World Applied Sciences Journal 3 (2): 244-253, 2008
ISSN 1818-4952
© IDOSI Publications, 2008

Multi-Expert Disease Diagnosis System over Symptom Data Grids on the Internet

Amin Milani Fard, Hossein Kamyar and Mahmoud Naghibzadeh

Department of Computer Engineering, Ferdowsi University of Mashhad, Iran

Abstract: Nowadays practical databases become so huge containing billions of records which are needed to be statistically analyzed for useful knowledge extraction. By the emergence of the new Grid technology, simultaneous effective use of distributed computational and informational resources has been provided. Users in a data grid can access information easily without knowing the resource position. Knowledge discovery from databases is the process of nontrivial extraction of implicit, previously unknown and potentially useful information from data. In this work a new grid knowledge discovery approach was designed and based on the idea a sample multi-expert desease diagnosis system has been developed. Game theory and soft computing mechanisms were also applied to our method for the improvement of knowledge mining and representation.

Key words: Association rule mining • data grids • data mining • Dempster-Shafer theory • game theory • multi-agent systems • soft computing

INTRODUCTION

Considering the fast growth of data contents size and variety, finding useful information from collections of data have been extensively investigated in past decades. Practical databases become so huge containing billions of records which are needed to be statistically analyzed for useful knowledge extraction. Therefore knowledge discovery from databases (KDD) techniques were introduced as the process of nontrivial extraction of implicit, previously unknown and potentially useful information from data [1]. Nowadays Grid technology [2] provides us simultaneous effective use of distributed computational and informational resources. Three main types of this technological phenomenon are known as resource discovery grids, computational grids and data grids. Research works are mostly focused on algorithms regarding discovery grids and computational grids, however in this works we focused on data grids in which users can access information easily without knowing the resource position.

In this project a multi-expert system for desease diagnosis knowledge discovery in an information Grid environment has been designed and developed based on our proposed general architecture. The structure of the paper is as follows. Section 2 describes related works on expert systems and distributed information retrieval in brief. The proposed multi-agent systems architecture and agents behaviours are declared in section 3; and a sample run result is dedicated to section 4. We finalize our work with a conclusion and future work part in section 5.

Related works: Expert sytems (ES) are software tools to help experts and specialists for patial knowledge substitution and decision making. These systems work based on a knowledge base composed of *Facts* and *Rules* and an inference engine. MYCIN [3] was actually the first successful ES designed in 1970 at Stanford University with the perpose of assisting physician in diognasis of infectious blood diseases and antibiotics. The system operated using a fairly simple inference engine and a knowledge base of almost 500 rules. It was never actually used in practice not because of any weakness in its performance but much because of ethical and legal issues related to the use of computers in medicine, in case it gives a wrong diagnosis.

There are two main methods of reasoning when using inference rules, forward chaining and backward chaining. Forward chaining [4] starts with the available data and uses inference rules to extract more data until reaching goal. Because the data determines which rules are selected and used, this method is called data-driven. Backward chaining [4], on the other hand, starts with a list of goals (or a hypothesis) and works backwards from the consequent to the antecedent to see if there is data available that will support any of these consequents. Because the list of goals determines which rules are

Corresponding Author: Mr. Amin Milani Fard, Department of Computer Engineering, Ferdowsi University of Mashhad, Iran

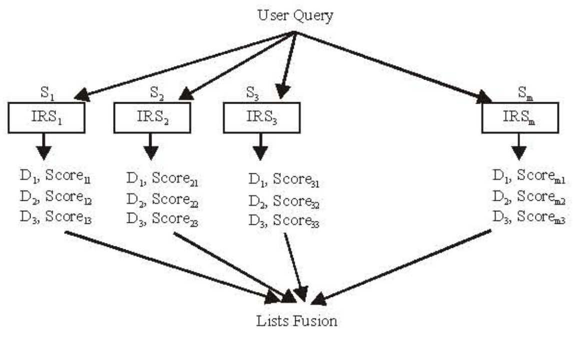

Fig. 1: An overview of information retrieval from multiple sources

selected and used, this method is called goal-driven. Both of the method explaind are often employed by expert systems. With the groth of distributed prosessing approaches, methods for combining multiple expert systems knowledge, known as multi-expert systems, have been widely studied in the last decade [5].

Distributed information retrieval (DIR) aims at finding information in scattered sources located on different servers on a network. This problem, also known as *federated search*, involves building resource descriptions for each database, choosing which databases to search for particular information and merging retrieved results into a single result list [6, 7]. Some applications of information retrieval from multiple sources include meta-search engines, distributed genomic search, newsletter gathering and etc. DIR includes four sub-problems of resource description, resource selection, query translation and result merging [7]. The resource description problem is how to learn and describe the topics covered by each different database. The resource selection means, given a query and a set of resource descriptions, how to select a set of resources to search. The query translation means mapping a given information need in some base representation, automatically to query languages appropriate for the selected databases. Finally, after result lists have been returned from the selected databases, result merging is used to integrate them into a single ranked list.

Let us consider *m* information servers, denoted by S_i and i=1...m; each of which provided with its own information retrieval system (IRS). Here we assume that the retrieval engines produce weighted lists of retrieved documents as an indicator of the estimated relevance of each document with respect to the query. In order to provide user with a single ordered list of relevant documents, the individual lists produced by IRSs must be fused [8]. This fusion is not an easy problem as IRSs produced their result with different criteria and statistics such as average document length, text frequency inverse document frequency and etc [9].

The proposed system: Implementation platform: Our proposed multi-agent system architecture is based on *Java Agent Deelopment* (JADE) framework [10]. JADE is a software development framework aimed at developing multi-agent systems and applications in which agents communicate using FIPA[1] Agent Communication Language (ACL) messages and live in containers which may be distributed to several different machines. JADE uses RMI[1] method for communication. One of the most important characteristics of this tool is that programmer is not required to handle variables and functions concurrency as it is done automatically by the system.

JADE is capable of linking Web services and agents together to enable semantic web applications. A Web service can be published as a JADE agent service and an agent service can be symmetrically published as a Web service endpoint. Invoking a Web service is just like invoking a normal agent service. Web services' clients can also search for and invoke agent services hosted within JADE containers.

The Web Services Integration Gateway (WSIG) [11] uses a Gateway agent to control the gateway from within a JADE container. Interaction among agents on different platforms is achieved through the Agent Communication Channel. Whenever a JADE agent sends a message and the receiver lives on a different agent platform, a Message Transport Protocol (MTP) is used to implement lower level message delivery procedures [12]. Currently there are two main MTPs to support this inter-platform agent communication - CORBA IIOP-based and HTTP-based MTP.

Considering large-scale applications over separated networks, agent communications has to be handled behind firewalls and Network Address Translators (NATs), however, the current JADE MTP do not allow agent communication through firewalls and NATs. Fortunately, the firewall/NAT issue can be solved by using the current JXTA implementation for agent communication [13].

JXTA is a set of open protocols for P2P networking. These protocols enable developers to build and deploy P2P applications through a unified medium [14]. Obviously, JXTA is a suitable architecture for implementing MTP-s for JADE and consequently

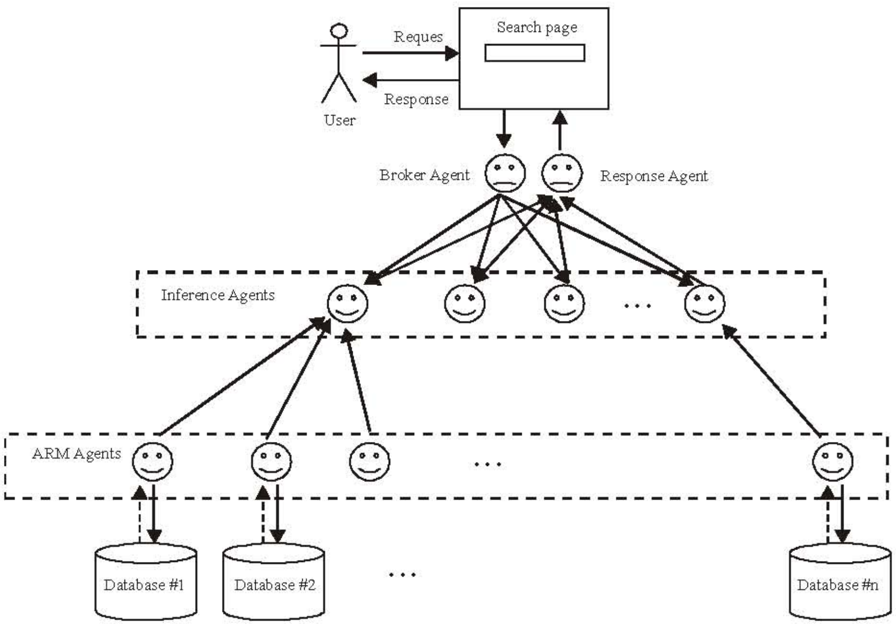

Fig. 2: The proposed multi-agent knowledge mining chitecture

JADE agent communication within different networks can be facilitated by incorporating JXTA technology into JADE [13].

A multi-agent system for intelligent information retrieval in heterogenous networks have been proposed in a previous work [15] and upon that architecture a microorganism DNA pattern search trough web-based genomic engine [16] and a web-based criminal face recognition system [17] was proposed. In this work, we also use a same infrastructure to solve agnet communication problem in the Grid using XML web servises. In our proposed architecture, we use five different types of agents, each having its own characteristics as the followings:

1. Manager Agent (MA): The MA has the responsibility of managing the whole system encluding other agent's creation. The creation node determination is influenced by different criterion such as CPU power, available processor load, total memory amount, used memory amount, traffic around node and etc.

2. Broker Agent (BA): These agents will deliver the query from user to Inference Agents. The query is in the form of facts to be included in IA knowledge base.

3. Association Rules Miner Agent (ARMA): ARMAs are used to discover useful association rules and convert them to First Order Logic (FOL) to be included in the local IA knowledge base. The local IA is the one which is responsioble for its local LAN ARM agents.

4. Inference Agent (IA): Inference Agents use FOL based rules gathered by ARMAs, FOL based facts from BA and apply the inference mechanism (here forward chaining) and return their inference result to the response agent.

5. Response Agent (RA): This agent has the responsibility to show the result of retrieved information. To do so, RA collects IAs results and combine them using Dempster-Shafer method and and then writes them on the screen ordered by relation percentage.

The proposed multi-agent architecture is shown in Fig. 2.

As the most innovative design parts were done on ARMA, IA and RA, we focus on their details in the following sections.

Arm agent behaviour: Having created each ARMA, they would start mining according to their predefined behaviour. The well-known *Apriori* Association rule mining algorithm [18] is used for this matter. Two main parameters in this algorithm are *MinSup* and *MinCon*, which denote minimum acceptable support and confidence respectively. Althogh mostly these parameters are set to 50%, in our thesis we used a game theory-based mechanism to define almost

```
1)  $L_1$ = {large 1-itemsets};
2)  for (k = 2; $L_{k-1} \neq \emptyset$; k++) do begin
3)  $C_k$ = apriori-gen ($L_{k-1}$); // new candidates
4)  forall transactions $t \in D$ do begin
5)  $C_t$ = subset ($C_k$,t); // candidates contained in t
6)  forall candidates $c \in C_t$ do
7)  c.count++;
8)  end
9)  $L_k = \{c \in C_k \backslash c.count \geq minsup\}$
10) end
11) Answer = $\cup_k L_k$;
```

Fig. 3: Apriori Algorithm

optimal *MinSup* and *MinCon* values for them. We continue by *Apriori* algorithm definition and game theory basics.

Apriori algorithm: The *Apriori* algorithm [18] computes the frequent itemsets in the database through several iterations. Iterations *i* computes all frequent *i*-itemsets (itemsets with *i* elements). Each iteration has two steps: *candidate generation* and *candidate counting and selection*. In the first phase of the first iteration, the generated set of candidate itemsets contains all *i*-itemsets (i.e, all items in the database). In the counting phase, the algorithm counts their support searching again through the whole database. Finally, only *i*-itemsets (items) with s above required threshold will be selected as frequent. Thus, after the first iteration, all frequent *i*-itemsets will be known.

Basically, all pairs of items are candidates. Based on knowledge about infrequent itemsets obtained from previous iterations, the *Apriori* algorithm reduces the set of candidate itemsets by pruning -*a priori*-those candidate itemsets that cannot be frequent. The pruning is based on the observation that if an itemset is frequent all its subsets could be frequent as well. Therefore, before entering the candidate-counting step, the algorithm discards every candidate itemset that has an infrequent subset [19].

Game theory approach: Game theory [20] provides us with the mathematical tools to understand the possible strategies that utility-maximizing agents might use when making a choice. It is mostly concerned with modeling the decision process of rational humans, a fact that should be kept in mind as we consider its applicability to multi-agent systems. The simplest type of game considered in game theory is the *single-shot simultaneous-move* game. In this game all agents must take one action simultaneously. Each agent receives a utility that is a function of the combined set of actions. This is a good model for the types of situations often faced by agents in a multi-agent system where the encounters mostly require coordination [21].

In the one-shot simultaneous-move game we say that each agent *i* chooses a strategy $s_i \in S_i$, where Si is the set of all strategies for agent *i*. These strategies represent the actions the agent can take. When we say that *i* chooses strategy s_i we mean that it chooses to take action s_i. The set of all strategies chosen by all the agents is the strategy profile for that game and it is denoted by $s \in S \equiv \times^{I}_{i=1} S_i$.

Once all the agents make their choices and form the strategy profile s then each agent *i* receives a utility which is given by the function $u_i(s)$. Notice that a player's utility depends on the choices made by all the agents. A Game [22] is a 3-tuple G = {N, α,Π} Where:

- N is the number of players
- $\alpha = \{\alpha^i\}_{i=1...N}$; $\alpha^i = \{\alpha^i_1, \ldots \alpha^i_{Mi}\}$ is the set of actions available to player *i*
- Π: $\times_i \alpha^i$?R^N is the Payoff function, i.e. assigns each player a real number payoff for any combination of all players actions.

It is possible that a player will choose randomly between its action choices, using different prior probabilities for each choice. These types of strategies are called mixed strategies and they are a probability distribution over an agent's actions. We say that a mixed strategy for agent *i* is $\sigma_i \in \Sigma_i \equiv P(S_i)$ where $P(S_i)$ is the set of all probability distributions over the set of pure strategies S_i [21]. The *Nash equilibrium* in an n-player game is a set of strategies, $\Sigma = \{\sigma^1, \ldots, \sigma^n\}$; such that, given that for all I player *i* plays σ^i, no player *j* can get a higher payoff by playing a strategy other then σ^j [23]. It has been shown that every game has at least one Nash equilibrium, as long as mixed strategies are allowed. If the system is in equilibrium then no agent will be tempted to take a different action.

A good classic example for equilibrium point in a game is *two prisoners dilemma* in which there are two prisoners kept in seprated cells. If both confess they would be sentenced three years in jail. If one confess and the other one don't (testify the one who confess is guilty) the first one goes 4 years in jail and another one would release. And if none of them confess they would be sentenced f or a less important crime and will go one year in the jail each. The matrix in Fig. 4 shows that the best action a player can do in not to confess and the Nash equilibrium is (don'confess, don'confess).

	Confess	don'tconfess
Confess	3-3	4-0
don't Confess	0-4	1-1

Fig. 4: two prisoners' dilemma

Our proposed game model is to find best *MinCon* and *MinSup* for ARMAs so that the total system performance improves. To model the game let's assume in the real world a computer conference is goint to be held and our players' aim is to publish at least a paper in the conference anyway possible. We also assume players are multi expert so they can publish paper in different fields of computer science and engineering. Also we know that in each field only a limited number of papers would be accepted. As a result each player tries to submit any paper in any field he can and with any quality. So if another player does not present a paper in a particular filed that the other one did, there would be a 100% chance of acceptance in contrast with the one who did not proposed. However, if both players submit papers since we do not have any information about the quality, there would be an equal 50% chance for each player. Therefore it is trivial that the game equilibrium is the case that each player submits any paper and with any quality he can. (We assume different paper submission does not have negative effects such as time wasting and etc.)

The scenario is somehow the same for our ARMAs. ARM agents which are in charge of gathering first level knowledge, will not get nay payoff if can not present any amount of knowledge. This is considered by choosing a suitable value for *MinCon* and *MinSup* in the run-time. So in contrast with suggested 50% confidence and 50% support, our game-theory based method will select the highest possible confidence and support even if they are under 50% for instance an association rule with 35% confidence.

This mechanism assures that all ARMAs do their best to propose something as a first level knowledge. One of the best examples that this approach would be successful is Diagnosis Expert Systems in which diagnostic test among different experts are different and maybe symptoms represent different desease with respect to conditions such as country, race, age and etc. The ARMA game matrix in Fig. 5 shows agents' payoff in winning percentage regarding their knowledge presentation.

Note that players also try to maximize their knowledge presentation *lift* (progress ratio) which is a measure of strength for an association.

	Present Knowledge	Don't Present Knowledge
Present Knowledge	50%-50%	100%-0%
Don't Present Knowledge	0%-100%	0%-0%

Fig. 5: proposed ARMA game matrix

Inference agent behaviour: The extracted association rules from ARMAs will be sent to IAs with their corresponding support and confidence. These *rules* plus the *fact* query sent by the BA will construct the IA knowledge base. By this time agents are able to inference results. Expert systems mostly use backward chaining; however, we decided to use forward chaining due to our system model which is goal-driven. The forward chaining starts by adding new fact *P* to the knowledge base and finds all conditional combinations having *P* in assumption.

Response agent behaviour: The response agent finally obtains all results from IAs and then combines them using Dempster-Shafer theory. These results will be then proposed to the user in an ordered list according to their relevancy. Now if the user is interested in selecting some particular results, the user profiling through system feedback is done using a hybrid neuro-fuzzy mechanism.

Dempster-Shafer theory of evidence: The Dempster-Shafer (DS) theory was first introduced by Dempster [24] and then extended by Shafer [25], but the kind of reasoning the theory uses can be found as far back as the seventeenth century. This theory is actually an extension to classic probabilistic uncertainty modeling. Whereas the Bayesian theory requires probabilities for each question of interest, belief functions allow us to base degrees of belief for one question on probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ from probabilities will depend on how closely the two questions are related. This theory has been used in information retrieval in [26-29].

This theory is a generalizatiuon to the Bayesian Theory of Probability. The most significant differences between DS theory and probability theory are the explicit representation of uncertainty and evidence combination mechanism in which made it more effective in document processing fields [26]. In information retrival the uncertainty ocuurs in three cases:

- Existence of different evidences regarding relation of a document to a query
- Unknown number of evidences regarding relation of a document to a query
- Existance of incorrect evidences regarding relation of a document to a query

In the DS theory of evidence, *Belief* is a value to express certainty of a proposition. This belief is calculated with respect to a density function *m: ρ(U) → [0,1] called Basic Probability Assignment* (BPA), where m(A) represents *Partial Belief* amount of A.

$$m(\phi)=0$$
$$\sum_{A\in U} m(A)=1$$

To measure the *Total Belief* amount of A⊂U the belief function is defined as:

$$Bel(A)=\sum_{\forall B\subset A} m(B)$$

Shafer defined *doubt* amount in A as the belief in A′ and the *Plausibilty* function as the total belief amount in A

$$Dou(A)=Bel(A)$$
$$Pl(A)=1-Dou(A)=$$
$$\sum_{B\subseteq U} m(B)-\sum_{B\subseteq A} m(B)=\sum_{B\cap A\neq\phi} m(B)$$

Pl(A) is actually the high boundary of belief in A so that the correct belief in A is in the interval of [Bel(A), Pl(A)] as shown in Fig. 6.

Dempster's rule of combination is a generalization of Bayes' rule. This rule strongly emphasises the agreement between multiple sources and ignores all the conflicting evidence through a normalization factor. Let m1 and m2 are the BPAs in a frame of discernment. The combination BPA is calculated in the following manner:

$$m(A)=m_1\otimes m_2=\frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{\sum_{B\cap C\neq\phi} m_1(B)m_2(C)}$$

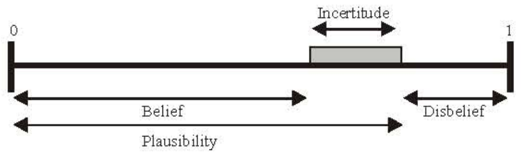

Fig. 6: Graphical view of the belief concept

Soft computing method in user behaviour modeling: Soft computing-based methods which are intelligent hybrid combination of techniques such as fuzzy logic and control, artificial neural networks, evolutionary algorithms and etc, are widely used in data mining and web content search in current years sometimes known as *Soft Web Mining* [30]. This is due to data properties in web environment such as being unlabeled, heterogeneous, semi-structured, time variant and etc.

Since users of information retrieval systems have nonlinear behaviour, it is not possible to model their profile using classic methods. Different user profiling approaches have been investigated in [31, 32] and one of the suggested models are based on neuro-fazzy systems for information filtering and personalization. In our project we take advantage of ANFIS (Adaptive-Network-based Fuzzy Inference Systems) [33] to satisfy the demand. This network is more flexible in contrast with the simple ones using a hybrid of fuzzy logic approach. It has been shown that ANFIS is equivalent to a fuzzy system.

Our suggested network has two inputs for *support* and *confidence* values and membership fuctions are partitioned to *low*, *medium* and *high*. In user behaviour modeling, we consider a neural network for each rule with the same THEN clause and train the network considering rule selection or not. Therefore, the user can personalize the knowledge retrieval system regarding desired support and confidence for particular rules.

System run sample: In order to show the knowledge mining process in our syatem, we take advantage of disease symptoms database[2]. Since using diseases with completely seprated symptoms are not desirable as a test case, we choosed those with common signs which are: *Cold*, *Flu*, *Bronchitis*, *Allergy*, *Asthma*, *Sinusitis*, *Strepthroat* and *Gastroenteritis*. At the first step, the extracted association rules done by ARMAs are as bellow:

{runnynose} => {cold}	sup: 0.58	conf: 1.0
{sorethroat} => {cold}	sup: 0.55	conf: 1.0
{headache} => {cold}	sup: 0.62	conf: 1.0
{fever} => {cold}	sup: 0.51	conf: 1.0
{cough} => {cold}	sup: 0.72	conf: 1.0
{sorethroat fever} => {cold}	sup: 0.34	conf: 1.0
{fever headache} => {cold}	sup: 0.20	conf: 1.0
{runnynose headache} => {cold}	sup: 0.24	conf: 1.0
{sorethroat runnynose} => {cold}	sup: 0.37	conf: 1.0
{sorethroat headache} => {cold}	sup: 0.24	conf: 0.9
{cough fever} => {cold}	sup: 0.41	conf: 1.0
{cough runnynose} => {cold}	sup: 0.37	conf: 1.0
{sorethroat cough} => {cold}	sup: 0.34	conf: 1.0
{fever runnynose} => {cold}	sup: 0.27	conf: 1.0
{cough headache} => {cold}	sup: 0.41	conf: 1.0
{cough fever runnynose} => {cold}	sup: 0.24	conf: 1.0
{sorethroat cough fever} => {cold}	sup: 0.27	conf: 1.0
{sorethroat fever runnynose} => {cold}	sup: 0.24	conf: 1.0
{sorethroat cough runnynose} => {cold}	sup: 0.24	conf: 1.0
{sorethroat cough fever runnynose}=>{cold}	sup: 0.24	conf: 1.0

{chills} => {flu}	sup: 0.62	con: 1.0
{runnynose} => {flu}	sup: 0.58	con: 1.0
{fever} => {flu}	sup: 0.51	con: 1.0
{muscleache} => {flu}	sup: 0.55	con: 1.0
{cough} => {flu}	sup: 0.72	con: 1.0
{chills muscleache} => {flu}	sup: 0.24	con: 1.0
{muscleache cough} => {flu}	sup: 0.34	con: 1.0
{cough runnynose} => {flu}	sup: 0.37	con: 1.0
{chills cough} => {flu}	sup: 0.41	con: 1.0
{fever runnynose} => {flu}	sup: 0.27	con: 1.0
{cough fever} => {flu}	sup: 0.41	con: 1.0
{chills runnynose} => {flu}	sup: 0.24	con: 1.0
{muscleache runnynose} => {flu}	sup: 0.37	con: 1.0
{muscleache fever} => {flu}	sup: 0.34	con: 1.0
{cough fever runnynose} => {flu}	sup: 0.24	con: 1.0
{muscleache cough runnynose} => {flu}	sup: 0.24	con: 1.0
{muscleache fever runnynose} => {flu}	sup: 0.24	con: 1.0
{fever runnynose} => {muscleache flu}	sup: 0.24	con: 0.875
{muscleache cough fever} => {flu}	sup: 0.27	con: 1.0
{muscleache cough fever runnynose}=> {flu}	sup: 0.24	con: 1.0

{itchy} => {allergy}	sup: 0.62	conf: 1.0
{sneeze} => {allergy}	sup: 0.79	conf: 1.0
{runnynose} => {allergy}	sup: 0.58	conf: 1.0
{itchy runnynose} => {allergy}	sup: 0.24	conf: 1.0
{sneeze itchy} => {allergy}	sup: 0.44	conf: 1.0
{itchy sneeze} => {allergy}	sup: 0.44	conf: 0.72
{sneeze runnynose} => {allergy}	sup: 0.44	conf: 1.0
{runnynose sneeze} => {allergy}	sup: 0.44	conf: 0.76
{itchy} => {allergy}	sup: 0.62	Con
{wheezing} => {bronchitis}	sup: 0.72	conf: 1.0
{breathshortness} => {bronchitis}	sup: 0.62	conf: 1.0
{fever} => {bronchitis}	sup: 0.51	conf: 1.0
{fever wheezing} => {bronchitis}	sup: 0.41	conf: 1.0
{breathshortnesswheezing}=>{bronchitis}	sup: 0.41	conf: 1.0
{fever breathshortness}=>{bronchitis}	sup: 0.20	conf: 1.0
{eyepain} => {sinusitis}	sup: 0.51	con: 1.0
{eyepain} => {sinusitis}	sup: 0.51	con: 1.0
{headache} => {sinusitis}	sup: 0.79	con: 1.0
{cough} => {sinusitis}	sup: 0.62	con: 1.0
{cough headache} => {sinusitis}	sup: 0.44	con: 1.0
{cough eyepain} => {sinusitis}	sup: 0.24	con: 1.0
{eyepain headache} => {sinusitis}	sup: 0.44	con: 1.0
{eyepain} => {sinusitis}	sup: 0.51	con: 1.0
{headache} => {strepthroat}	sup: 0.79	con: 1.0
{sorethroat} => {strepthroat}	sup: 0.58	con: 1.0
{fever} => {strepthroat}	sup: 0.72	con: 1.0
{fever headache} => {strepthroat}	sup: 0.55	con: 1.0
{sorethroat headache} => {strepthroat}	sup: 0.44	con: 1.0
{sorethroat fever} => {strepthroat}	sup: 0.34	con: 1.0
{sorethroat fever headache} => {strepthroat}	sup: 0.24	con: 1.0

{breathshortness} => {asthma}	sup: 0.58	con: 1.0
{wheeze} => {asthma}	sup: 0.79	con: 1.0
{cough} => {asthma}	sup: 0.62	con: 1.0
{wheeze cough} => {asthma}	sup: 0.44	con: 1.0
{wheeze breathshortness} => {asthma}	sup: 0.44	con: 1.0
{cough breathshortness} => {asthma}	sup: 0.24	con: 1.0

The produced knowledge base by the IA is then

IF has ?x itchy THEN has ?x allergy(S:0.62,C:1.0)
IF has ?x sneeze THEN has ?x allergy(S:0.79,C:1.0)
IF has ?x runnynose THEN has ?x allergy(S:0.58,C:1.0)
IF has ?x itchy has ?x runnynose THEN has ?x allergy(S:0.24,C:1.0)
IF has ?x sneeze has ?x itchy THEN has ?x allergy(S:0.44,C:1.0)
IF has ?x itchy has ?x sneeze THEN has ?x allergy(S:0.44,C:0.72)
IF has ?x sneeze has ?x runnynose THEN has ?x allergy(S:0.44,C:1.0)
IF has ?x runnynose has ?x sneeze
THEN has ?x allergy(S:0.44,C:0.76)
IF has ?x wheezing THEN has ?x bronchitis(S:0.72,C:1.0)
IF has ?x breathshortness THEN has ?x bronchitis(S:0.62,C:1.0)
IF has ?x fever THEN has ?x bronchitis(S:0.51,C:1.0)
IF has ?x fever has ?x wheezing
THEN has ?x bronchitis(S:0.41,C:1.0)
IF has ?x breathshortness has ?x wheezing
THEN has ?x bronchitis(S:0.41,C:1.0)
IF has ?x fever has ?x breathshortness
THEN has ?x bronchitis(S:0.20,C:1.0)
IF has ?x chills THEN has ?x flu(S:0.62,C:1.0)
IF has ?x runnynose THEN has ?x flu(S:0.58,C:1.0)
IF has ?x fever THEN has ?x flu(S:0.51,C:1.0)
IF has ?x muscleache THEN has ?x flu(S:0.55,C:1.0)
IF has ?x cough THEN has ?x flu(S:0.72,C:1.0)
IF has ?x chills has ?x muscleache THEN has ?x flu(S:0.24,C:1.0)
IF has ?x muscleache has ?x cough THEN has ?x flu(S:0.34,C:1.0)
IF has ?x cough has ?x runnynose THEN has ?x flu(S:0.37,C:1.0)
IF has ?x chills has ?x cough THEN has ?x flu(S:0.41,C:1.0)
IF has ?x fever has ?x runnynose THEN has ?x flu(S:0.27,C:1.0)
IF has ?x cough has ?x fever THEN has ?x flu(S:0.41,C:1.0)
IF has ?x chills has ?x runnynose THEN has ?x flu(S:0.24,C:1.0)
IF has ?x chills has ?x fever THEN has ?x flu(S:0.20,C:1.0)
IF has ?x muscleache has ?x runnynose
THEN has ?x flu(S:0.37,C:1.0)
IF has ?x muscleache has ?x fever THEN has ?x flu(S:0.34,C:1.0)
IF has ?x cough has ?x fever has ?x runnynose
THEN has ?x flu(S:0.24,C:1.0)
IF has ?x muscleache has ?x cough has ?x runnynose
THEN has ?x flu(S:0.24,C:1.0)
IF has ?x muscleache has ?x fever has ?x runnynose
THEN has ?x flu(S:0.24,C:1.0)
IF has ?x muscleache has ?x cough has ?x fever
THEN has ?x flu(S:0.27,C:1.0)
IF has ?x muscleache has ?x cough has ?x fever has ?x runnynose
THEN has ?x flu(S:0.24,C:1.0)
IF has ?x eyepain THEN has ?x sinusitis(S:0.58,C:1.0)
IF has ?x headache THEN has ?x sinusitis(S:0.79,C:1.0)
IF has ?x cough THEN has ?x sinusitis(S:0.62,C:1.0)
IF has ?x cough has ?x headache THEN has ?x sinusitis(S:0.44,C:1.0)
IF has ?x cough has ?x eyepain THEN has ?x sinusitis(S:0.24,C:1.0)
IF has ?x eyepain has ?x headache
THEN has ?x sinusitis(S:0.44,C:1.0)
IF has ?x headache THEN has ?x gastroenteritis(s:0.62,c:1.0)
IF has ?x muscleache THEN has ?x gastroenteritis(s:0.55,c:1.0)

IF has ?x vomit/diarrhea THEN has ?x gastroenteritis(s:0.72,c:1.0)
IF has ?x fever THEN has ?x gastroenteritis(s:0.51,c:1.0)
IF has ?x vomit/diarrhea has ?x fever
THEN has ?x gastroenteritis(s:0.41,c:1.0)
IF has ?x muscleache has ?x headache
THEN has ?x gastroenteritis(s:0.24,c:1.0)
IF has ?x muscleache has ?x vomit/diarrhea
THEN has ?x gastroenteritis(s:0.34,c:1.0)
IF has ?x fever has ?x headache
THEN has ?x gastroenteritis(s:0.20,c:1.0)
IF has ?x muscleache has ?x fever
THEN has ?x gastroenteritis(s:0.34,c:1.0)
IF has ?x vomit/diarrhea has ?x headache
THEN has ?x gastroenteritis(s:0.41,c:1.0)
IF has ?x muscleache has ?x vomit/diarrhea has ?x fever THEN has ?x gastroenteritis(s:0.27,c:1.0)
IF has ?x headache THEN has ?x strepthroat(s:0.79,c:1.0)
IF has ?x sorethroat THEN has ?x strepthroat(s:0.58,c:1.0)
IF has ?x fever THEN has ?x strepthroat(s:0.72,c:1.0)
IF has ?x fever has ?x headache THEN has ?x strepthroat(s:0.55,c:1.0)
IF has ?x sorethroat has ?x headache THEN has ?x strepthroat(s:0.44,c:1.0)
IF has ?x sorethroat has ?x fever THEN has ?x strepthroat(s:0.34,c:1.0)
IF has ?x sorethroat has ?x fever has ?x headache THEN has ?x strepthroat(s:0.24,c:1.0)
IF has ?x runnynose THEN has ?x cold(s:0.58,c:1.0)
IF has ?x sorethroat THEN has ?x cold(s:0.55,c:1.0)
IF has ?x headache THEN has ?x cold(s:0.62,c:1.0)
IF has ?x fever THEN has ?x cold(s:0.51,c:1.0)
IF has ?x cough THEN has ?x cold(s:0.72,c:1.0)
IF has ?x sorethroat has ?x fever THEN has ?x cold(s:0.34,c:1.0)
IF has ?x fever has ?x headache THEN has ?x cold(s:0.20,c:1.0)
IF has ?x runnynose has ?x headache THEN has ?x cold(s:0.24,c:1.0)
IF has ?x sorethroat has ?x runnynose THEN has ?x cold(s:0.37,c:1.0)
IF has ?x sorethroat has ?x headache THEN has ?x cold(s:0.24,c:0.9)
IF has ?x cough has ?x fever THEN has ?x cold(s:0.41,c:1.0)
IF has ?x cough has ?x runnynose THEN has ?x cold(s:0.37,c:1.0)
IF has ?x sorethroat has ?x cough THEN has ?x cold(s:0.34,c:1.0)
IF has ?x fever has ?x runnynose THEN has ?x cold(s:0.27,c:1.0)
IF has ?x cough has ?x headache THEN has ?x cold(s:0.41,c:1.0)
IF has ?x cough has ?x fever has ?x runnynose THEN has ?x cold(s:0.24,c:1.0)
IF has ?x sorethroat has ?x cough has ?x fever THEN has ?x cold(s:0.27,c:1.0)
IF has ?x sorethroat has ?x fever has ?x runnynose THEN has ?x cold(s:0.24,c:1.0)
IF has ?x sorethroat has ?x cough has ?x runnynose THEN has ?x cold(s:0.24,c:1.0)
IF has ?x sorethroat has ?x has ?x cough has ?x fever has ?x runnynose
THEN has ?x cold(s:0.24,c:1.0)
IF has ?x breathshortness THEN has ?x asthma(s:0.58,c:1.0)
IF has ?x wheeze THEN has ?x asthma(s:0.79,c:1.0)
IF has ?x cough THEN has ?x asthma(s:0.62,c:1.0)
IF has ?x wheeze has ?x cough THEN has ?x asthma(s:0.44,c:1.0)
IF has ?x wheeze has ?x breathshortness THEN has ?x asthma(s:0.44,c:1.0)
IF has ?x cough has ?x breathshortness THEN has ?x asthma(s:0.24,c:1.0)

Now the query containing facts of "headache" and "fever" is added to the KB:

has patient headache
has patient fever

In this sample the two IAs use forward chaining and send the two results to the RA depicted below. Finally in the last phase, RA implies D-S combination method and returns the final result to the user.

IA#1

has patient bronchitis(S:0.51,C:1.0) m=1/55
has patient flu(S:0.51,C:1.0) m=1/55
has patient sinusitis(S:0.79,C:1.0) m=1/55
has patient gastroenteritis(s:0.62,c:1.0) m=1/55
has patient gastroenteritis(s:0.20,c:1.0) m=1/55
has patient strepthroat(s:0.79,c:1.0) m=1/55
has patient cold(s:0.62,c:1.0) m=1/55
has patient cold(s:0.51,c:1.0) m=1/55

IA#2

has patient gastroenteritis(s:0.51,c:1.0) m=1/28
has patient strepthroat(s:0.72,c:1.0) m=1/28
has patient strepthroat(s:0.55,c:1.0) m=1/28
has patient cold(s:0.20,c:1.0) m=1/28

	Bronchitis: 0.0181	Flu: 0.0181	Sinusitis: 0.0181	Gastroenteritis 0.0363	Strepthroat: 0.0181	Cold: 0.0363
m(Gastro)						
Bronchitis: 0	0	0	0	0	0	0
Flu: 0	0	0	0	0	0	0
Sinusitis: 0	0	0	0	0	0	0
Gastroenteritis: 0.0357	0.00064617	0.00064617	0.00064617	0.00129591	0.00064617	0.00129591
Strepthroat: 0.0714	0.00129234	0.00129234	0.00129234	0.00259182	0.00129234	0.00259182
Cold: 0.0357	0.00064617	0.00064617	0.00064617	0.00129591	0.00064617	0.00129591

m(Gastroenteritis)	=	1.0171*0.00129591=0.001318070061
m(Cold)	=	1.0171*0.00129591=0.001318070061
m(Strepthroat)	=	1.0171*0.00129234=0.001314439014
m(Bronchitis)	=	1.0171*0=0
m(Flu)	=	1.0171*0=0
m(Sinusitis)	=	1.0171*0=0

Finally, most probable diognasis would be *Gastroenteritis*, *Cold*, *Strepthroat*, *Bronchitis*, *Flu* and *Sinusitis* respectively.

CONCLUSIONS AND FUTURE WORK

In this thesis, the KDD process and data mining methods was investigated and then a new multi-agent architecture for grid environment have been proposed. The most innovative techniques in our design of the system include game-theory modeling for competitive knowledge extraction, hierarchical knowledge mining and Dempster-Shafer result combination. A distributed desesase diagnosis expert system was designed and implemented upon the proposed method in which results shows its performance in knowledge gathering and inferencing.

Regarding project novelty and open research areas in the field, there are surely good potentials to complete hierarchical knowledge mining process including usage of approximation algorithms in knowledge extraction, modeling the process as a mixed strategic games and finding Nash equilibrium and developing an infrastructure for a high performance gred based search engine.

ACKNOWLEDGEMENTS

Authors would like to thank Dr. Mohsen Kahani and Dr. Mohammad R. Akbarzadeh, for many helpful comments that improved the work and Saied Moghaddaszadeh who contributed in system implementation.

REFERENCES

1. Frawley, W., G. Piatetsky-Shapiro and C. Matheus, 1992. Knowledge Discovery in Databases: An Overview. AI Magazine, pp: 213-228.
2. Foster, I. and C. Kesselman, (eds.) 1999. The Grid: Blueprint for a Future Computing Inf., Morgan Kaufmann Publishers, pp: 105-129.
3. Shortliffe, E.H., 1974. "MYCIN: A rule-based computer program for advising physicians regarding antimicrobial therapy selection". Ph.D. dissertation, Stanford University,
4. Russell, S.J. and P. Norvig, 1995. Artificial Intelligence: A modern Approach, Prentice Hall,
5. Waterhouse, S.R., 1997. "Classification and regression using mixtures of experts", Ph.D., Thesis, Department of Engineering, Cambridge University,
6. Si, L. and J. Callan, 2003. "A Semisupervised Learning Method to Merge Search Engine Results", ACM Transactions on Information Systems, 21(4): 457-491.
7. Callan, J., 2000. "Distributed information retrieval". In Advances in Information Retrieval, W.B. Croft, Ed. Kluwer Academic Publishers, pp: 127-150.
8. Pasi, G. and R.R. Yager, 2000. "Document retrieval from multiple sources of information", in Uncertainty in Intelligent and Information Systems, edited by B. Bouchon-Meunier, R.R. Yager and L.A. Zadeh, World Scientific: Singapore, pp: 250-261.
9. Baeza-Yates, R. and B. Ribeiro-Neto, 1999. "Modern Information Retrieval", Addison-Wesley,
10. Bellifemine, F., G. Caire, T. Trucco and G. Rimassa, 2006. "JADE Programmer's Guide", 21-August-2006.
11. JADE Board, 2005. "JADE WSIG Add-On Guide JADE Web Services Integration Gateway (WSIG) Guide", 03-March-2005.
12. Cortese, E., F. Quarta, G. Vitaglione and P. Vrba, 2002. "Scalability and Performance of the JADE Message Transport System". Analysis of Suitability for Holonic Manufacturing Systems, exp, 2002.
13. Liu, S., P. Küngas and M. Matskin, 2006. "Agent-Based Web Service Composition with JADE and JXTA", Proceedings of the 2006 International Conference on Semantic Web and Web Services, SWWS 2006, Las Vegas, Nevada, USA, June 26-29, 2006.
14. Gradecki, J.D., 2002. "Mastering JXTA: Building Java Peer-to-Peer Applications", JohnWiley& Sons.
15. Amin Milani Fard, Mohsen Kahani, Reza Ghaemi and Hamid Tabatabaee, 2007. "Multi-agent Data Fusion Architecture for Intelligent Web Information Retrieval", International Journal of Intelligent Technology, 2(3): 1305-6417.
16. Mohebbat Mohebbi, R. Mohammad, T. Akbarzadeh and Amin Milnai Fard, 2007. "Microorganism DNA PatternSearch in a Multi-agent Genomic Engine Framework", World Applied Science Journal, 2(5): 1818-4952.
17. Hamid Tabatabaee, Amin Milani Fard, R. Mohammad and T. Akbarzadeh, 2007. "A Multi-agent Framework for Web-based Criminal Face Recognition", The First Joint Congress on Intelligent and Fuzzy Systems, 7th Iranian Conference on Fuzzy Systems and 8th Conference on Intelligent Systems, August 29-31, 2007, Mashhad, Iran

18. Agrawal, R., T. Imielinski and A.N. Swami, 1993. "Mining Association Rules between Sets of Items in Large Databases." SIGMOD. 22(2): 207-16.
19. Mehmed Kantardzic, 2003. “Data Mining: Concepts, Models, Methods and Algorithms”, John Wiley & Sons, ISBN: 0471228524.
20. John von Neumann and Oskar Morgenstern: Theory of Games and Economic Behavior, Princeton University Press,1944.
21. Jos´e M. Vidal, 2003. Learning in Multiagent Systems: An Introduction from a Game-Theoretic Perspective In Eduardo Alonso, editor, Adaptive Agents: LNAI 2636. Springer Verlag,
22. A. Neyman. 1985. Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. Economic Letters, pp: 227-229.
23. Yishay Mor, Claudia V. Goldman and Jeffrey S. Rosenschein, 1996. Learn your Opponent's strategy (in Polynomial Time)!, Adaptation and Learning in Multi-Agent Systems, IJCAI95 Workshop, Montreal, Canada, August 1995, Proceedings. Lecture Notes in Artificial Intelligence Vol. 1042, G. Weiss and S. Sen (eds.) Springer Verlag, 1996.
24. Dempster, A., 1968. A generalization ofba yesian inference. Journal of the Royal Statistical Society, 30: 205-247.
25. Schafer, G., 1976. A mathematical theory of evidence, volume 2702. Princetown University Press.
26. Verikas, A., A. Lipnickas, K. Malmqvist, M. Bacauskiene and A. Gelzinis. 1999. “Soft combination of neural classifiers: A comparative study”. Pattern Recognition Letters, 20: 429-444.
27. Krogh, A. and J. Vedelsby, 1995. “Neural network ensembles, cross validation and active learning”. In G. Tesauro, D.S. Touretzky and T.K. Leen, editors, Advances in Neural Information Processing Systems, volume 7, pp: 231-238. MIT Press, Cambridge, MA, 1995.
28. Kuncheva, L.I., 2004. “Combining Pattern Classifiers: Methods and Algorithms”, John Wiley & Sons, ISBN: 0-471-21078-1,
29. Gros, X.E., 1997. “NDT Data Fusion”, John Wiley & Sons Inc.,
30. Pal, S.K., V. Talwar and P. Mitra, 2002. Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions, IEEE Transactions on Neural Networks, vol. 13, NO. 5, September 2002.
31. Frias-Martinez, E., G. Magoulas, S. Chen and R. Macredie, 2004. “Recent Soft Computing Approaches to User Modeling in Adaptive Hypermedia” in Book: Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, Volume 3137/2004, December 02, 2004, ISBN 978-3-540-22895-0.
32. Frias-Martinez, E., G.D. Magoulas, S.Y. Chen and R.D. Macredie, 2005. Modeling Human Behavior in User-Adaptive Systems: Recent Advances Using Soft Computing Technique. Expert Systems with Applications, 29(2).
33. Jang, J.S.R., 1993. ANFIS: adaptive-network-based fuzzy inference systems, IEEE Transactions on Systems, Man and Cybernetics, 23: 665-685.