{m}_{ki} \quad (11)$$

The messages of C_i and C_j are calculated by Eq. (12):

$$\widetilde{m}_{ij} = \frac{\sum_{C_i - S_{ij}} \widetilde{\beta}_i}{\widetilde{m}_{ji}} \quad (12)$$

This equation requires \widetilde{m}_{ij} to be nonzero. Otherwise, the result will be meaningless. As for the similar situation, when there is a “0” in the message \widetilde{m}_{ki} sent by neighbor nodes of C_i , the ultimate potential energy has to be zero. Park and Darwiche proposed an algorithm to resolve this problem by introducing a pair of figures (z, b) to indicate whether zero exists [Park and Darwiche (2003)]:

$$real(z, b) = \begin{cases} z & b = f \\ 0 & other \end{cases} \quad (13)$$

where (1) b is Boolean number, indicating whether “0” exists in all \widetilde{m}_{ki} . If “0” exists, $b=t$ (true); otherwise, $b=f$ (false); (2) z is the product of all messages whose value is not zero, i.e., $\prod_{m_{ki} \neq 0} \widetilde{m}_{ki}$.

In addition, the zero-perception table can also identify zero, and maps variable example into data that can detect zero. Suppose that Ψ is the table with zero detection and Φ is the standard table without zero detection, $real(\Psi)$ is the standard table containing results obtained via $real(z, b)$ in table Ψ .

Using Eq. (12), the message \widetilde{m}_{ij} from C_i to another clique C_j and ultimate potential energy $\widetilde{\beta}_i$ of C_i can be calculated. Message \widetilde{m} is transmitted between the leaf node and the root node. Every node will multiply its belief with newly received fuzzy message and divide the result by fuzzy message received from the same nodes last time. $\widetilde{\Psi}$ is defined as the table with zero detection of $\widetilde{\psi}_i$, and $\widetilde{\Phi}$ is the standard table of \widetilde{m}_{ij} . The message sent from C_i to C_j is calculated by Eq. (14):

$$\begin{aligned} \widetilde{\Psi}_{temp} &\leftarrow \sum_{C_i \setminus S_{ij}} \widetilde{\Psi}_i \\ \widetilde{\Psi}_j &\leftarrow \widetilde{\Psi}_j \left(\frac{\widetilde{\Psi}_{temp}}{\widetilde{\Phi}_{ij}} \right) \end{aligned} \quad (14)$$

$$\tilde{\Phi}_{ij} \leftarrow \text{real}(\tilde{\Psi}_{temp})$$

The specific steps of Belief Propagation algorithm based on fuzzy probability are as follows:

(1) Initialization

The fuzzy set of (input or output) variable is obtained through fuzzification according to the language value of variable, and the generated SSFBN is divided. For every clique C_i , $\tilde{\beta}_i$ is calculated to initialize $\tilde{\psi}_i$ using Eq. (10). The parameter of $\text{real}(z, b)$ is initialized to $(1, f)$.

(2) Information transfer

The message is transferred from the leaf node to the root node, and then transferred from the root node to the leaf node. In all directions, every cut set $S_{i,j}$ will consider $\Phi_{i,j}$ as the previous message transmitted along the edge $(i-j)$. This message can prevent repeated calculation, and every node will immediately update its belief after receiving messages from other nodes. It will multiply its belief with newly received message and divide the result by the message received from the same nodes last time. Here real function is used to avoid meaningless message.

The algorithm of Fuzzy-Probability Belief Propagation (FPBP)

Procedure JTree- FPBP (Φ, T)// Φ : set of factors, T : clique tree in Φ

- 1: Initialize-JTree(T)
- 2: When a clique without information exists in T
- 3: Choose $(i - j) \in \varepsilon_T$
- 4: MessagePropagation(i, j)//Message passing
- 5: Return to $\{\tilde{\Psi}_i\}$

Procedure Initialize-JTree(T)//Initialize the clique method

- 1: For each clique C_i
- 2: $\tilde{\Psi}_i = \prod_{\phi: \alpha(\phi)=i} \phi \cdot \bigcup_{\mu: \alpha(\mu)=i} [T(\Gamma_\mu^* \times \tilde{R}_\mu)]$
- 3: For each edge $(i - j) \in \varepsilon_T$
- 4: $\tilde{\Phi}_{ij} \leftarrow 1$
- 5: The parameter of $\text{real}(z, b)$ is initialized to $(1, f)$

Procedure MessagePropagation(i, j)//Message passing method: i sending clique, j receiving clique

- 1: $\tilde{\Psi}_{temp} \leftarrow \sum_{C_i \setminus S_{ij}} \tilde{\Psi}_i$
- 2: $\tilde{\Psi}_j \leftarrow \tilde{\Psi}_j(\tilde{\Psi}_{temp}/\tilde{\Phi}_{ij})$
- 3: $\tilde{\Phi}_{ij} \leftarrow \text{real}(\tilde{\Psi}_{temp})$

3.3 Establishment and reasoning of Situation-Specific Fuzzy Bayesian Network (SSFBN)

The premise of MEBN reasoning is the generation of Situation-Specific Bayesian Network, i.e., SSBN. The minimum BN that satisfies the query request is first obtained through MTheories of domain-specific knowledge and a group of observed results, and then the reasoning algorithm of Bayesian network is used to calculate the posterior probability of searching for the set of target random variables gathered from observation evidence.

The algorithm of Fuzzy MEBN is a simple bottom-up algorithm and starts from the finite set containing target random variables and finite set of random variables gathered from observation evidence. The two sets are combined to build an approximate SSFBN. In the approximate SSFBN, if one node is the parent of another node or the context variable of its superior FMFrag, an edge between the two nodes is added. In each step, the algorithm builds a new approximate SSFBN by adding evidence nodes and instantiates superior FMFragments of random variables in the query sets and their ancestor nodes. The result random variables are added to the query set, any node irrelevant to query is removed, and the result sets of random variables are combined to create a new approximate SSFBN. The above process repeats until the approximate SSFBN remains unchanged or the stopping condition is met. If the algorithm has no stopping condition, the result of the algorithm is either an accurate response or a warning that the evidence information is inconsistent when the construction of SSFBN is done. If the algorithm does not stop, it produces many approximate SSFBNs that can generate correct query responses. Generally speaking, there is no evidence of finite length consistent with the query fragment, but inconsistent query fragments can be detected when SSFBN is constructed.

To minimize the final SSFBN and improve the reasoning efficiency, while constructing SSFBN, evidence node, target node and internal node are retained while redundant node, barren node and d-separated node irrelevant to the ultimate result or with a small influence are removed.

Suppose that E is the set of observed evidences, Q is the set of queries, B is Bayesian network, and V is the set of nodes. The construction process of SSFBN is consists of the following steps:

(1) Initialization

The set of queries Q is set to be the union set of random variables of target nodes and evidence nodes; the random variable example R_0 is set to be Q and the set of evidences E is set to be null; the maximum number N_0 of states of every random variable is a positive integer user specified.

(2) SSFBN Construction

Suppose that the current SSFBN B_i includes nodes in R_i , and all edges are defined in influential configurations. Remove any isolated nodes from B_i , and conduct d-separation from target nodes to evidence nodes, without updating the marginal distribution of nuisance nodes.

Whenever a new evidence is added to E , a new SSFBN is constructed and any unnecessary node, such as barren node $BNode(v)$ or worthless node $NNode(v)$, including d-separated node connected to target node and marginal distribution node, is removed. Later, the FPBP algorithm in Section 3.2.3 can be used to calculate local

distribution. Finally, the FBN algorithm is run according to Fuzzy Belief Propagation on the manicured network B according to Q and E . The goal of this algorithm is to construct the Bayesian network of context that meets the consistency condition.

In the SSFBN construction algorithm, the situation-specific random variable node is generated based on query variable Q and evidence set E . The context node of every random variable has passed satisfiability test. If the condition is met, this node and its parent node will be created.

SSFBN construction algorithm

Procedure Generate-SSFBN(Q, E)

```

1:  $V \leftarrow Q \cup E$  //Set of all query and evidence nodes
2:  $G \leftarrow \emptyset$ 
3: while  $V \neq \emptyset$  do
4:    $v \leftarrow V - \{v\}$ 
5:   if Is-Satisfied( $v$ ) then //Whether the consistency condition is met
6:      $G \leftarrow \text{Create-Node}(v)$ 
7:      $\Pi \leftarrow \text{Get-Parents}(v)$ 
8:     for all  $\pi \in \Pi$  do
9:        $G \leftarrow \text{Create-Node}(v)$ 
10:     $V \leftarrow V - \pi$ 
11: return  $G$ 

```

FBN reasoning algorithm

Procedure FBN-Reasoning(B, Q, E)

```

1: Construct clique tree  $T$  in  $B$  and gain the set of factors  $\Phi$ 
2:  $R_i \leftarrow \text{JTree-FPBP}(\Phi, T)$  //Belief propagation
3: if all  $((e \in E) \wedge (P(e) = 0))$  then
4:   Report inconsistency of evidence random variables
5: else
6:   return  $R_i$ 

```

(3) Local distribution construction

We set local distribution in B_i , modify the local distribution and restrict random variables to N_i possible values, and estimate the influence of random variables not listed. Besides, the calculation should not exceed K_i steps.

(4) Reasoning

Given evidence random variables, we calculate the local distribution of target random variables via the FPBP algorithm in Section 3.2.3. We run the Fuzzy Belief

Propagation algorithm on the manicured network B according to Q and E , and calculate the conditional probability distribution of non-query nodes. Inconsistency of evidences is reported if the evidence random variable has probability 0, and if the query contents are consistent, the random variable of query set of the current SSFBN, R_i , is outputted.

(5) Example enumeration and approximate parameter updating

If the stopping condition is met, output R_i . Otherwise, the parent nodes of other random variables of R_i is added. As the distribution might be changed when extra parent nodes are added, N_i and K_i are added before going to Step 2.

4 Experiment and analysis

We used the tenfold cross validation method to evaluate the veracity of the FPBP algorithm by comparing it with the classic Belief Propagation algorithm (BP). Our test dataset is Iris, which consists of 50 samples from each of three species of Iris (*Iris virginica*, *Iris setosa*, and *Iris versicolor*) and each sample has 4 attributes: *sepal length*: 4.3 cm~7.9 cm; *sepal width*: 2.0 cm~4.4 cm; *petal length*: 1 cm~6.9 cm; *petal width*: 0.1 cm~2.5 cm.

The goal of our experiment was to predict the specie of an Iris flower from four attribute values including sepal length, sepal width, petal length, and petal width, as shown in Fig. 6.

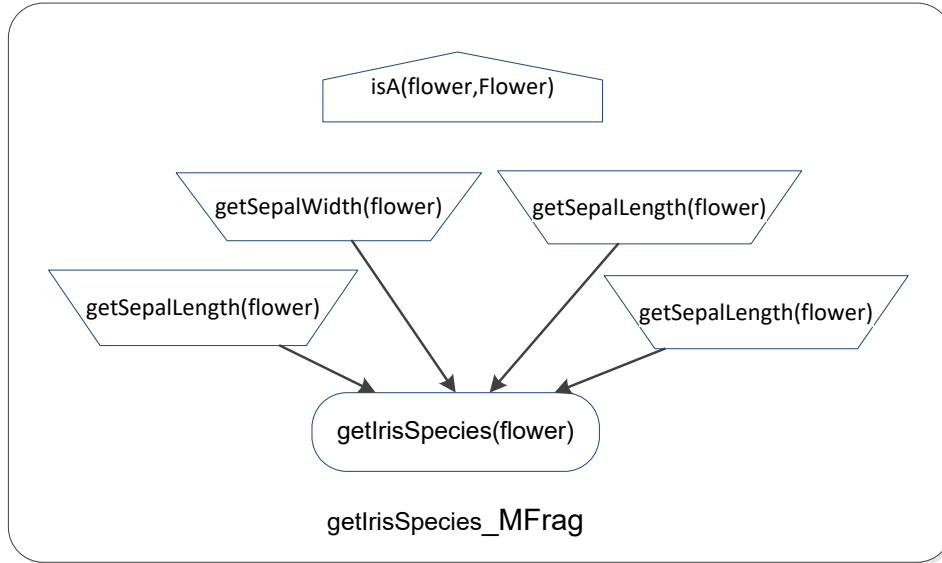


Figure 6: Species fragment of Iris

In the experiment, random variables, such as sepal length *getSepalLength (flower)*, are classified into three categories: short, medium and long. For the training data, the values in the sepal length column are first sorted in ascending order and the resulting sequence is $b_{i,1}, b_{i,2}, \dots, b_{i,n}$ where the first subscript i is the column index number, the second subscript represents the index in the sequence, and n is the size of training set. Then the sequence is equally divided into three parts and the boundaries of three categories can be determined,

i.e., short ($< b_{1,n/3}$), medium ($b_{1,n/3}, b_{1,2n/3}$) and long ($> b_{1,2n/3}$). Similar method can also be used to classify other random variables, as shown in Tab. 3.

Table 3: Category table of random variables of Iris

Random variable	Category 1	Category 2	Category 3
getSepalLength(flower)	Short($<b_{1,n/3}$)	Medium($b_{1,n/3} \sim b_{1,2n/3}$)	Long($>b_{1,2n/3}$)
getSepalWidth(flower)	Short($<b_{2,n/3}$)	Medium($b_{2,n/3} \sim b_{2,2n/3}$)	Long($>b_{2,2n/3}$)
getPetalLength(flower)	Short($<b_{3,n/3}$)	Medium($b_{3,n/3} \sim b_{3,2n/3}$)	Long($>b_{3,2n/3}$)
getPetalWith(flower)	Short($<b_{4,n/3}$)	Medium($b_{4,n/3} \sim b_{4,2n/3}$)	Long($>b_{4,2n/3}$)
getIrisSpecies(flower)	Iris virginica	Iris setosa	Iris versicolor

By aiming at the characteristics of Iris dataset, the triangle membership degree function was used to calculate the membership degree from the values of random variables like petal length. Fig. 7 shows the membership degree function of the Iris dataset, in which a is equivalent to the minimum value of random variable, b is the median $b_{i,n/2}$ of the attribute sequence, and c is the maximum value of this attribute. For instance, a , the membership degree of petal length random variable, has the minimum value (1 cm), b is the median ($b_{3,n/2}$) of the petal length sequence $b_{3,1}, \dots, b_{3,n/2}, \dots, b_{3,n}$, and c is the maximum value (6.9 cm). Suppose that the petal length of a specific line of data in training set is x . The membership degree vector can be calculated by using the membership degree function.

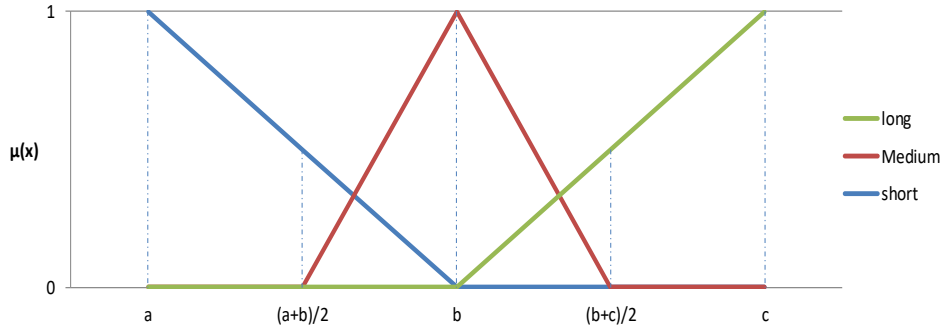


Figure 7: Membership degree function of Iris

We chose the tenfold cross validation method, which is the veracity test method of the common algorithm, to evaluate our proposed solution. We first create 10 sample data sets from the Iris dataset. 9 of them were set as the training data in turn and 1 was used as the test data. The average classification accuracy (or error rate) of 10 runs was used to measure the performance of the algorithm.

Tab. 4 shows the tenfold cross validation results of the proposed FPBP algorithm (retain two digits after the decimal point). Meanwhile, we applied the same validation process to the BP algorithm using the same 10 sample data sets and compared the performance of two algorithms. As shown in Tab. 5, the average classification accuracy of FPBP is 89.2%, which is nearly 34% higher than the average accuracy of BP (66.7%). Obviously, FPBP, the fuzzy probability algorithm that adds fuzzy membership degree calculation, has higher

accuracy than the BP algorithm that simply calculates the probability, which demonstrates the veracity and validity of our algorithm.

Table 4: Experimental results of tenfold cross validation of FPBP

	1	2	3	4	5	6	7	8	9	10
Sampling 1	86.67%	100%	93.33%	100%	100%	80%	80%	73.33%	93.33%	86.67%
Sampling 2	86.67%	93.33%	80%	86.67%	86.67%	93.33%	80%	93.33%	93.33%	93.33%
Sampling 3	93.33%	100%	80%	80%	86.67%	86.67%	100%	93.33%	80%	93.33%
Sampling 4	93.33%	93.33%	86.67%	80%	86.67%	86.67%	93.33%	100%	73.33%	93.33%
Sampling 5	93.33%	93.33%	86.67%	93.33%	73.33%	86.67%	93.33%	100%	93.33%	86.67%

Table 5: Comparison of average classification accuracy between FPBP and BP

	Sampling 1	Sampling 2	Sampling 3	Sampling 4	Sampling 5	average
FPBP	89.33%	88.67%	89.33%	88.67%	90%	89.2%
BP	66.7%	66.7%	66.7%	66.7%	66.7%	66.7%

The time complexity of the BP algorithm on the Iris dataset is $T(n, s, p) = T_1(n) * T_3(p) * T_2(s) = O(f(n) * \phi(p) * g(s))$, in which n is the size of dataset, p represents the number of random variables like petal length, and s is the number of variable states. By adding matrix multiplication related to number of categories, the time complexity of FPBP is as following:

$$T(n, s, p) = T_1(n) * (T_2(s) * T_3(p) + T_2(s) * T_2(s)) = O(f(n) * g(s) * \max(g(s), \phi(p))) \quad (15)$$

In practice, the size of data is often greater than the number of attributes and the number of categories, and the number of attributes is greater than the number of categories. Therefore, compared with the BP algorithm, the increase in time complexity of the FPBP algorithm is still within a reasonable range.

5 Conclusions and future work

In this paper, by integrating the Belief Propagation algorithm of Bayesian network with the fuzzy probability theory, a fuzzy belief propagation algorithm is proposed and applied to the representation and reasoning framework of Fuzzy PR-OWL ontology. Besides, the framework from ontology analysis to reasoning model was created for representation and reasoning of fuzzy probability knowledge. According to the tenfold cross validation experiment results and the time complexity analysis, the FPBP algorithm has much higher classification accuracy than the BP algorithm without increasing the time complexity too much.

Although this paper has made some progress in reasoning study about fuzzy probability ontology, due to increasing complication and expansion of ontology, researchers in the field of uncertainty reasoning of ontology are still faced with many challenges. For example, the veracity of our algorithms highly depends on the selection of the membership degree function, and how to select appropriate membership degree function can be further explored. In addition, although our algorithm can handle most fuzzy probability reasoning

problems of various scales, the efficiency can be further improved when the node number reaches a certain scale.

Acknowledgments: The authors are grateful to the editors and reviewers for their suggestions and comments. This work was supported by National Key Research and Development Project (2018YFC0824400), National Social Science Foundation project (17BXW065), Science and Technology Research project of Henan (1521023110285), Higher Education Teaching Reform Research and Practice Projects of Henan (32180189).

References

- Berg, J. V. D.; Kaymak, U.; Rui, J. A.** (2013): Conditional density estimation using probabilistic fuzzy systems. *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 5, pp. 869-882.
- Calegari, S.; Ciucci, D.** (2007): Fuzzy ontology, fuzzy description logics and fuzzy-OWL. *International Workshop on Fuzzy Logic and Applications: Applications of Fuzzy Sets Theory*, pp. 118-126.
- Carvalho, R.; Laskey, K.; Costa, P.** (2010): UnBBayes: modeling uncertainty for plausible reasoning in the Semantic Web. *Semantic Web*.
- Carvalho, R. N.; Laskey, K. B.; Costa, P. C.** (2013): Uncertainty reasoning for the Semantic Web II. *Berlin Heidelberg Springer*, pp. 1-18.
- Chantas, G.; Kitsikidis, A.; Nikolopoulos, S.** (2014): Multi-entity Bayesian networks for knowledge-driven analysis of ICH content. *Computer Vision-ECCV 2014 Workshops*, pp. 355-369.
- Costa, P. C.; Laskey, K. B.; Laskey, K. J.** (2008): PR-OWL: a Bayesian ontology language for the Semantic Web. *Berlin Heidelberg Springer*, pp. 88-107.
- Ding, Z.; Peng, Y.; Pan, R.** (1970). BayesOWL: uncertainty modeling in Semantic Web ontologies. *Soft Computing in Ontologies and Semantic Web*, pp. 3-29.
- Golestan, K.; Karray, F.; Kamel, M. S.** (2013): High level information fusion through a fuzzy extension to multi-entity Bayesian networks in vehicular Ad-Hoc networks. *International Conference on Information Fusion*, pp. 1180-1187.
- Golestan, K.; Karray, F.; Kamel, M. S.** (2014): Fuzzy multi-entity Bayesian networks: a model for imprecise knowledge representation and reasoning in high-level information fusion. *IEEE International Conference on Fuzzy Systems*, pp. 1678-1685.
- Heckerman, D.** (1997): Bayesian networks for data mining. *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 79-119.
- Lauritzen, S. L.; Spiegelhalter, D. J.** (1988): Local computations with probabilities on graphical structures and their applications to expert systems. *Proceedings of the Royal Statistical Society B*, vol. 50, no. 2, pp. 154-227.
- Noy, N. F.; Fergerson, R. W.; Musen, M. A.** (2000): The knowledge model of Protégé-2000: combining interoperability and flexibility. *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 17-32.

- Owens, A.; Seaborne, A.; Gibbins, N.** (2008): Clustered TDB: a clustered triple store for Jena. <http://eprints.ecs.soton.ac.uk/16974/>.
- Park, J. D.; Darwiche, A.** (2003): Morphing the Hugin and Shenoy-Shafer architectures. *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp.149-160.
- Shafer, G. R.; Shenoy, P. P.** (1990): Probability propagation. *Annals of Mathematics and Artificial Intelligence*, vol. 2, no. 1, pp. 327-351.
- Yang, Y.** (2007): A framework for decision support systems adapted to uncertain knowledge. *Karlsruhe: Universitaet Fridericiana zu Karlsruhe*.
- Yang, Y.; Calmet, J.** (2006): OntoBayes approach to corporate knowledge. *International Conference on Foundations of Intelligent Systems*, pp. 274-283.
- Zheng, Z. Y.; Liu, Z. Y.; Li, L.** (2016): A fuzzy MEBN ontology language based on OWL2. *The Second International Conference on Fuzzy Systems and Data Mining*, pp. 71-80.
- Zadeh, L. A.** (1968): Probability measure of fuzzy events. *Journal of Mathematical Analysis & Applications*, vol. 23, no. 2, pp. 421-427.