International Journal of Advanced Smart Convergence Vol.8 No.2 94-101 (2019)
http://dx.doi.org/10.7236/IJASC.2018.8.2.94

IJASC 19-2-11

Belief Function Retraction and Tracing Algorithm for Rule Refinement

Gye Sung Lee

Dept. of Software, Dankook University, Korea
gslee@dankook.ac.kr

Abstract

Building a stable knowledge base is an important issue in the application of knowledge engineering. In this paper, we present an algorithm for detecting and locating discrepancies in the line of the reasoning process especially when discrepancies occur on belief values. This includes backtracking the rule firing from a goal node of the rule network. Retracting a belief function allows the current belief state to move back to another belief state without the rule firing. It also gives an estimate, called contribution measure, of how much the rule has an impact on the current belief state. Examining the measure leads the expert to locate the possible cause of problem in the rule. For non-monotonic reasoning, the belief retraction method moves the belief state back to the previous state. A tracing algorithm is presented to identify and locate the cause of problem. This also gives repair suggestions for rule refinement.

Keywords: *Dempster-Shafer theory, belief function, contribution measure, rule retraction, knowledge refinement*

1. Introduction

To achieve reliable performance in knowledge based reasoning, the first task is to build reliable knowledge bases. In most cases, the knowledge base is built in two stages: knowledge acquisition and knowledge refinement. The first stage is to gather information for constructing the initial knowledge base and the second stage is to perform iterative refinement of this knowledge base to address incompleteness, inconsistency, and performance problems [1]. In the first stage, different tools are used to extract knowledge from experts. In the second phase, the human expert wants to update and change the knowledge base. The expert adds more knowledge to widen the scope of knowledge base or simply makes up incomplete knowledge. This phase can be referred to as incremental knowledge acquisition task [2-4]. During the course of this process, the expert uses test cases to establish the correctness of knowledge base. This task of identifying and correcting faulty knowledge or discovering missing knowledge is called knowledge refinement [2, 5].

Dealing with incompleteness and inconsistency of the knowledge base is looked upon as an incremental refinement process, where existing knowledge base is extended and refined by running test cases. This process is divided into three steps: detection and identification of possible discrepancies in the knowledge base,

Manuscript Received: Apr. 10, 2019 / Revised: Apr. 17, 2019 / Accepted: Apr. 22, 2019
Corresponding Author: gslee@dankook.ac.kr
Tel: +82-10-3344-7529, Fax: +82-31-8005-3223
Dept. of Software, Dankook University, Korea

suggestion of possible repairs, and validation of candidate repairs. Discrepancy detection requires identification of the problem in system behavior, and then possible causes for this discrepancy are pursued. There are two ways to detect discrepancies and suggest repairs: either manually by an expert or automatically by the built-in detection tool implemented in the system. In interactive systems, the expert takes the initiative in critiquing the performance of the system. For this type of system, a list of actions performed by the system during consultation is logged for examination. This information helps the expert focus and identify faulty rules or detect missing rules. The automated discrepancy detection tool includes a separate control module that monitors the behavior of the system during the reasoning process. After analyzing all the possible cases of discrepancies, meta rules [6] are constructed with the cooperation of the human expert and the knowledge engineer. Meta rules should be equipped in the control module and used to check the premise failures during the run [2].

Once deficiency is identified in the knowledge base, the system must suggest repairs to the knowledge base or request the expert to handle them. In addition to identifying and localizing discrepancies, it is also important to know the context in which they occurred. In most systems the repair work is carried out by the human expert. Discrepancy detection involves identifying the rules used for deducing an incorrect or missing conclusion. The system then lets the expert perform clause-to-clause analysis of conditions that specify the current situations. Conclusions and conditions of the applicable rules are examined for the possible cause of discrepancy. If degree of belief is involved in the conclusion, the adjustment of this value also becomes possible repair candidates.

Many reasoning systems adopt an approach to uncertainty handling [7]. Formal probability theory is the basis of determining the possibility of an event or an outcome. Bayesian inference is a method of statistical inference that deals with uncertainty and requires updating the probability for a hypothesis as more evidence or information becomes available [8]. Fuzzy logic is an approach to computing based on degrees of truths rather than the Boolean logic of true and false [9]. The Dempster-Shafer (DS) theory adopted in this research is referred to as the theory of belief functions or evidence theory [10, 11]. As more evidence is available, the belief values of the hypothesis set are updated by the rule of combination. Among other uncertainty handling schemes, the evidence combination of the DS theory is quite unique. The DS theory has been used for reasoning with degree of belief. Degree of belief is measured by estimating how much the evidence is supportive of a hypothesis [10]. The theory involves the assigning of degree of belief over proper subsets of the hypotheses. When new evidence is available, it is combined with the current set of degrees of belief. In reasoning with uncertainty by the DS scheme, the belief values for a set of hypotheses are constantly updated as more evidence is available [11-13]. This forms a belief state. Reasoning with DS theory can be regarded as belief state transition, moving from one belief state to another.

The result of evidence combination may lead to confirming belief state. Conflicting evidence may lead to disconfirming belief state, decreasing degree of belief in the hypothesis. For example, there is a hypothesis set, H, for a proposition, say “the dog is sick”, H = {sick, healthy}. From a set of symptoms and test results of the dog, the first rule, Rule 1, concludes that the dog is sick with the belief value of .5. The rest of the belief value, 0.5, is assigned to the whole set, H, which corresponds to the frame of discernment. If another piece of evidence becomes available and supports the contradicting hypothesis, ‘healthy’ by Rule 2, with a belief value of 0.5, both are reduced to belief value of 0.33. If there is no rule firing of Rule 2, the belief value for ‘sick’ would increase from 0.33 to 0.5. From this, we can define the contribution factor of a rule. The amount of contribution of the rule can be computed by retracting the rule from the line of reasoning. Retracting a rule could lead to the reduction in the degree of belief in a hypothesis if the rule supports the hypothesis under consideration. It also could increase the degree of belief in a hypothesis if the rule supports other hypotheses which is the case of evidence confliction. Retracting a rule is sometimes needed if two rules conflict each other.

In general, most of uncertainty handling schemes are based on the supportive reasoning approach, in which evidence should support the hypothesis. However, we can imagine a situation where another piece of evidence refutes the same hypothesis. If the evidence supporting 'healthy' should not have been asserted, the amount of support for 'healthy' should be retracted from the belief value of 'healthy'. The belief values for 'sick' and 'healthy' should be recalculated. These types of problems can easily be found in real world applications and is well studied in the field of non-monotonic reasoning with propositional or predicate logics [14].

The rest of the paper is organized as follows. In section 2, basic principles of the Dempster-Shafer theory are discussed. In section 3, the belief function retraction that reverses the effect of evidence combination is described. The contribution measure of a rule is also defined. In section 4, a tracing algorithm using contribution measure is discussed for rule refinement. Section 5 concludes the paper.

2. Dempster-Shafer Theory

Dempster-Shafer (DS) evidence theory can be interpreted as a generalization of probability theory. The main difference from probability theory is that probabilities are assigned to sets as opposed to mutually exclusive singletons. Evidence can be associated with multiple possible events or hypotheses. This includes the rule of combination that allows one to combine evidence from different sources and arrive at a degree of belief that takes all the evidence into account. Original theory has been expanded to adapt to different kinds of situations [15]. It has been known to be an effective tool in handling conflicts among evidence. The DS theory begins with a frame of discernment (Ω) which is composed of a set of all possible singleton hypotheses. Ω is also referred to as a domain. Given Ω, a probability mass is assigned to each subset of Ω. In classical probability theory, a probability mass is given to each element. Such an assignment is called a basic probability assignment and is represented by a mass function, m. Therefore, $0 \leq \mathrm{m}(\mathrm{A}) \leq 1$ if A is a subset of Ω. The mass function, m, satisfies two conditions: the mass of empty set (ϕ) is zero, and the sum of masses for all subset of Ω is 1. The measure, $\mathrm{m}(\mathrm{A})$, quantifies the proportion of all relevant and available evidence that supports A. The belief function for set A is defined in Eq. (1) as the sum of all the basic probability assignments of the proper subsets of the set A, i.e., for $\mathrm{B} \subseteq \mathrm{A}$. $\mathrm{Bel}(\mathrm{A}) = \sum_{B \subseteq A} m(B)$ [12].

Belief in a hypothesis expresses the amount of belief that directly supports the given hypothesis and its proper subsets. Evidence combination needs to combine two independent sets of basic probability assignments. The joint mass is calculated from the two sets of masses, m_1 and m_2 as in Eq. (1) [7, 10].

$$m_{1,2}(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \phi} m_1(B) m_2(C) \tag{1}$$

where K, representing basic probability mass associated with conflict, is defined as in Eq. (2).

$$\mathrm{K} = \sum_{B \cap C = \phi} m_1(B) m_2(C) \tag{2}$$

K is a measure of the amount of conflict between the two mass sets. The denominator in the joint mass, $1 - \mathrm{K}$, is used as a normalization factor by which the conflict is completely ignored.

3. Belief Function Retraction and Contribution Measure

The DS scheme includes the rule of evidence combination that combines two belief functions and generates a new belief function [15], i.e.,

$$Bel_p \oplus Bel_q = Bel_r \tag{3}$$

The $\oplus$ operator is used to represent the evidence combination between two belief functions. Bel_p, belief function for new evidence is combined to Bel_q, the belief function for all the evidence that has already been combined so far. A belief state set for a node (hypothesis set) is defined by <E, H, Bel_q>. E is a set of evidence and H is a set of hypotheses. Bel_q is called the current belief state for hypothesis set, H. With a new belief function, Bel_p, Bel_r will be the next state of belief. The focal elements (FEs) of Bel_p is represented by $\{p_i | 1 \leq i \leq l\}$ and mass function, m_p. The focal elements of Bel_q, FE_q is represented by $\{q_i | 1 \leq i \leq m\}$ and mass function m_q. The focal elements of Bel_r, called updated belief state resulting from evidence combination is represented by $\{r_i | 1 \leq i \leq n\}$ and mass function m_r.

When evidence for Bel_p is retracted from an evidence pool, its effect contributing to the degrees of belief in the current hypothesis set should also be retracted by reversing the evidence combination. Retracting Bel_p from Bel_r as shown in Eq. (5) recovers the belief state back to the state of Bel_q [16]. $\ominus$ represents the evidence retraction operator.

$$Bel_q = Bel_r \ominus Bel_p \tag{4}$$

After evidence combination, the resultant Bel_r includes the focal element set of Bel_p and Bel_q. It also includes $\{p_1q_1, p_1q_2, \ldots\}$, where p_iq_j is a non-empty set with non-zero belief value resulting from set intersection of p_i and q_j. This implies that Bel_r always includes the non-empty intersection set of p_i and q_j as well as p_i's and q_j's. Therefore, Bel_p and Bel_q are subsets of Bel_r. As new evidence is asserted, evidence combination increases the size of the focal element set, $FE_p \subseteq FE_r$ and $FE_q \subseteq FE_r$. Table 1 shows the combination. $a_i's$ and $b_i's$ are belief values for $p_i's$ and $q_i's$, respectively and $c_{ij} = a_i \cdot b_j$. The p_iq_j term represents set intersection of $p_i \in FE_p$ and $q_j \in FE_q$. The focal elements that exist in FE_r but not in FE_q are added with zero belief value to Bel_q. This makes the reverse evidence combination computation possible because Bel_q is a subset of Bel_r and the amount of effect for that particular set of focal elements should be identified. The number of focal elements of Bel_q should be the same as the size of Bel_r, n [16].

Table 1. Dempster's Rule of Combination

		Bel_q			
		$q_1(b_1)$	$q_2(b_2)$	...	$q_n(b_n)$
Bel_p	$p_1(a_1)$	$p_1q_1(c_{11})$	$p_1q_2(c_{12})$	...	$p_1q_n(c_{1n})$
	$p_2(a_2)$	$p_2q_1(c_{21})$	$p_2q_2(c_{22})$	...	$p_2q_n(c_{2n})$
	$\cdots$	$\cdots$	$\cdots$	...	$\cdots$
	$p_l(a_l)$	$p_lq_1(c_{l1})$	$p_lq_2(c_{l2})$	...	$p_lq_n(c_{ln})$

Note that $q_i = r_i$ for $1 \leq i \leq n$. p_iq_j could be either a null set or a member of focal element of Bel_r. Belief value, c_k, for a focal element, r_k, of Bel_r can be computed as in Eq. (5).

$$Bel_r(c_k) = \frac{1}{1-K}\sum_{i=1}^{l}\sum_{j=1}^{n} c_{ij} \ \text{ for } \ p_iq_j = c_k, \text{ where } K = \sum_{i=1}^{l}\sum_{j=1}^{n} c_{ij} \ \ for \ \ p_iq_j = \phi \tag{5}$$

To find the belief values of Bel_q, n equations for Bel_r are rearranged for q_i. These form a set of linear equations. Solving this set of equations produces the belief values for the focal elements of Bel_q. The given p_i's and r_i's, simultaneous equations are represented by $\mathbf{C} \cdot \mathbf{q} = \mathbf{r}$, where $\mathbf{C}$ is a coefficient matrix obtained from these equations and $\mathbf{q}$ is a variable vector composed of q_i's. **r** is a constant vector given by r_i's. After retracting a belief function, the resultant belief values can be compared with the old belief values. The contribution measure is computed by the change in belief value for each hypothesis. The contribution measure is defined as the amount of change in belief with respect to the current belief value.

Table 2. Contribution of a Rule

	Belief Function for H			Belief State			Contribution Measure		
	h_1	h_2	h_3	h_1	h_2	h_3	h_1	h_2	h_3
R1	0.3	0.5	0.0	0.300	0.5	0.0	-1.0	-1.0	-1.0
R2	0.2	0.0	0.3	0.379	0.379	0.091	-0.208	0.319	-1.0
R3	0.3	0.0	0.4	0.490	0.204	0.223	-0.227	0.858	-0.592
R4	0.3	0.1	0.2	0.558	0.211	0.182	-0.122	-0.033	0.225
R5	0.0	0.2	0.5	0.358	0.246	0.364	0.559	-0.142	-0.500

For a given hypothesis set $\mathrm{H} = \{h_1, h_2, h_3\}$, assume that five rules, $\mathrm{R} = \{R_1, R_2, R_3, R_4, R_5\}$, are fired sequentially with belief values on hypothesis set, H, as in Table 2. Table 2 shows three columns. Belief functions are listed on the first column. Updated belief values after each rule firing are shown in the Belief State column. Next to it, contribution measure for each rule is listed. The contribution measure ranges from -1.0 to 1.0. Negative value means that belief in hypothesis decreases if the rule were not fired. The rule has positive support to the hypothesis. The greater the value, the greater change we see in belief value. Positive value means the reverse. Figure 1 shows the changes in belief values as rules are fired from R1 to R5. Substantial change occurs after R3. The belief value in h2 reduces from 0.379 to 0.204. The contribution measure by evidence from R3 is the greatest in the table, 0.858. Before R3 is fired, both h_1 and h_2 are likely with belief values of 0.379. R3 firing makes h_1 much more likely than h_2, 0.490 vs 0.204. When a hypothesis is newly asserted, the contribution measure becomes -1.0.

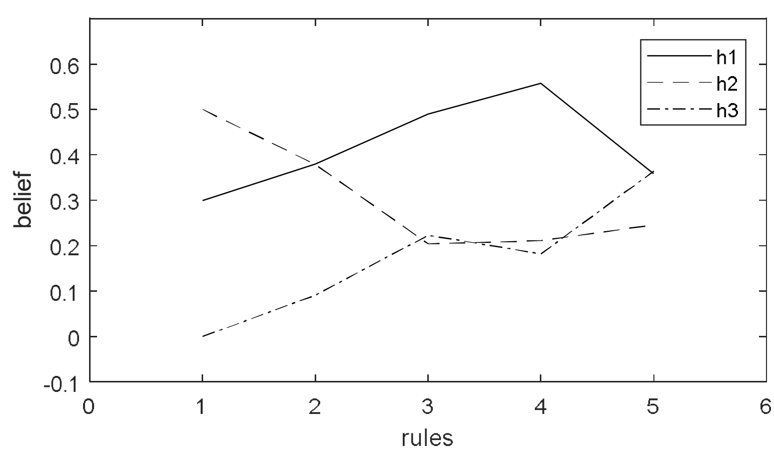

Figure 1. Belief Value Changes

4. Tracing Algorithm by Using Contribution Measure

Checking individual rules during incremental knowledge acquisition allows the expert to resolve the deficiencies in the knowledge base to some extent. However, it turns out that rules that look fine individually may not work as intended when they are put to work together in a specific context during test case runs. This is mainly because it is practically impossible for the expert to enumerate all possible cases. Therefore, such problems should be dealt with subsequently by a knowledge refinement methodology. In practice, test cases are used to identify and locate deficiencies in the rules. When test runs for cases produce discrepant results, the first step is to identify and locate the cause of the problem and then go through repair suggestion. Rerunning the test case would validate the repair. In general, the human expert has good knowledge about rule structures and can analyze the consequences of the evidence on a test case. A tool for knowledge refinement is needed for the human expert to perform fault diagnosis. The tool helps the expert to detect and locate the problem present in the knowledge base. The repair should be decided entirely by the expert. Once the new rule is added or an existing rule is modified, the validation task follows to make sure that the repair would produce an intended effect.

One of the key tasks in successful knowledge base refinement is to have the ability to trace case runs and identify the set of evidence and rules involved in drawing intermediate and final conclusions. This also includes a method to determine the degree of the contribution of the evidence in terms of the supportive belief. In some cases, the system may have reached the expected conclusions, but may not provide sufficient confidence in these conclusions. In other cases, when multiple conclusions are derived, the expert may be in disagreement with the ranking of these conclusions. We have developed a tracing algorithm to help the expert locate faulty rules. The tracing algorithm allows the expert to go through the reasoning steps of a test case. The method allows the expert to go through individual rules, hypotheses, and belief values. It also helps the expert analyze how individual rules are applied to reach the current belief state. The knowledge refinement module does not do automatic fault location, but acts like a debugging tool to assist the expert in locating the faults by providing the tracing mechanism. We use geological cases and illustrate the algorithm for locating faulty rules in the rule base. Tracing algorithm starts from the goal node of the rule network when the expert identifies a node for deducing inconsistent conclusions. As shown in Figure 2, which depicts a portion of the rule network after a test case run, the system concludes that the hydrocarbon charge is (good = 0.533) and (poor = 0.096).

However, the expert pointed out that the test case should have shown more belief in the hypothesis that hydrocarbon charge is good. The expert is required to select a set of expected hypotheses in the top-level goal node. If they are selected, the system computes contribution measure for each hypothesis. When this information is ready, the expert can perform blame assignment analysis. In this case, (hydrocarbon-charge = good) is selected as a positive hypothesis and (hydrocarbon-charge = poor) as a negative hypothesis. Using retraction algorithm, the contribution to each hypothesis is computed and represented by a number (%). In Figure 2, the nodes in the first two levels are displayed with their contribution measure to both hypotheses. The first number next to each node is the contribution to the conclusion 'good' and second number to the conclusion 'poor'. If the number is negative, it means that the node has a supportive role. Without its support it would result in a decrease in belief. Positive means the opposite. If the contribution measure is zero it means that the given evidence is irrelevant to the hypothesis set.

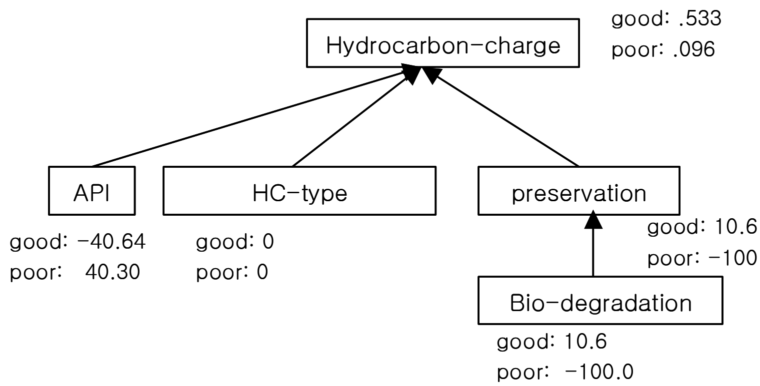

Figure 2. Contribution Measures of Nodes

Numbers in first level indicate important facts. A set of evidence about API has the most contribution (-40.64%) to the current belief state in a supportive way. On the other hand, hydrocarbon type, HC-type has no effect on the conclusions. Two other nodes that are not shown in this figure also have a significantly supportive effect on the conclusions. Notice, however, that preservation contributes in an opposite way, i.e., if no conclusions are drawn on this attribute, it would have increased belief values in (Hydrocarbon-charge = good) by 10.6% and it could completely eliminate (Hydrocarbon-charge = poor). This seems to be a source of the problem. The expert may lay blame on preservation and take a closer look at the rules involved in concluding (preservation = altered). The expert points out that (preservation = altered) did not support (Hydrocarbon-charge = good). Rather it supports the negative hypothesis (Hydrocarbon-charge = poor). The expert thought that the rule was fine, and it was not supposed to fire. To identify why the rule fired, the expert moves down to the 'Bio-degradation' node and checks the rules associated with Bio-degradation. (Bio-degradation= high) was derived from two other nodes associated with it, which is not shown here. Given the evidence available on these two attributes, the conclusion on Bio-degradation was desired to be low, which did not happen in the actual case run. The repair suggestion would be to look for rules that support (Bio-degradation=low). Adding more conditions to the rule that draw conclusion on (Bio-degradation=high) restricts the possibility of firing the rule. In this process, the expert has the control in locating faulty rules and also suggesting repairs. On the other hand, to make it possible, the system should furnish detailed information at the expert's request to help identify the problem and locate the cause of the problem. Tracing mechanisms should include facilities that retrieve important information from rule network, e.g., retrieving related rules and evidence with belief values.

5. Conclusion

Issues such as knowledge acquisition and knowledge refinement are important in building reliable knowledge bases. Discrepancy is detected when results of test cases are different from those actually derived from the system. The system helps the expert perform clause-to-clause analysis on fired rules. Belief state for a hypothesis set is introduced to define the state of reasoning. The retraction mechanism has been provided to retract the effect of a belief function. Contribution measure is provided to estimate the impact of a rule in the line of reasoning process. The belief function retraction method is found to be useful when uncertainty is involved in the reasoning process. Tracing algorithm starts from the goal node of the rule network when the expert identifies a node for deducing inconsistent conclusions. The contribution measure is used to guide the expert to the faulty rule and identify the cause of the problem. The Dempster-Shafer theory chosen for evidence combination handles uncertain situations effectively even when conflicting evidence is fed into the reasoning system. However, this also gives rise to inconsistent conclusions. We have shown that the belief retraction method and the tracing algorithm facilitate the rule refinement.

References

[1] C. Wagner, "Breaking the Knowledge Acquisition Bottleneck Through Conversational Knowledge Management," *Information Resources Management Journal*, Vol. 19, No. 1, pp. 70-83, 2006.

[2] C.B. Wang, Y.J Chen, and Y.M. Chen, "Knowledge Refinement for Engineering Knowledge Management," *Concurrent Engineering*, Vol. 13, Issue 1, pp. 43-56, 2005.

[3] G. Schreiber, "Knowledge Acquisition and the Web," *Inter. Journal of Human-Computer Studies*, Vol. 71, Issue 2, pp. 206-210, 2013.
DOI: https://doi.org/10.1016/j.ijhcs.2012.10.001.

[4] B.J. Wielinga, "Reflections on 25+ Years of Knowledge Acquisition," Vol. 71, Issue 2, pp.211-215, 2013.
DOI: https://doi.org/10.1016/j.ijhcs.2012.10.007.

[5] G. Li, "Human-in-the-loop Data Integration," *Proc. of the VLDB Endowment*, Vol. 10 No. 12, pp.206-217. 2017.
DOI: https://doi.org/10.14778/3137765.3137833

[6] D. Dubois, H. Prade, and S. Schockaert, "Rules and Meta-rules in the Framework of Possibility Theory and Possibilistic logic," *Scientia Iranica*, Vol. 18, No. 3, pp. 566-573, 2011.
DOI: https://doi.org/10.1016/j.scient.2011.04.008

[7] L.S. Dutt and M. Kurian, "Handling of Uncertainty – A Survey," *Int. Journal of Scientific and Research Publications*, Vol. 3, Issue 1, pp. 1-3, 2013.

[8] J. Stone, "Bayes's rule: A Tutorial Introduction to Bayesian Analysis," Sebtel Press, England, 2013.

[9] R. Yager, "Fuzzy Relations between Dempster-Shafer Belief Structures," *Knowledge-based System*, Vol 105, pp. 60-67, 2016.
DOI: https://doi.org/10.1016/j.knosys.2016.04.027

[10] K. Sentz and S. Ferson, "Combination of Evidence in Dempster-Shafer Theory," Sandia National Lab., Apr. 2002.

[11] K. Zhou, A.Martin, and Q. Pan, "Evidence Combination for a Large Number of Sources," *20th Int. Conf. on Information Fusion*, Jul. 2017, China

[12] Q. Chen, A. Whitbrook, U. Aickelin, and C. Roadknight, "Data Classification Using the Dempster-Shafer Method," *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 26, Issue 4, pp. 493-517, 2014.
DOI: https://doi.org/10.1080/0952813X.2014.886301

[13] Q. Chen and U. Aickelin, "Anomaly Detection Using the Dempster-Shafer Method," *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 26, Issue 4, pp. 493-517, 2014.

[14] L. Giordano and F. Toni, "Knowledge Representation and Non-monotonic Reasoning," A 25-year Perspective on Logic Programming, pp. 87-111, 2010.

[15] Y Deng, "Generalized Evidence Theory," *Applied Intelligence*, Vol. 43, Issue 3, pp. 530-543, 2015.

[16] G. Lee, "An Evidence Retraction Scheme on Evidence Dependency Network," *Int. Journal of Advanced Smart Convergence*, Vol. 8, No. 1, 2019, pp. 133-140,
DOI: https//dx.doi.org/10.7236/IJASC.2019. 8.1.133