Chapter 19
Competitive-Cooperative Automated Reasoning from Distributed and Multiple Source of Data

Amin Milani Fard[1]

Abstract Knowledge extraction from distributed database systems, have been investigated during past decade in order to analyze billions of information records. In this work a competitive deduction approach in a heterogeneous data grid environment is proposed using classic data mining and statistical methods. By applying a game theory concept in a multi-agent model, we tried to design a policy for hierarchical knowledge discovery and inference fusion. To show the system run, a sample multi-expert system has also been developed.

19.1 Introduction

Reasoning is perhaps the most powerful and useful method for information systems which is applied to foundational issues in distributed AI. Considering the fast growth of data contents size and variety, finding useful information from collections of scattered data in a network have been extensively investigated in past decades. Practical databases are now becoming very huge containing billions of records and 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].

In this paper a multi-expert competitive mechanism for knowledge discovery process in heterogeneous distributed information systems based on our previous work [2] is investigated. The structure of the paper is as follows. Section 19.2 describes related works on expert systems and distributed information retrieval in brief. The proposed multi-agent systems architecture and agents behaviors are declared in section 19.3; and a sample run result is dedicated to section 19.4. We finalize our work with a conclusion and future work part in section 19.5.

School of Computing Science, Simon Fraser University, BC, Canada
milanifard@cs.sfu.ca

L. Cao (ed.), Data Mining and Multi-agent Integration, DOI: 10.1007/978-1-4419-0522-2_19,

19.2 Related Work

Expert systems (ES) are software tools to help experts and specialists for partial knowledge substitution and decision making. ES 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 purpose of assisting physician in diagnosis of infectious blood diseases and antibiotics. 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 its diagnosis is wrong.

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 a 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 selected and used, this method is called goal-driven. Both of the methods explained are often employed by expert systems. With the growth of distributed processing approaches, methods for combining multiple expert systems knowledge, known as multi-expert systems (MES), 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.

19.3 The Proposed Approach

In order to reach an organized deduction, a hierarchy is performed. At the first phase, association rules are generated from each database. In the second phase rules and facts compose a knowledge base and makes local deduction. The third phase combines these results into a single list ordered by relevancy.

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

[1] Foundation for Intelligent Physical Agents (http://www.fipa.org)

which may be distributed to several different machines. JADE uses RMI[2] 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. 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.

Since we aim to design a large-scale knowledge mining system for heterogeneous separated networks, agent communications has to be handled behind firewalls and Network Address Translators (NATs). Although the current JADE MTP does not allow agent communication through firewalls and NATs, fortunately the problem 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]. Consequently 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 heterogeneous 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 agent communication problem in a heterogeneous network such as data Grids. In our proposed architecture, we use five different types of agents, each having its own characteristics as the followings:

a) Manager Agent (MA): MA has the responsibility of managing the whole system including other agents 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.

b) 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.

c) 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 responsible for its local LAN ARM agents.

---

[2] Remote Method Invocation

d) 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.

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

The proposed multi-agent architecture is shown in Fig. 19.1. As the most innovative design parts were done on ARMA, IA, and RA, we focus on their details in the following sections

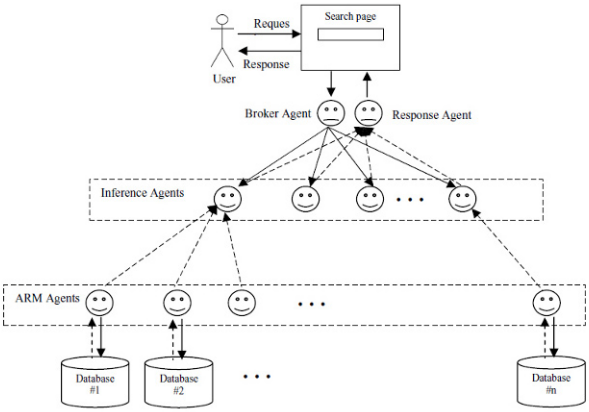

Fig. 19.1: The proposed multi-agent knowledge mining architecture

19.3.1 ARM Agent Behavior

Having created each ARMA, they would start mining according to their predefined behavior. 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. Although mostly these parameters are set to 50% , in our thesis we used a game theory-based mechanism to define almost optimal *MinSup* and *MinCon* values for them.

19.3.1.1 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. 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 apriori 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].

```
L_1={Large 1-itemsets};
for (k=2; L_k-1≠∅; k++) do begin
  C_k=apriori-gen(L_k-1); // new candidates
  forall transactions t∈D do begin
    C_t=subset(C_k,t);   // candidates contained in t
    forall candidates c∈C_t do
      c.count++;
  end
  L_k={c∈C_k|c.count ≥ minsup}
end
Answer = ∪_k L_k;
```

Fig. 19.2: Apriori Algorithm

19.3.1.2 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. 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 S_i 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 *Nash equilibrium* in an n-player game is a set of strategies, $\sigma = \{\sigma^1, ..., \sigma^n\}$, 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. [22] 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 common classic example for equilibrium point is *two prisoners dilemma* (PD) in which there are two prisoners kept in separated cells. If both confess they would be sentenced three years in jail. If one confess and the other one dont (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 for a less important crime and will go one year in the jail each. The matrix in Fig. 19.3 shows that the best action a player can do is not to confess and the Nash equilibrium is (dont confess, dont confess).

	confess	Do not confess
Confess	3 - 3	0 – 4
Do not confess	4 - 0	1 – 1

Fig. 19.3: The 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 lets assume in the real world a computer multi-conference is going 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. Intuitively the game equilibrium would be the case in which 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 in our work is somehow the same for ARM agents which are in charge of gathering first level knowledge. They will not get any 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. Here we propose a lemma explaining the competitive knowledge presentation.

Lemma 19.1. *Competitive Knowledge Presentation: There exist equilibrium for knowledge presentation game in which players would select their knowledge with the highest possible confidence and support even if is less than common suggested 50% MinCon and MinSup.*

Proof. Let A be the payoff for mutual knowledge presentation, B the payoff for the presenter and D for not-presenter in case one presents and the other does not, and C the payoff when both do not present. In case A, both players receive a reward of 50% which is chance of presenting their work. In case B, the one who presented the knowledge would get 100% and the other player receives the payoff 0%; and in case C both players receive payoff 0%. Obviously if you think the other player will not present then you should present to give a payoff of 100% and in case he other guy also presents, you must present to have a 50% chance to win the game. No matter what option your opponent chooses, you should present your own work. According to equilibrium condition in PD, certain following conditions have to hold: $B \geq A \geq C \geq D$. Also an even chance of being exploited or doing the exploiting is not as good an outcome as both players mutually presenting. Therefore, the reward for A should be greater than the average of the payoff for the B and the D. That is, the following must hold $A \geq (D+B)/2$. And we see that our model yields these equations, therefore it has equilibrium, the same as PD. □

This mechanism assures that all ARMAs do their best to propose something as a first level knowledge. The ARMA game matrix in Fig. 4 shows agents payoff in winning percentage regarding their knowledge presentation.

	Present knowledge	Do not present knowledge
Present knowledge	50% - 50%	100% - 0%
Do not present knowledge	0% - 100%	0% - 0%

Fig. 19.4: proposed ARMA game matrix

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 disease with respect to conditions such as country, race, age and etc.

19.3.2 Inference Agent Behavior

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.

19.3.3 Response Agent Behavior

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.

Dempster-Shafer theory of evidence: The Dempster-Shafer (DS) theory was first introduced by Dempster (1968) [23] and then extended by Shafer (1976) [24], 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 [25], [26], [27].

This theory is a generalization 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 [25]. In information retrieval the uncertainty occurs 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
- Existence 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 $m : \rho(U) \rightarrow [0,1]$, called *Basic Probability Assignment* (BPA), where m(A) represents *Partial Belief* amount of A. Note that $m(\phi)=0$ and $\sum_{A \in U} m(A) = 1$.

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

$$Bel(A) = \sum_{\forall B \subset A} m(B) \tag{19.1}$$

Shafer defined *doubt* amount in A as the belief in A', and the *plausibilty function* as the total belief amount in A

$$Dou(A)=Bel(A') \tag{19.2}$$

$$Pl(A)=1-Dou(A) = \sum_{B \subseteq U} m(B) - \sum_{B \subseteq A'} m(B) = \sum_{B \cap A = \phi} m(B) \tag{19.3}$$

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)*]. Dempster's rule of combination is a generalization of Bayes' rule. This rule strongly emphasizes 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:

$$m(A) = m_1 \otimes m_2 = \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{\sum_{B\cap C\neq\phi} m_1(B)m_2(C)} \quad (19.4)$$

19.4 System Run Sample

In order to show the knowledge mining process in our system, we used a self-generated patient database, according to a disease symptoms website[3] . Since using diseases with completely separated symptoms are not desirable as a test case, we chosen those with common signs which are: *Cold*, *Flu*, *Bronchitis*, *Allergy*, *Asthma*, *Sinusitis*, *Strepthroat*, and *Gastroenteritis*. Some of the extracted association rules done by ARMAs are as bellow:

$\{runnynose\} \longrightarrow \{cold\} sup : 0.58, conf : 1.0$
$\{sorethroat\} \longrightarrow \{cold\} sup : 0.55, conf : 1.0$
$\{headache\} \longrightarrow \{cold\} sup : 0.62, conf : 1.0$
$\{cough\} \longrightarrow \{cold\} sup : 0.72, conf : 1.0$
$\{sorethroat fever\} \longrightarrow \{cold\} sup : 0.34, conf : 1.0$
...
$\{chills\} \longrightarrow \{flu\} sup : 0.62, conf : 1.0$
$\{runnynose\} \longrightarrow \{flu\} sup : 0.58, conf : 1.0$
$\{fever\} \longrightarrow \{flu\} sup : 0.51, conf : 1.0$
$\{chillsmuscleache\} \longrightarrow \{flu\} sup : 0.24, conf : 1.0$
$\{muscleachecough\} \longrightarrow \{flu\} sup : 0.34, conf : 1.0$
...
$\{itchy\} \longrightarrow \{allergy\} sup : 0.62, conf : 1.0$
$\{sneeze\} \longrightarrow \{allergy\} sup : 0.79, conf : 1.0$
$\{itchyrunnynose\} \longrightarrow \{allergy\} sup : 0.24, conf : 1.0$
$\{sneezeitchy\} \longrightarrow \{allergy\} sup : 0.44, conf : 1.0$
...
$\{wheezing\} \longrightarrow \{bronchitis\} sup : 0.72, conf : 1.0$
$\{breathshortness\} \longrightarrow \{bronchitis\} sup : 0.62, conf : 1.0$
$\{feverwheezing\} \longrightarrow \{bronchitis\} sup : 0.41, conf : 1.0$
...
$\{eyepain\} \longrightarrow \{sinusitis\} sup : 0.51, conf : 1.0$
$\{eyepain\} \longrightarrow \{sinusitis\} sup : 0.51, conf : 1.0$
$\{cough\} \longrightarrow \{sinusitis\} sup : 0.62, conf : 1.0$
$\{coughheadache\} \longrightarrow \{sinusitis\} sup : 0.44, conf : 1.0$
...

[3] http://familydoctor.org/

$\{headache\} \longrightarrow \{strepthroat\} sup : 0.79, conf : 1.0$
$\{sorethroat\} \longrightarrow \{strepthroat\} sup : 0.58, conf : 1.0$
$\{fever\} \longrightarrow \{strepthroat\} sup : 0.72, conf : 1.0$
$\{feverheadache\} \longrightarrow \{strepthroat\} sup : 0.55, conf : 1.0$
...
$\{breathshortness\} \longrightarrow \{asthma\} sup : 0.58, conf : 1.0$
$\{wheeze\} \longrightarrow \{asthma\} sup : 0.79, conf : 1.0$
$\{cough\} \longrightarrow \{asthma\} sup : 0.62, conf : 1.0$
$\{wheezecough\} \longrightarrow \{asthma\} sup : 0.44, conf : 1.0$
$\{wheezebreathshortness\} \longrightarrow \{asthma\} sup : 0.44, conf : 1.0$
...

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 itchy has X sneeze THEN has X allergy(S:0.44,C:0.72)
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)
...

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 shown below:

IA#1:
has patient gastroenteritis(s:0.51,c:1.0)
has patient strepthroat(s:0.72,c:1.0)
has patient strepthroat(s:0.55,c:1.0)
has patient cold(s:0.20,c:1.0)

IA#2:
has patient bronchitis(S:0.51,C:1.0)
has patient flu(S:0.51,C:1.0)
has patient sinusitis(S:0.79,C:1.0)
has patient gastroenteritis(s:0.62,c:1.0)
has patient gastroenteritis(s:0.20,c:1.0)
has patient strepthroat(s:0.79,c:1.0)
has patient cold(s:0.62,c:1.0)
has patient cold(s:0.51,c:1.0)

Finally and in the last phase, RA implies D-S combination method, shown in Table 19.1, and returns the final result to the user. After computing the D-S combination table, the final probable amount would be: m(Gastroenteritis)=0.001318070061 , m(Cold)= 0.001318070061, m(Strepthroat)= 0.001314439014, m(Bronchitis)= 0, m(Flu)= 0, and m(Sinusitis)= 0. Therefore the most probable diagnosis through this process would be *Gastroenteritis*, *Cold*, *Strepthroat*,*Bronchitis*, *Flu*, and *Sinusitis* respectively.

Table 19.1: D-S combination table

	Bronchitis: 0.0181	Flu: 0.0181	Sinusitis: 0.0181	Gastroenterit: 0.0363	Strepthroat: 0.0181	Cold: 0.0363
Bronchitis: 0	0	0	0	0	0	0
Flu: 0	0	0	0	0	0	0
Sinusitis: 0	0	0	0	0	0	0
Gastroenterit: 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

19.5 Conclusion and Future Work

In this work, 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 diseases 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 grid-based search engine

Acknowledgements Author would like to thank contribution of Prof. Mahmoud Naghibzadeh, Hossein Kamyar, and Saied Moghaddaszadeh from Ferdowsi University of Mashhad, Iran, for their modest comments and help in part of system implementation.

References

1. Frawley, W., Piatetsky-Shapiro G., Matheus C.: Knowledge Discovery in Databases: An Overview, pp: 213-228. AI Magazine (1992)
2. Milani Fard A., Kamyar H., Naghibzadeh M. :Multi-expert Disease Diagnosis System over Symptom Data Grids on the Internet: World Applied Sciences Journal, ISSN 1818-4952, Volume 3 Number 2, (2008)
3. Shortliffe, E. H.: MYCIN: A rule-based computer program for advising physicians regarding antimicrobial therapy selection, Ph.D. dissertation, Stanford University (1974)
4. Russell, S.J., Norvig, P.: Artificial Intelligence: A modern Approach, Prentice Hall (1995)

5. Waterhouse, S.R.: Classification and regression using mixtures of experts, Ph.D. dissertation, Department of Engineering, Cambridge University (1997)
6. Si, L., Callan, J.: A Semisupervised Learning Method to Merge Search Engine Results. ACM Transactions on Information Systems, Vol. 21, No. 4, 457-491, (2003)
7. Callan, J.: Distributed information retrieval. In: Advances in Information Retrieval, W. B. Croft, Ed., pp. 127150, Kluwer Academic Publishers (2000)
8. Pasi, G., Yager, R. R.: Document retrieval from multiple sources of information. In: Uncertainty in Intelligent and Information Systems, Bouchon-Meunier, B., Yager, R. R., Zadeh, L. A. (eds.), pp. 250-261, World Scientific, Singapore (2000)
9. Baeza-Yates, R., Ribeiro-Neto B.: Modern Information Retrieval. Addison-Wesley, (1999)
10. Bellifemine, F., Caire, G., Trucco, T., Rimassa, G.: JADE Programmers Guide. (2006)
11. JADE Board: JADE WSIG Add-On Guide JADE Web Services Integration Gateway (WSIG) Guide. 03 March, (2005)
12. Cortese, E., Quarta, F., Vitaglione, G., Vrba, P.: Scalability and Performance of the JADE Message Transport System. (2002)
13. Liu, S., Kngas, P., Matskin, M.: Agent-Based Web Service Composition with JADE and JXTA. In: SWWS 2006, Las Vegas, USA, June 26-29, (2006)
14. Gradecki, J. D.: Mastering JXTA: Building Java Peer-to-Peer Applications JohnWiley (2002)
15. Milani Fard, A., Kahani, M., Ghaemi, R., Tabatabaee, H.: Multi-agent Data Fusion Architecture for Intelligent Web Information Retrieval. In: International Journal of Intelligent Technology Volume 2 Number 3 (2007)
16. Mohebbi, M., Akbarzadeh T., M. R. , Milnai Fard, A.: Microorganism DNA PatternSearch in a Multi-agent Genomic Engine Framework. In World Applied Science Journal, Volume 2 Number 5, Sep-Oct (2007)
17. Tabatabaee, H., Milani Fard, A., Akbarzadeh T., M. R.: Cooperative Criminal Face Recognition in Distributed Web Environment. In: The 6th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-08), Doha, Qatar, March 31 - April 4 (2008)
18. Agrawal, R., Imielinski, T. , Swami, A. N.: Mining Association Rules between Sets of Items in Large Databases. In: SIGMOD, 22(2):207-16, June (1993)
19. Kantardzic, M.: Data Mining: Concepts, Methods, and Algorithms. JohnWiley (2003)
20. Von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press (1944)
21. Vidal, J. M.: Learning in Multiagent Systems: An Introduction from a Game-Theoretic Perspective. In: Alonso, E. (eds.) Adaptive Agents. LNAI 2636. Springer Verlag (2003)
22. Mor, Y., Goldman, C. V., Rosenschein, J. S.: Learn your Opponent's strategy (in Polynomial Time)!. In: Adaptation and Learning in Multi-Agent Systems, IJCAI95 Workshop, Montreal, Canada, August 1995, In: Proceedings. Lecture Notes in Artificial Intelligence Vol. 1042, G. Weiss and S. Sen (eds.) Springer Verlag, (1996)
23. Dempster, A.: A generalization ofba yesian inference. In: Journal of the Royal Statistical Society, 30:205-247 (1968)
24. Schafer, G.: A mathematical theory of evidence., Princetown University Press (1976)
25. Verikas, A., Lipnickas, A. ,Malmqvist, K., Bacauskiene M., Gelzinis, A.: Soft combination of neural classifiers: A comparative study In: Pattern Recognition Letters, 20:429–444 (1999)
26. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation and active learning. In: G. Tesauro, D.S. Touretzky, T. K., Leen, (eds.) Advances in Neural Information Processing Systems, volume 7, pp. 231-238. MIT Press, Cambridge, MA (1995)
27. Kuncheva, L. I.: Combining Pattern Classifiers: Methods and Algorithms. John Wiley (2004)