2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Collaborative Medical Diagnosis through Fuzzy Petri Net Based Agent Argumentation

Xuehong Tao
College of Education
Victoria University
Melbourne, Australia

Yuan Miao, Yanchun Zhang
Centre for Applied Informatics
College of Engineering and Science
Victoria University
Melbourne, Australia

Zhiqi Shen
School of Computer Engineering
Nanyang Technological University
Singapore

***Abstract*—Online health information services and self diagnosis systems become popular recent years. We propose a computing model for collaborative medical diagnosis through multi agent argumentation. In this model, the agents are able to communicate with each other to share information, critique and verify each other's knowledge, and collaboratively make diagnosis based on multiple agents' knowledge through an argumentation process. Fuzzy Petri Net (FPN) is adopted as the agents' knowledge model. Different from the commonly used FPNs that assign tokens in places, we assign tokens on arcs and also give places capability in controlling the inference of FPN. The FPN based argumentation is automated with algorithms. The proposed model can be employed to achieve collaborative healthcare diagnosis systems, where agents with different expertise collaboratively argue with each other to come up with a mutually agreed diagnosis.**

Keywords—fuzzy petri net; collaborative argumentation; multi agent systems; medical diagnosis

I. INTRODUCTION

THERE are many online health information services that provide informative medical knowledge for the public as well as tools for self diagnosis. These services provide additional sources to promote awareness of diseases. They also enable people to make prediction based on symptoms, get alert for certain signs of disease, as well as act as initial steps of heath care management. These services will not replace but act as a compliment to formal medical consultations.

There exist large amount of people who use the Internet to access health information and make self diagnosis. According to the Bupa Health Pulse Survey 2011 [1], of the 13,373 respondents from 12 countries, 83% say they "often" or "sometimes" search the Internet for information and advice about their health, medicines or medical conditions. Specifically, there are 70% in UK, 77% in Australia and 94% in China. Overall, well over a third (39%) seeks out information to make a self diagnosis.

The healthcare diagnosis systems typically provide interfaces guiding users to describe their symptoms, then predict the possible illness and display the illness related knowledge and treatment methods. For example, WebMD [2] and Mayo Clinic [3] are two such online services. However, patients usually don't know which kinds of information are important for diagnosis and are not able to describe their conditions precisely. The situations are different when visiting human doctors. Human doctors ask patients questions to clarify and refine the symptoms until precise health conditions are obtained. It is desirable for online diagnosis systems to be able to conduct such interactive dialogues.

Doctors started to participate in online healthcare services in recent years, such as Chunyuyisheng [4] and DoctorSpring [5]. Due to the availability of doctors or asynchronous communication protocols, it takes time for patients to receive responses from doctors. In addition, health professionals possess expertise only on specific areas. For complex medical cases, diagnosis may encompass several consultations with different professionals before arriving at a final diagnosis. Collaboration is a beneficial method for doctors to access each other's expertise. However, it is not easy to call together all relevant human doctors to discuss asynchronously online for the diagnosis of a patient.

People need interactive healthcare diagnosis systems where doctors collaboratively make diagnosis. Intelligent agents could play important roles in this context. Agents are autonomous entities that can communicate with other agents or humans to perform tasks automatically. Multi agent collaborative diagnosis is a practical solution to online medical diagnosis applications.

There are several multi agent based applications in medical diagnosis. For example, to have a master agent share tasks among different specialist doctor agents based on the symptoms of a patient [6]; to have specialist agents contribute to the diagnosis or treatment opinions in special domains, examiner agents conduct lab and other examinations, and joint decision maker agents who gather the diagnoses produced by the specialists and using decision making method (such as weighted voting, bidding) to choose a final diagnosis [7]; to employ information searching agents find information from the databases (for example, patient's historical data, side effect of medicine) to help the physician agent in the medical diagnosis and treatment process [8]. All the aforementioned researches focus on coordinating multi agents to finish different tasks within a diagnosis problem. There is no collaborative diagnosis based on multiple agents' knowledge.

This paper proposes a method to support multi agents' diagnosis through collaborative argumentation. Collaborative

This research is supported by Australian Research Council Linkage Project LP100100624.

argumentation is a form of collaborative discussion in which multiple parties work together to resolve an issue, and in which all participants expect to find agreement by the end of the argumentation [9]. Collaborative argumentation have many benefits in medical diagnosis, such as more precise diagnosis based on multiple parties' knowledge; better understand of patients' health condition through interaction; and knowledge creation by learning from each other. In a collaborative medical argumentation, the multiple agents should share knowledge with each other, verify each other's decision and collaboratively come up with a final mutually agreed diagnosis.

Agents' knowledge can be extracted from the enormous amounts of medical data collected nowadays by machine learning techniques. The knowledge can then be applied in corresponding fields to increase the quality of decision making, predict possible diseases and support medical diagnosis. A knowledge rule mined from dataset usually has a confidence value attached which indicates how certain the knowledge is. Fuzzy petri net is adopted in this paper to handle the inference with uncertainties.

The rest of this paper is organized as follows: section 2 introduces fuzzy petri net; section 3 introduces the diagnosis agent we designed; section 4 introduces the automation of agent argumentation; section 5 illustrates the method with an example and section 6 concludes the paper.

II. Fuzzy Petri Net

Petri net is a promising graphical and mathematical modeling tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic [10]. To represent systems that contain fuzzy behaviour, petri nets are extended to fuzzy petri nets [11, 12].

A. Fuzzy Petri Net (FPN)

A Fuzzy Petri Net (FPN) is a bipartite directed graph which contains two types of nodes: places and transitions, where circles represent places and bars represent transitions. The relationships from places to transitions and from transitions to places are represented by directed arcs. A generalised fuzzy petri net structure can be defined as an 8-tuple [11]:

FPN=(P, T, D, I, O, CF, V, M),

where

- $P = \{p_1, p_2, \ldots p_n\}$ is a finite set of places;
- $T = \{t_1, t_2, \ldots t_m\}$ is a finite set of transitions;
- $D = \{d_1, d_2, \ldots d_n\}$ is a finite set of propositions;
- I: $T \rightarrow P^{\infty}$ is an input function which maps transitions to their input places;
- O: $T \rightarrow P^{\infty}$ is an output function which maps transitions to their output places;
- CF: T→[0, 1] is a function which maps each transition to a real value between zero and one. The values for transitions are noted as cf_1, cf_2, … cf_m;
- V: T→[0, 1] is a function which maps each place to a real value between zero and one. The values for places are noted as v_1, v_2, … v_n;
- M: $P \rightarrow D$ is an association function which maps each place to a proposition. Places are graphical representations of propositions in an FPN.

In a fuzzy petri net, function I describes the input of a transition, and function O describes the output of a transition. If $p_j \in I(t_i)$, then there exists a directed arc a_{ji} from place p_j to transition t_i. If $p_k \in O(t_i)$, then there exists a directed arc a_{ik} from transition t_i to place p_k.

Rules can be represented by fuzzy petri nets [11]. In a fuzzy petri net, T corresponds to rules, P and D correspond to propositions of rules. P is the graphical representation of propositions and D contains the meaning of propositions. The fuzziness of FPNs is manifested by the certainty factors of places and transitions. V is a set of certainty factors for places which are the degree of truth for propositions. CF is a set of certainty factors for transitions which are the degree of truth for rules.

For example, Fig. 1 shows a fuzzy petri net, where,

$P=\{p_1, p_2, p_3\}$ $T=\{t_1\}$

$D=\{$ d_1="eat lots of sugar", d_2="seldom exercise",
d_3="has diabetes"$\}$

$I(t_1)=\{p_1, p_2\}$ $O(t_1)=\{p_3\}$

$CF(t_1)=0.6$

$V(p_1)=0.9$, $V(p_2)=0.8$

$M(p_1)=d_1$, $M(p_2)=d_2$, $M(p_3)=d_3$

This FPN describes a belief with 60% confidence that if somebody eats lots of sugar and seldom takes exercise, he/she may have diabetes. It can also be represented as a fuzzy rule: if eat lots of sugar (0.9) and seldom exercise (0.8), then the person may have diabetes. The certainty factor of this rule is 0.6.

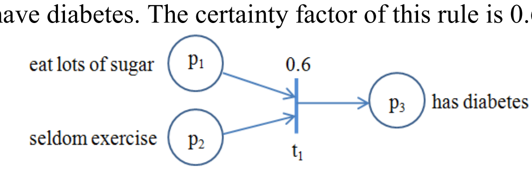


Fig. 1. A fuzzy petri net

B. Inference via Fuzzy Petri Net

Inferences of FPNs are achieved through the firing of transitions. Tokens are often used to control the inference process. A token is represented by a dot in a place. The conditions for a transition to be ready to fire may include: whether the certainty factor of each input place satisfies a predefined threshold value λ', whether each input place has a token, and whether the certainty factor of the transition satisfies a predefined threshold value λ. It depends on specific situations to decide which conditions to be used. In this paper, threshold λ' is not used. Only certainty factors of transitions and tokens are used to control the firing of transitions. A transition is enabled if it is ready to fire.

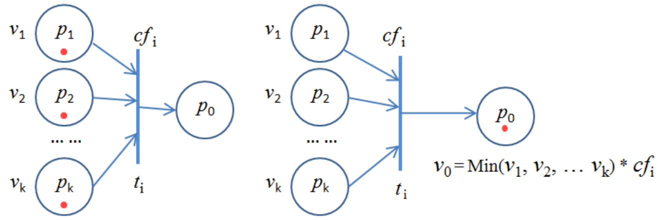
When a transition fires, it removes the tokens from all its input places and deposits one token into each of its output places. It also calculates new certainty factors for its output places.

Paper [11] gave methods to calculate certainty factors of output places, for FPNs that represent four different types of rules. FPN reasoning processes for two types of rules are relevant to this paper and they are introduced below.

If a rule is

If p_1 and p_2 and … and p_k Then p_0 (cf_i)

The fuzzy reasoning process of this type of rule can be modelled by an FPN shown in Fig. 2. The certainty factor of the premise is the minimum value of certainty factors of the individual propositions in the premise. This aligns with real world situations, that is, if any part of a whole is not certain, the whole thing is not certain. In the example of Fig. 1, the certainty factor for p_3 will be Min(0.9, 0.8)*0.6=0.48.



(a) Before firing transition t_i (b) After firing transition t_i

Fig. 2. Firing process for a transition with multiple inputs

If there are many rules have the same conclusion, such as

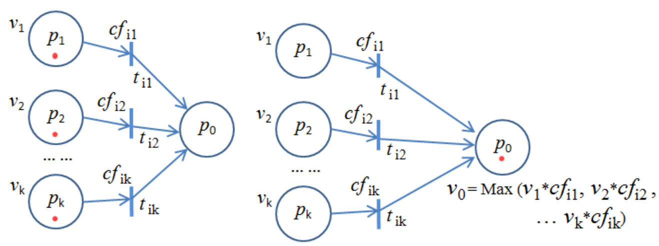
If p_1 Then p_0 (cf_{i1})

If p_2 Then p_0 (cf_{i2})

… …

If p_k Then p_0 (cf_{ik})

The fuzzy reasoning process of this type of rule can be modelled by an FPN shown in Fig. 3. The certainty factor of the conclusion is the maximum value of the certainty factors obtained by applying different rules. This aligns with real world situations, as we usually use our most confident methods (rules) to make decisions.



(a) Before firing transitions (b) After firing transitions

Fig. 3. Firing process for a place with multiple inputs

With the firing process of FPNs, we can predict the truth of propositions based on the certainty factors of known propositions. The next section will introduce the design of a diagnosis agent that is able to reason based on FPN.

III. Fuzzy Petri Net Based Diagnosis Agent

This section introduces the agent we designed for collaborative diagnosis, including its knowledge base, reasoning mechanism and knowledge update.

A. Knowledge Base

Diagnosis agents possess medical rules for diagnosis. Medical rules can be in different formats. We only consider Horn clause format rules in this paper. That is, each rule is in the form of $h_1, h_2, \ldots h_k \rightarrow h_0$. It represents that if a patient has health conditions of $h_1, h_2, \ldots h_k$, he/she may have condition h_0.

The knowledge base of a diagnosis agent is a collection of medical rules the agent possesses. Medical rules can be obtained from medical experts or by mining medical datasets. In this paper, the knowledge base is defined as a 4-tuple:

KB={H, R, C, S}, where

$H = \{ h_i \mid i = 1, 2, \ldots n \}$

$R = \{ r_i: h_{i1}, h_{i2}, \ldots h_{ik} \rightarrow h_{i0} \mid h_{i0}, h_{i1}, \ldots h_{ik} \in H, i = 1, 2, \ldots m \}$

$C = \{ c_i \mid i = 1, 2, \ldots m \}$

$S = \{ s_i \mid i = 1, 2, \ldots m \}$

$= \{ <(agent^1, x^1_i, y^1_i), (agent^2, x^2_i, y^2_i), \ldots$

$(agent^k, x^k_i, y^k_i), \ldots > \mid i=1, 2, \ldots m, k=1, 2, \ldots \}$

H is a set of health indicators. R is a relationship set where each relationship r_i describes how the health indicators are related to each other. The parts in a rule before the arrow is called premise, and after the arrow is called conclusion. C is a confidence set. c_i is the confidence of r_i which describes how certain rule r_i is.

S is a set of confidence related records collected during the interaction with other agents. s_i is confidence related records of r_i, where x^k_i refers that according to the knowledge of $agent^k$, there are x^k_i patient cases match the premise of the rule, y^k_i means that according to the knowledge of $agent^k$, the rule correctly identified y^k_i cases to be in the health condition indicated by the conclusion of the rule. For easy reference, s_i is called *confidence-records* of r_i . s_i is illustrated in Table 1.

TABLE 1.
Illustration of Confidence-Records of a Rule

	$agent^1$	$agent^2$	…	$Agent^k$	…
Matched	x^1_i	x^2_i	…	x^k_i	…
Correct	y^1_i	y^2_i	…	y^k_i	…

So $c_i = (y^1_i + y^2_i + \ldots + y^k_i + \ldots)/(x^1_i + x^2_i + \ldots + x^k_i + \ldots)$. The agent has a threshold λ. If a rule's confidence is not less than λ, this rule can be applied in diagnosis.

For example, agent$_1$ has the following knowledge base,

H= { h_1 = "high BMI (Body Mass Index)",

h_2 = "diabetes positive",

h_3 = "high blood sugar", h_4 = "old age" }

R= { r_1: $h_1, h_3 \rightarrow h_2$, r_2: $h_4 \rightarrow h_2$ }

C = { 0.75, 0.5}

S= {< (agent^1, 2000,800) , (agent^2, 8000, 6700) >,

<(agent^1, 1000,500)>}

There are two rules in this example. r_1 tells us that if a person has high BMI and high blood sugar, he/she has diabetes. The confidence of this rule is 0.75 which comes from two agents' knowledge. Agent$_1$ has 2000 patients who have high BMI and high blood sugar. Among these patients, 800 patients were diagnosed diabetes. If only consider the patients records in agent$_1$, the confidence of this rule is 40% (i.e. 800/2000). If 70% is the threshold, this rule cannot be used in diagnosis. Agent$_1$ also receives information from agent$_2$. Agent$_2$ has 8000 patients match the premise of the rule and 6700 patients were diagnosed diabetes. The confidence of this rule is 83.75% (i.e. 6700/8000) according to the knowledge of agent$_2$. After incorporation of the information from agent$_2$, the confidence of r_1 is (800+6700)/(2000+8000)=75%. It can be used in diagnosis as the confidence is bigger than the threshold 70%. Information exchange among agents can help them to learn from each other and gradually abandon local bias. Agent$_1$ also has another rule, r_2, which tells us that if a person is in senior age, he/she has diabetes. The confidence of this rule is 50%, so it cannot be used for diagnosis.

During the interaction with others, an agent will revise its knowledge base to incorporate new rules from other agents. Of course the confidence and *confidence-records* of some existing rules will be updated.

A rule based knowledge base can be represented to its equivalent FPN [11]. In the transformation, rules become transitions, propositions become places, and confidences of rules become certainty factors of transitions. The diagnosis agent is designed to be able to use and transform between rule representation and FPN representation of its knowledge. Rules are meaningful, so the agent uses rule representation to communicate with human users and other agents. FPN is efficient in computation, so the agent uses FPN representation to perform reasoning.

B. *Reasoning Mechanism*

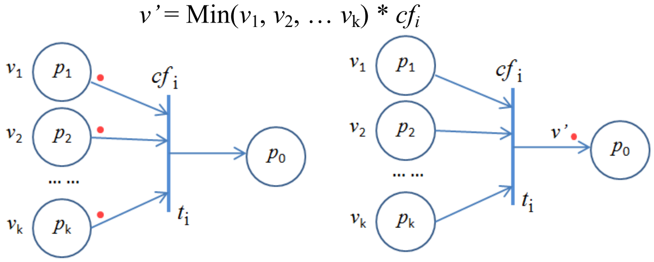
To enable FPN suitable for medical diagnosis, we modified the use of tokens. Tokens exist in places according to most petri nets [10] and fuzzy petri nets [11], this paper places tokens on arcs. One arc is restricted to have maximum one token.

Tokens in places control the firing of transitions by the availability of resources. For example, two transitions with a same input place may not fire at the same time due to the insufficient tokens in the place. This research uses FPN for logical inference. The truth of a proposition should be passed to all transitions with this proposition as input. We put tokens on arcs. Once there are updates in a place, the changes are passed to all its output arcs by placing a token on those arcs.

The reasoning process of a diagnosis agent is to make inference from what is known to what is unknown. It includes firing of transitions and updating of places.

Firing of transitions: If each input arc of a transition has a token, and a transition's certainty factor is not smaller than a predefined threshold λ, a transition is enabled to fire. When a transition fires, all the tokens on its input arcs are removed and a token is added on its output arc, as shown in Fig. 4. Meanwhile, a certainty factor is calculated for the output arc using the following formula:

$$v' = \mathrm{Min}(v_1, v_2, \dots v_k) * cf_i$$



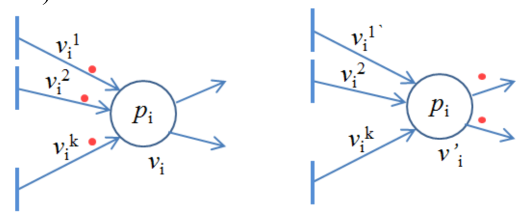
(a) Before firing a transition (b) After firing a transition

Fig. 4. Firing process for a transition

Updating of places: If one or more input arcs have tokens, a place is enabled to update. Updating a place p_i follows the following method (Fig. 5):

- Remove tokens from the input arcs of p_i.
- Suppose the old certainty factor of p_i is v_i . Update the certainty factor of p_i to $v'_i = \mathrm{Max}(v_i^1, v_i^2, \dots v_i^k)$.
- If $v'_i \neq v_i$ and p_i has output places, add a token to each of the output arcs of p_i (if that arc doesn't have one token at the moment).

So the certainty factor of a place is updated whenever new information is received from its input transitions, and only changed certainty factor is passed on (by placing tokens on its output arcs) for further inferences.



(a) Before updating a place (b) After updating a place

Fig. 5. Updating process for a place

C. *Health Status Prediction*

By firing transitions and updating places, FPNs can be used to predict the degree of truth of propositions based on known propositions. Medical diagnosis is to predict patients' health status from some known health indicators. For example, if we know a patient has high blood pressure, does he/she have heart disease? Or how certain he/she has heart disease? An algorithm for prediction is listed below.

Algorithm. Prediction

Input: *Known*: is a set of places with known certainty factors, which correspond to patient's known health indicators.
Certainty: is a set of certainty factors for places in *Known*.

Output: Certainty factors of places in set *Target*. Places in *Target* correspond to health indicators to be predicted.

// Initialising

For each place p_i in *Known*

$v_i = x_i$ //set certainty factor of p_i to the known value.

Put a token in each output arc of p_i

For each place p_i not in *Known*

$v_i = 0$

// Reasoning

Continue firing enabled transitions and updating places that are enabled but not included in *Known*, until no more actions can be done.

Whenever the certainty factor of a place is changed, use an arrow to mark the input arc which has the maximum certainty factor. The arrow starts from place and points to transition. If there are two or more input arcs with the same certainty factor and which is the biggest, randomly mark one arc.

// Reporting results

For each place p_i in *Target*,

Output p_i and v_i

If $v_i \neq 0$, set $Proof_i$ to the set of transitions obtained by tracing the arrows from p_i. Output $Proof_i$. The transitions in $Proof_i$ are rules used to predict p_i to have degree of truth of v_i.

End of Prediction.

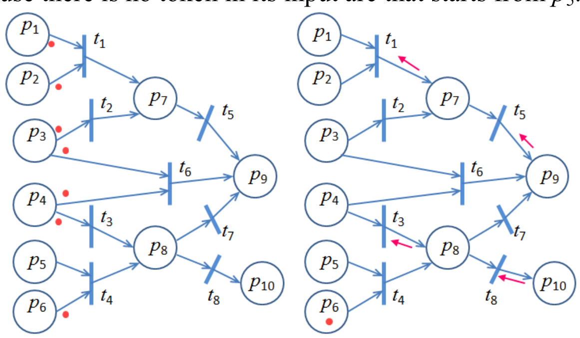
Note that certainty factors for places in *Known* are not updated, as they are considered as input and don't need to be predicted.

Suppose there is an FPN which represents a rule set

$\{ p_1, p_2 \rightarrow p_7, \quad p_3 \rightarrow p_7, \quad p_7 \rightarrow p_9,$
$p_3, p_4 \rightarrow p_9, \quad p_4 \rightarrow p_8, \quad p_5, p_6 \rightarrow p_8,$
$p_8 \rightarrow p_9, \quad p_8 \rightarrow p_{10} \}$

Know $=\{p_1, p_2, p_3, p_4, p_6\}$ are places that their certainty factors are known. *Target* $=\{p_9, p_{10}\}$ are places that their certainty factors are to be predicted. If we apply algorithm *Prediction* to this FPN, after the initializing and reasoning steps, the FPN might look like the graphs in Fig. 6 (a) and (b) respectively. When the reasoning process stops, certainty factors of p_9 and p_{10} are the predicted degree of truth of the corresponding propositions.

By tracing the arrows, transitions used to predict p_9 is recorded as $\{t_1, t_5\}$ or $\{p_1, p_2 \rightarrow p_7, \; p_7 \rightarrow p_9\}$. There are other paths that can predict p_9, but the path with arrow marks gives p_9 the highest certainty factor. Similarly, transitions used to predict p_{10} is recorded as $\{t_3, t_8\}$ or $\{p_4 \rightarrow p_8, \; p_8 \rightarrow p_{10}\}$. t_4 is not fired because there is no token in its input arc that starts from p_5.



(a) After initialising (b) After reasoning

Fig. 6. Prediction process

In this paper, we made changes to the traditional FPNs. We put tokens on arcs, and let places have capability to control the propagation of tokens. This ensures that each enabled transition only fires once and only changes are propagated along the fuzzy network.

D. Knowledge Base Update

During the interaction with others, an agent will receive new information that it doesn't know before. Accept or reject? This type of decisions also needs to be made in human beings' everyday lives. Human beings have the default readiness to accept new things. In human cognitive functioning, there is a strong tendency to accept incoming information as true, as Gilbert [13] pointed out that "unacceptance is a more difficult operation than is acceptance" (p. 111). Lee [14] stated that "this natural preference for acceptance over rejection is a manifestation of the fundamental psychological tendency shaped through the course of human evolution". Mantovani [15] explained this in evolutionary terms, "we act in a world in which it is important to respond promptly to situations, while accuracy usually is not the top priority. The result is that human cognitive systems have developed adaptively the tendency to treat all representations as if they were true, except when there is proof to the contrary" (p.680). Artificial intelligence is to simulate human's behaviors. We also design our agents to have the readiness to accept new knowledge unless they have proof that the new knowledge is incorrect.

During collaborative diagnosis, an agent may receive knowledge from others. For the knowledge that is totally new and the agent doesn't know it before, the agent accepts the new knowledge. For the knowledge that the agent can prove its confidence to be too low based on the agent's knowledge base, the agent doesn't use it for diagnosis.

If an agent receives a set of rules *RSet*, and the *confidence-records* of rules in *RSet*, the following algorithm can update the agent's FPN based on the knowledge in *RSet*.

Algorithm. Update (*RSet*, *S*)
Input: *RSet*: a rule set $\{r_1, r_2, \ldots r_n,\}$
S: a set $\{s'_1, s'_2, \ldots s'_n\}$ contains *confidence-records* for rules in *RSet*.
Output: Updated FPN.

For each r_i in *RSet*
If r_i does not exist in FPN
Add a corresponding transition t_i in FPN if it does not cause loop,
$s_i = s'_i$ // set s'_i as *confidence-records* of r_i
Else // r_i exists as transition t_i, with *confidence-records* s_i
Update s_i based on information in s'_i.
Set certainty factor cf_i based on s_i.
End of Update.

The *Update* algorithm incorporates new knowledge and corrects the certainty factors of existing knowledge. Low confidence rules are not deleted in the update process. They are still stored in the knowledge base but disabled from any diagnosis. They may become enabled again if their certainty factors meet the threshold after later update processes.

The update process helps the agent to switch from local bias belief to considerably impartial global views.

IV. Diagnosis through Argumentation

A. Argumentation Dialogue Types

Dialogues are the basic components in an argumentation. There are several dialogue types proposed in the literature for human or agent communication [16, 17, 18, 19]. The existing studies on argumentation dialogues showed that although the detailed dialogue types may vary depending on the context, the common dialogues are those to express one's position, justify one's position, attack the other's position, and exchange information with others. We use the following dialogue types in this paper:

Propose: propose the predicted disease together with rules as justifications.
Disagree: show disagreement with certain rules.
Question: ask questions.
Information: provide information.

These dialogues enable agents to collaboratively critique and evaluate each other's predictions to have a more precise diagnosis, and exchange information with others to extend and refine their knowledge base.

Each dialogue has a head which indicates the type of the dialogue, followed by additional information. The propose dialogue consists of keyword *Propose*, followed by the proposed disease, certainty factor of the disease and the rules used to make the prediction. The *Disagree* dialogue lists the rules it disagrees. The *Question* dialogue seeks information on one health indicator. For example, dialogue [Question | mental disorder] asks other agents whether the patient has mental disorder. The *Information* dialogue provides information on a health indicator. Format of the four dialogues are:

- [Propose | Proposed disease | Certainty factor of the disease | Rules used to justify the proposal, and the corresponding *confidence-records* of rules]
- [Disagree | Rules disagreed, and the corresponding *confidence-records* of rules]
- [Question | Health indicator]
- [Information | Health indicator | Certainty factor of the indicator]

B. *Argumentation Automation*

When a patient starts to describe his/her symptoms, a diagnosis starts. Since the agents' knowledge comes from various sources and with different specialties, the agents may have different predictions. These agents need to argue with each other to come up with a mutually agreed diagnosis. Now comes to the question that how to efficiently manage the agents' argumentation dialogues. Of course, these agents cannot talk whenever they want. Agent's discussion is similar to human doctor's discussion. If all doctors explain their own point of view at the same time, it is hard to draw a final conclusion and the discussion will not make sense to the listeners. In human's cases, we usually limit the number of persons in an argumentation.

To coordinate the agents' argumentation and make sure the discussion makes sense to listeners, the number of agents in an argumentation should be restricted. Some researchers use the idea from the Arena Contest of Chinese KungFu to transform multiple party argumentation into two-party argumentation [20, 21]. We follow the similar idea.

For easy understanding of the protocol, let's imagine an argumentation stage and maximum two agents are allowed to argue on the stage. There is a chair agent who is in charge of the argumentation. When a patient finishes describing his/her symptoms, the chair agent announces the start of the argumentation. The first agent who would like to make diagnosis goes up the stage to propose illness the patient is in risk or ask patient questions. Another agent can go up the stage and argue with the first agent. If an agent has no higher certainty proposals and no more to argue, it leaves the stage. The agent with the higher certainty proposal stays on the stage and waits for other challengers. If none of the agents on the stage has proposals finally, both of them leave the stage, and other agents go up the stage to argue. This process continues until no more agents go up the stage. If finally a proposal exists, the patient has some certainty of this disease. If later there are agents have proposals on other disease, an argumentation on that disease will start. We let the agents argue on one disease at a time.

Now we will concentrate on the argumentation among two agents. When an agent receives some health indicators regarding a patient, the diagnosis starts. An agent can go up the argumentation stage to make proposal or ask questions to clarify with the patient. A function for agents to make proposal is as follows.

Proposal (p_i) // p_i is a concerned disease
{
Set *Known* to a set of known health indicators of the patient, *Certainty* is a set of certainty factors for health indicators in *Known*.
Set *Target* to { p_i }
Apply algorithm *Prediction*
// the algorithm will output v_i and $Proof_i$,
// $Proof_i$ is a set of rules used for this prediction
If $v_i \neq 0$,
Generate proposal dialogue [Propose | p_i |v_i |$Proof_i$ and the corresponding confidence records set]
Else
Pick up a p_k which does not have input arcs and has path to p_i
Generate dialogue [Question | p_k]
}

Up to two agents can stay on the stage and argue about the diagnosis. When agent A receives dialogues from agent B, A will do three things: first, update its knowledge base with the rules used by B; second, generate a *Disagree* dialogue if A believes some rules from B are not correct; finally, generate *Propose* dialogue to make new diagnosis.

The procedure of how an agent replies to others is described as follows. Note that an agent may receive Disagree dialogue and Propose dialogue at one time.

Reply ()
{ // step 1: update knowledge base
If receives [Disagree | *Rules* and confidence records set *S*] ,
Update (*Rules*, *S*)
If receives [Propose | p_i | v_i | $Proof_i$ and the corresponding confidence records set *S*]
Update ($Proof_i$, *S*)
If receives [Information | p_k | v_k]
Set the certainty factors of p_k to v_k
// step 2: generate disagreement
If there are rules in $Proof_i$ with certainty factor smaller than λ, assign these rules in a set *DisagreedRules*,
Generate dialogue [Disagree | *DisagreedRules*, and the corresponding confidence records set]
// step 3: make new proposal
If the agent has not made a proposal before
Initialise FPN with *Known* and *Certainty*
Else // initialise with the updates
- For each rule updated or added in step 1, add a token to all its input arcs.
- For each place receives new information in step 1, add it to *Known*, add a token to all its output arcs.
Apply algorithm *Prediction* without the initialising step.
If $v_i \neq 0$ // p_i is the currently concerned disease
If no current proposal or the current proposal is disagreed or v_i is bigger than the current v_i,
Generate proposal dialogue

Else // no prediction on p_i

Generate question dialogue if needed.

}

The *reply*() method ensures that the agent can update its knowledge base and conduct argumentation simultaneously.

V. ILLUSTRATIVE EXAMPLE

This section uses an example to illustrate the diagnosis through agents' collaborative argumentation. The diseases concerned here are diabetes and heart disease. Now let's equip the agents with knowledge. Only rules that can diagnose the presence of a disease is stored in the knowledge base, as rules that predict the existence of certain disease are more important than rules predict the absence of such disease.

Suppose there are four agents. Agent A's knowledge was obtained from a diabetes dataset [22] with 768 records. For illustration purpose, some changes were made to the dataset: only six attributes were considered which are the number of times of pregnancy(Preg), plasma glucose concentration (Glu), blood pressure (Bp), body mass index (BMI), age and class; removed records with missing values, so 724 records left; converted the continuous data to categorical data followed method in [23]. In the resulting dataset, each attribute has the value of *L*(Low), *M*(Medium) or *H*(High), the class is diabetes positive and negative. We then used data mining tool Weka [26] to mine the dataset. By applying association rule classification method Apriori [24, 25] provided in Weka, some rules were obtained (values in brackets are certainty factors of rules):

KB^A={ r^A_1:Glu_H, BMI_H→Diabetes_Positive (0.78),
r^A_2: Glu_H, Bp_M→ Diabetes_Positive (0.75),
r^A_3: Glu_H → Diabetes_Positive (0.74) }

Suppose agent B's knowledge comes from knowledge discovery algorithm from an elderly service centre.

KB^B= { r^B_1: Age_H → Diabetes_Positive (0.7) }
r^B_2: Age_H → HeartDisease_Present (0.72) }

Agent C had knowledge obtained via Weka association rule mining on a heart disease dataset [27]. There are 270 instances, only attributes of age, sex, blood pressure (Bp) and class were considered. Age and Bp were converted to categorical data followed the method in [23]. The class has two labels, *Absent* and *Present*.

KB^C={r^C_1:Age_H,Sex_Male,Bp_H→
HeartDisease_Present(0.61),
r^C_2: Age_H, Sex_Male→ HeartDisease_Present (0.61),
r^C_3: Age_H→ HeartDisease_Present (0.5) }

Agent D had the following knowledge base (knowledge comes from [28]):

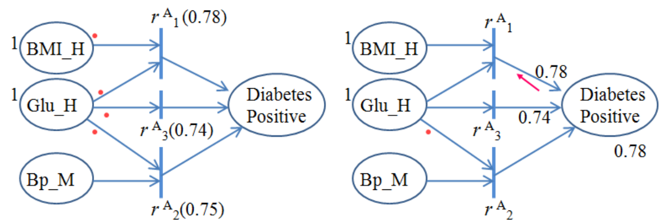
KB^D={r^D_1: Age_H, Diabetes_Positive, Area_Rural
→Heart Disease_Present (0.75) }

Suppose there is a 55 (Age_H) years old lady lived in a rural area, who has plasma glucose level 187 (Glu_H), BMI 35 (BMI_H). To clearly illustrate the FPN, propositions are used as names of places, certainty factors of places and transitions are put beside them.

In this case, *Known* ={Age_H, Glu_H, BMI_H}, *Certainty*={1,1,1}, Target ={Diabetes_Positive}. Suppose the threshold for rules is 0.7.

After the Chair announces the start of argumentation, agent B makes a proposal [Propose|Diabetes_Positive |0.7|{ r^B_1}] (*confidence-records* of supporting rules are omitted for simplification).

Agent A has a higher confidence proposal [Propose |Diabetes_Positive |0.78|{ r^A_1}]. Fig. 7 shows the FPN after initialising and after reasoning.



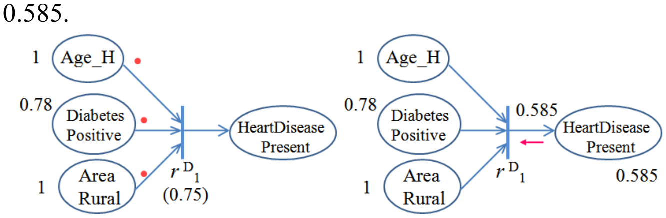
(a) After initialising (b) After reasoning

Fig. 7. FPN of Agent A

Agent B has no better proposals, agent A wins this round of argumentation. No agent has arguments on A's proposal, the chair announces that the patient has diabetes with 0.78 certainty.

The chair starts an argumentation for new possible disease. Agent B makes a proposal [Propose|HeartDisease_Present |0.72|{ r^B_2}]. Agent C has a rule r^C_3 (0.5) which is the same as r^B_2(0.72). After updating with the *confidence-records* from B, the certainty factor of r^C_3 is lower than the threshold 0.7. Therefore, C generates dialogue [Disagree|{ r^C_3}] with the updated *confidence-records*. Both B and C have their knowledge updated during the interaction. They cannot make other proposals and have nothing to argue, so they leave the argumentation stage.

D has insufficient information to make a proposal so it asks a question [Question|Area_Rural]. After the patient confirms that she lives in rural area, D proposes [Propose |HeartDisease_Present |0.585| r^D_1]. The corresponding FPN is shown in Fig. 8. No more arguments on this proposal, the chair announces that the patient has heart disease with certainty factor 0.585.



(a) After initialising (b) After reasoning

Fig. 8. FPN of Agent D

The argumentation dialogues among agents contain all the details such as *confidence-records*. The dialogues are also presented in human readable format for doctors and patients to participate in the collaborative diagnosis. Fig. 9 shows a sample interface for patients, where propositions are described by natural language, certainty factor (v) is described by fuzzy description of {no risk($v \leq 0.5$), low risk($0.5< v \leq 0.65$), medium risk ($0.65< v \leq 0.75$), high risk($v > 0.75$)}.

Through the argumentation, agents' knowledge is validated by other agents. The final diagnosis is agreed by all agents which removes the local bias. New knowledge may be created during the argumentation. For example, agent D obtains new knowledge Age_H, Glu_H, BMI_H, Area_Rural → HeartDisease_Present (0.585) from r^{A}_{1} and r^{D}_{1}. If it encounters such patient later, it will propose heart disease check up.

Chair:	Start diagnosis....
Agent B:	I propose medium risk of diabetes, because high age --> diabetes
Agent A:	I propose high risk of diabetes, because high glucose, high BMI --> diabetes
Chair:	The patient has high risk of diabetes. Start diagnosis....
Agent B:	I propose medium risk of heart disease, because high age --> heart disease
Agent C:	I disagree with high age --> heart disease
Agent D:	Dear patient, you live in rural area?
Patient:	Yes.
Agent D:	I propose low risk of heart disease, because high age, diabetes, rural area --> heart disease
Chair:	The patient has low risk of heart disease.
Chair:	Start diagnosis....
Chair:	Finish.

Fig. 9. Sample dialogues

VI. CONCLUSION

This paper proposed a computing model for collaborative medical diagnosis through multiple agent argumentation.

We applied fuzzy petri nets as the knowledge model to handle the uncertainty in reasoning. Uncertain rules are common for medical agents, especially when the knowledge is mined from databases. Two changes are made to FPNs: to put tokens on arcs (not in places as that of the usual FPNs), and to give places capability in controlling the propagation of tokens. The changes make the FPN more suitable for logical inference and more flexible in applications where knowledge is frequently updated.

We designed algorithms to automate the agent argumentation dialogues. Through argumentation, the agents can share information, critique and verify each other's knowledge, learn from each other and collaboratively come up with a mutually agreed diagnosis based on the collective expertise of multiple agents.

The argumentative diagnosis model proposed in this paper provides a channel for better patient symptoms understanding, extensive knowledge sharing and learning, as well as more accurate diagnosis.

REFERENCES

[1] Bupa Health Pulse 2011: International Healthcare Survey - Global Trends, Attitudes and Influences, http://www.bupa.com.au/staticfiles/BupaP3/Health%20and%20Wellness/MediaFiles/PDFs/Bupa_health_pulse_2011.pdf, last visited on 25/11/2013.

[2] www.webmd.com

[3] www.mayoclinic.com

[4] www.chunyuyisheng.com

[5] http://www.doctorspring.com/

[6] S. Gupta & S. Pujari, "A multi-agent system (MAS) based scheme for health care and medical diagnosis system", *International Conference on Intelligent Agent & Multi-Agent Systems*, 2009, pp.1-3.

[7] Q. Yang & J. S. Shieh, "A multi-agent prototype system for medical diagnosis", *3rd International Conference on Intelligent System and Knowledge Engineering*, 2008, pp.1265-1270.

[8] M. Oprea, "MEDICAL_MAS:An agent-based system for medical diagnosis", *Artificial Intelligence Applications and Innovations III, IFIP International Federation for Information Processing*, vol.296, 2009, pp. 225-232.

[9] J. Andriessen, "Arguing to learn", In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences*, New York: Cambridge University Press, 2006, pp. 443-459.

[10] T. Murata, "Petri nets: Properties, analysis and applications", Proceedings of the IEEE, vol. 77, no. 4, 1989, pp. 541-580.

[11] S. M. Chen, J. S. Ke & J. F. Chang, "Knowledge representation using fuzzy petri nets", IEEE Transactions on Knowledge and Data Engineering, vol. 2, no. 3, 1990, pp.311-319.

[12] C.G.Looney, "Fuzzy petri nets for rule-based decisionmaking", IEEE Transactions on Systems, Man, and Cybernetics, vol. 18, no. 1, 1988, pp.178-183.

[13] D. T. Gilbert, "How mental systems believe", *American Psychologist*, vol. 46, no. 2, 1991, pp. 107–119.

[14] K. M. Lee, "Why presence occurs: Evolutionary psychology, media equation, and presence", *Presence: Teleoperators and Virtual Environments*, Vol. 13, No. 4, 2004, pp. 494–505.

[15] G. Mantovani, "Virtual reality as a communication environment: Consensual hallucination, fiction, and possible selves", *Human Relations*, vol. 48, no. 6, 1995, pp. 669-683.

[16] D. N. Walton & E. C. W. Krabbe, *Commitment in dialogue: Basic concepts of interpersonal reasoning*, State University of New York Press, Albany, NY, USA, 1995.

[17] P. Mcburney & S. Parsons, "Games that agents play: A formal framework for dialogues between autonomous agents", *Journal of Logic, Language and Information*, vol. 11, no. 3, 2002, pp. 315-334.

[18] P. McBurney & S. Parsons, "Locutions for argumentation in agent interaction protocols", AAMAS '04 *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 3, 2004, pp. 1238 - 1239.

[19] S. Heras, M. Rebollo & V. Julian, "A dialogue game protocol for recommendation in social networks", in E. Corchado, A. Abraham and W. Pedrycz (Eds): HAIS'08 *Proceedings of the 3rd International Workshop on Hybrid Artificial Intelligence Systems, Lecture Notes in Artificial Intelligence* 5271, 2008, pp. 515-522.

[20] L. Yao, J. Xu, J. Li, X. Qi, "Evaluating the valuable rules from different experience using multiparty argument games", IEEE/WIC/ACM *International Conferences on Web Intelligence and Intelligent Agent Technology*, 2012 , pp. 258 – 265.

[21] L. Yao, X. Qi, J. Yuan, "Dialectic analysis model: An approach to assist decision-making based on dialectic thinking", *Second International Conference on Intelligent System Design and Engineering Application*, 2012 , pp. 235 – 242.

[22] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes

[23] G. Karegowda, V. Punya, M. A. Jayaram, A. S. Manjunath, "Rule based classification for diabetic patients using cascaded k-means and decision tree C4.5", *International Journal of Computer Applications*, vol. 45, no. 12, 2012, pp. 45-50.

[24] R. Agrawal, T. Imielinski & A. Swami, "Mining association rules between sets of items in large databases", *Proceedings of the ACM SIGMOD Conference on Management of Data*, Washington, D.C., 1993, pp. 207-216.

[25] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, 1994, pp. 487-499.

[26] http://www.cs.waikato.ac.nz/ml/weka/

[27] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)

[28] M. A. Jabbar , P. Chandra , B. L. Deekshatulu, "Heart disease prediction system using associative classification and genetic algorithm", *International Conference on Emerging Trends in Electrical, Communication and Information Technologies*, vol. 1, 2012, pp. 183-192.