

# Integrating Knowledge Through Cooperative Negotiation – A Case Study in Bioinformatics

Cassia T. dos Santos and Ana L.C. Bazzan

Instituto de Informática, Universidade Federal do Rio Grande do Sul,  
Caixa Postal 15064, 90.501-970, Porto Alegre, RS, Brazil  
{ctsantos, bazzan}@inf.ufrgs.br

**Abstract.** Data Mining techniques have been used for knowledge extraction from large volumes of data. A recent practice is to combine Data Mining and Multi-Agent Systems approaches. In this paper we propose the use of cooperative negotiation to construct an integrated and coherent domain model from several sources. Agents encapsulate different symbolic machine learning algorithms to induce their individual models. After this, a global model yields from the interaction via cooperative negotiation of these agents. The results shows that the proposed approach improves the accuracy of the individual models, integrating the best representations of each one.

## 1 Introduction

Knowledge discovery and data mining are concerned with the use of machine learning techniques for knowledge extraction from large volumes of data. A recent practice is to combine data mining and multi-agent systems approaches [16, 18]. In these systems, a group of agents are responsible for applying different machine learning algorithms and/or using subsets of the data to be mined, and are able to cooperate to discover knowledge from these subsets. This approach has shown high potential for many applications and it is opening interesting research questions. An important one is how to integrate the discovered knowledge by agents into a globally coherent model.

In this paper we propose the use of cooperative negotiation to construct an integrated domain model from several sources. In our approach, agents encapsulate different symbolic machine learning algorithms to induce their individual models. After this, a global model results from the interaction of these agents via cooperative negotiation.

To demonstrate the application of this approach, we use a scenario from bioinformatics, which is related to automated annotation of proteins' keywords. However, any domain can be used, provided the problem can be stated as a set of attribute-value pairs, as it is the standard practice in machine learning. This is so because the agents do not use any domain dependent information as they just encapsulate data and machine learning algorithms.

In the case of bioinformatics, the machine learning algorithms are used to induce models to predict the annotation, using data from biological databases.

Since these databases are very large, both in number of records and in number of attributes, using all available data for symbolic machine learning algorithms is prohibitive. Our approach can be used to solve this problem, allowing several agents to cooperate to induce a global classifier, from subsets of the data to be mined.

This paper is structured as follows. The next section briefly reviews some issues in cooperative negotiation. Section 3 introduces our approach to integrate knowledge of learning agents, while Section 4 presents its use in the automated annotation of proteins. Section 5 describes some related work regarding multi-agent systems and data mining. Section 6 presents the final remarks and the future works.

## 2 Cooperative Negotiation

Negotiation is a process in which two or more parties make a joint decision. It is a key form of interaction that enables groups of agents to arrive to mutual agreement regarding some belief, goal or plan [2]. Hence the basic idea behind negotiation is reaching a consensus [9].

According to [12], a negotiation model is composed by two basic components: the negotiation protocol and the negotiation strategies. The protocol specifies the rule of encounter between the negotiation participants. That is, it defines the circumstances under which the interaction between agents take place, what deals can be made and what sequences of actions are allowed [6]. An strategy is the specification of the sequence of actions the agent plans to make during the negotiation.

Negotiation usually proceeds in a series of rounds, with every agent making a proposal at each round. According to [11], the process can be described as follow. One agent generates a proposal and other agents review it. If some other agent does not like the proposal, it rejects and may generate a counter-proposal. If so, the others agents (including the agent that generated the first proposal) then review the counter-proposal and the process repeats. It assumes that a proposal becomes a solution when it is accepted by all agents.

Cooperative negotiation is a particular kind of negotiation where agents cooperate and collaborate to achieve a common objective, in the best interest of the system as a whole [2, 9]. In cooperative negotiation, each agent has a partial view of the problem and the results are put together via negotiation trying to solve the conflicts posed by having only partial views [8]. According to [10], cooperative negotiations can be described as the decision-making process of resolving a conflict involving two or more parties over multiple independent, but non-mutually exclusive goals.

This kind of negotiation has been currently adopted in resource and task allocation fields [3, 13, 20]. In these approaches, the agents try to reach the maximum global utility that takes into account the worth of all their activities. In our case, we view cooperative negotiation as a way to achieve knowledge integration, in order to remove redundancy or conflict between the partial views.

### 3 Approach

In our distributed learning system, the agents encapsulate different symbolic machine learning algorithms and use subsets of the data to be mined. Individual models are integrated using cooperative negotiation. This approach has the following advantages. First, each agent is responsible for a subset of considerably smaller size and dimensionality, improving the accuracy and performance of the algorithms. Second, the cooperative negotiation allows us to deal with redundancy and conflicts. Finally, no algorithm can be the best choice in all possible domains. Each algorithm contains an explicit or implicit bias that leads it to prefer certain generalizations over others: the strength of one can be the other's weakness [5]. Therefore, different machine learning techniques applied to the same data set hardly generate the same results [18]. An algorithm A can construct an accurate model for concept X and a weak description for concept Y, while the algorithm B constructs an accurate model for concept Y and a weak model for concept X. The combination of different learning algorithms can lead to more accurate models.

Basically, the process involves the following phases: (1) preparation of the data, (2) generation and evaluation of the individual models, and (3) construction of the integrated model through cooperative negotiation.

In the phase (1) the data to be mined is split into training, validation, and test subsets. The training subset is divided equally among the agents in the systems, creating views of it. The validation and test subsets are not divided. The former is used to evaluate the individual models (all agents use the same subset). The latter is reserved for the final evaluation of the integrated model.

In the phase (2), the agents work in an independent manner, and as a result they produce their individual models. Each agent constructs its own model based on its own subset of data, using one symbolic machine learning algorithm (here we use C4.5 [15] or CN2 [4]). The individual models are evaluated using the validation subset and each rule accuracy is estimated applying the Laplace expected error estimate (Equation 1). The formula depends on TP (number of true positives which means the number of examples correctly covered by the rule), FP (the false positives or the number of examples wrongly covered by the rule), and K (the number of classes in the domain).

$$\text{LaplaceAccuracy} = (TP + 1) / (TP + FP + K) \quad (1)$$

Phase (3) is dedicated to constructing an integrated model based on the results obtained in phase (2). Considering that each learning algorithm uses a proper syntax to describe the induced model, it is necessary to have the models represented in the same format. The PBM [14] format is adopted, which generates sets of propositional rules of the form: *if <condition> then <class=C>*. The representation process takes place after the agents apply their learning algorithms to induce its models, before the negotiation process starts.

The negotiation process involves two types of agents: learning agents and mediator agent. The learning agents encapsulate the machine learning algorithms

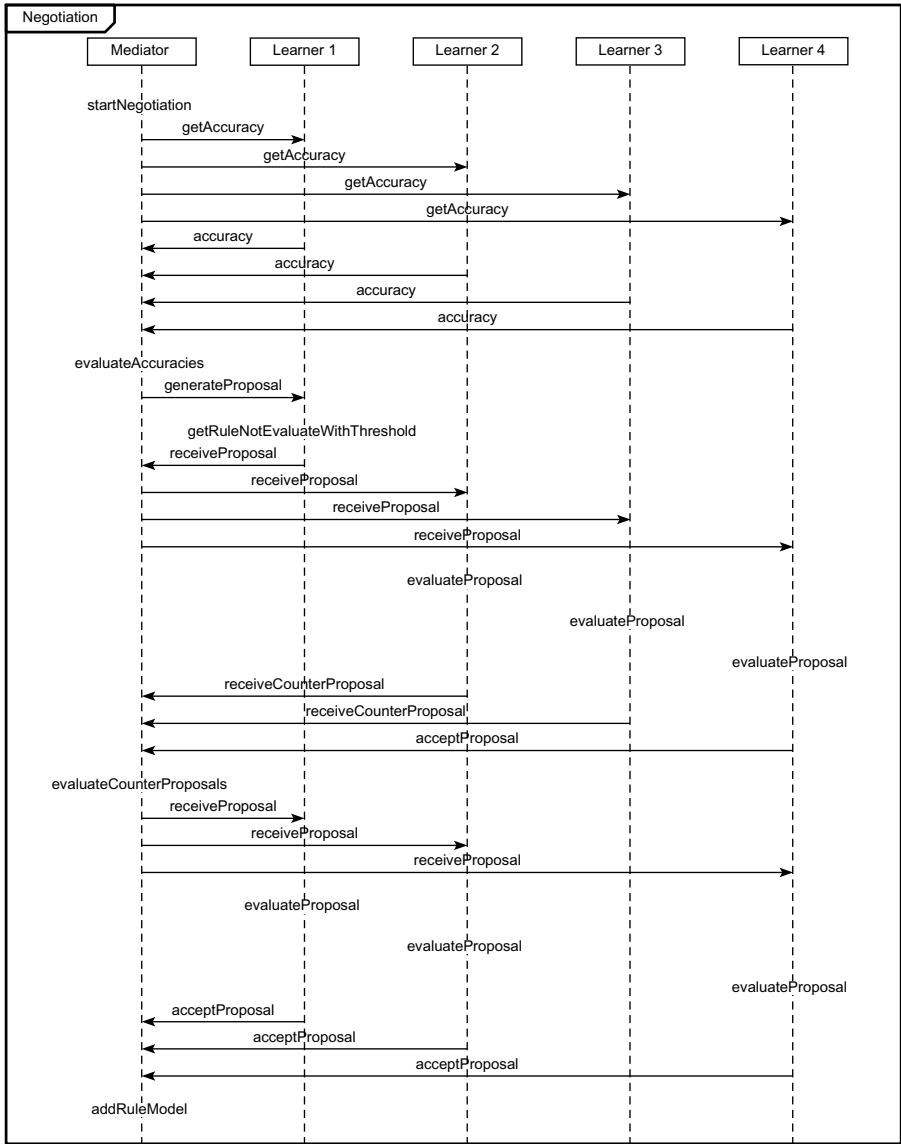


Fig. 1. AUML interaction diagram

and the mediator agent is responsible for controlling the communication among the learning agents and finalize the negotiation.

Figure 1 shows an AUML interaction diagram with the messages exchanged between the agents during a negotiation round. We use an extension of AUML-2 standard to represent agents' actions (e.g. startNegotiation, getAccuracy, etc.).

The negotiation process starts with the mediator agent asking the learning agents to send their overall accuracies. The first learning agent to generate a proposal is the one that has the poorest overall accuracy (learner 1, in the specific example). The proposal contains the first rule in the rule set of this agent, which has not yet been evaluated, and which has an accuracy equal or better than a threshold. Rules which do not satisfy the threshold are not considered. This proposal is then sent to the mediator agent, which repasses it to others agents. Each agent then evaluates the proposal, searching for a equivalent rule with better accuracy.

This evaluation requires comparison of rules and, therefore, it is necessary to establish equivalences for them. One rule is equivalent to another one when both describe the same concept and at least one attribute overlaps. If an equivalent rule is not found, or an equivalent one does not have better accuracy, then the agent accepts the proposal (in the example in Figure 1, learner 4 has accepted the first proposal). Otherwise, if the agent has a rule with a better accuracy than the proposed rule, then its rule is sent as a counter-proposal to the mediator agent, which evaluates the several counter-proposals received. This is so because several agents can send a counter-proposal. In our example, learners 2 and 3 have both generated counter-proposals. The one which has the best accuracy is selected. Here, this is the case for learner 3's rule. The selected counter-proposal is then sent to the other agents (including the agent that generated the first proposal). These review it, and the process is repeated. When a proposal or a counter-proposal is accepted by all agents, the mediator adds the corresponding rule in the integrated model and the learning agents mark its equivalent one as evaluated. The negotiation ends when all rules were evaluated.

## 4 Case Study in Automated Annotation of Proteins

We apply the proposed approach to induce a model for annotation of proteins, regarding specifically the field called *Keywords* in Swiss-Prot<sup>1</sup> [7]. Swiss-Prot is a curated database which provides a high level of annotation for each protein, including a description of its function, its domain structure, etc. It has also extensive links to others databases. The *Keyword* field gives several hints to experts as to what regards proteins functions and structure.

### 4.1 Methods

We used proteins from the model organism *Arabidopsis thaliana*, which are available in public databases such as Swiss-Prot. The data used comes from a local version of the Swiss-Prot database downloaded in November of 2004. Using it, 3038 proteins were found which relates to *A. thaliana*. 347 keywords appeared in the data but we focus here on those whose number of instances is higher than 100. The number of keywords satisfying this criterion is 29.

---

<sup>1</sup> <http://www.expasy.ch/spot>

The attributes which appeared in this data are only the *accession numbers* (a sort of protein family classification) for all attributes related to the Interpro<sup>2</sup> [1], which appear in Swiss-Prot as a cross-referenced database. The number of attributes of this domain is 1496.

For each keyword, the collected data was split in three subsets: training (80% – 2430 instances), validation (10% – 304 instances) and test (10% – 304 instances). The training set was split in four subsets, each one with approximately 607 examples. The data was split according to the appearance order in the database. For instance, the first 2430 instances were used in the training set. Since there is no ordering in the original database, the split is not biased.

Each subset of the training set was assigned to one of the learning agents A, B, C, and D. Besides, two agents encapsulate the C4.5 algorithm (agents A and C) and the other two (agents B and D) use the CN2 algorithm. These subsets were organized in input files, respecting the specific syntax of the C4.5 and CN2.

Using the input data, the four learning agents induce rules for each keyword (target class). Figure 2 (a) shows the output of the agent A (C4.5) for the keyword “Iron”. Due to lack of space, we do not show all induced rules for this keyword. The rule 3 in Figure 2 suggests the annotation of the keyword “Iron” for a given protein if it belongs to the IPR002226 family of proteins. According to the rule 4, the protein must be annotated with the “Iron” keyword if it belongs to the IPR001199. Similar boolean tests are performed in the rule 5, here suggesting that the keyword should not be annotated if it does not belong to the Interpro families IPR002680, IPR006186, IPR002226, and IPR001199.

Figure 2 (b) shows the output of the agent B (CN2) for the same keyword. The rules are read as follows. If the protein belongs to Interpro family IPR005708 (first rule) or IPR005956 (second rule), then it shall be annotated with the keyword “Iron”. According to the third rule, if the protein does not belong to the Interpro families IPR005123, IPR000181, IPR006057, IPR008331, IPR001260, IPR005708, IPR001015, and IPR005956 then “Iron” shall not be annotated.

Once the rules were generated, they were transformed into the PBM format. Figures 3 (a) and 3 (b) respectively show the rules presented in the Figures 2 (a) and 2 (b) in the PBM format.

After the rule transformation process, the rules of the four agents are evaluated using the validation subset. For each rule of the model, its accuracy was estimated by applying the Laplace expected error measure (Equation 1). Only rules with accuracy equal or better than a threshold are considered. The overall accuracy of the agent is obtained from the average accuracy of its rules that satisfied the threshold.

Once the evaluation of the individual models was made, the negotiation process starts. As said above, the negotiation process (Figure 1) starts with the mediator agent asking to the four learning agents for their overall accuracies. The first learning agent to generate a proposal is the one that has the poorest overall accuracy. This then selects the first rule in its rule set that satisfies the

<sup>2</sup> <http://www.ebi.ac.uk/interpro>

Rule 3:	IF IPR005708 = y
. IPR002226 = y	THEN class = Iron [1 0]
. -> class Iron [63.0%]	
Rule 4:	IF IPR005956 = y
. IPR001199 = y	THEN class = Iron [1 0]
. -> class Iron [50.0%]	
Rule 5:	IF IPR005123 = n
. IPR002680 = n	AND IPR000181 = n
. IPR006186 = n	AND IPR006057 = n
. IPR002226 = n	AND IPR008331 = n
. IPR001199 = n	AND IPR001260 = n
. -> class n [99.0%]	AND IPR005708 = n
	AND IPR001015 = n
	AND IPR005956 = n
	THEN class = n [0 590]
(a)	(b)

**Fig. 2.** (a) Output of the C4.5; (b) output of the CN2

R0003.	IF IPR002226 = y	R0007.	IF IPR005708 = y
.	THEN CLASS = Iron	.	THEN CLASS = Iron
R0004.	IF IPR001199 = y	R0008.	IF IPR005956 = y
.	THEN CLASS = Iron	.	THEN CLASS = Iron
R0005.	IF IPR002680 = n	R0009.	IF IPR005123 = n
.	AND IPR006186 = n	.	AND IPR000181 = n
.	AND IPR002226 = n	.	AND IPR006057 = n
.	AND IPR001199 = n	.	AND IPR008331 = n
.	THEN CLASS = n	.	AND IPR001260 = n
		.	AND IPR005708 = n
		.	AND IPR001015 = n
		.	AND IPR005956 = n
		.	THEN CLASS = n
(a)		(b)	

**Fig. 3.** (a) C4.5 rules in the PBM format; (b) CN2 rules in the PBM format

threshold and sends it to the mediator agent, which repasses the rule to the others learning agents. These agents evaluate the rule, searching for an equivalent one, with better accuracy. If an equivalent rule with better accuracy than the proposed rule is not found, the agent that is evaluating the proposal accepts the proposed rule. Otherwise, if it has an equivalent rule with better accuracy, it sends it as a counter-proposal to the mediator agent, which evaluates the several rules received and selects the one that has the best accuracy. The selected rule is then sent to other agents which review it.

A rule is added to the integrated model if it is accepted by all agents. This rule and its equivalent one are marked as evaluated by the corresponding agent during the evaluating rules process. The negotiation process is repeated until there are no more rules to evaluate. This process was done for each keyword (i.e. the agents negotiate integrated models for each keyword).

## 4.2 Results and Discussion

The integrated model generated through cooperative negotiation was evaluated using the test subset. For comparative purposes, the individual models were also evaluated using this subset. Table 1 shows the overall accuracies obtained for each individual model (agents A, B, C and D) and for the integrated model, for each keyword. In these experiments, a threshold equal to 0.5 was used and only rules with accuracy equal or superior than this threshold were considered in the negotiation process.

As shown in Table 1, the integrated model has a better overall accuracy than the individual models. This result was obtained for all keywords.

Regarding the individual learning agents, those that encapsulated the C4.5 algorithm obtained better results than those with the CN2 algorithm. For in-

**Table 1.** Overall accuracies obtained using validation subset

Keyword	Agent A	Agent B	Agent C	Agent D	Integrated model
	C4.5	CN2	C4.5	CN2	
Alternative-splicing	0.72	0.51	0.94	0.51	0.94
ATP-binding	0.55	0.52	0.54	0.53	0.96
Calcium	0.98	0.66	0.62	0.62	0.98
Cell-wall	0.88	0.55	0.88	0.55	0.88
Chloroplast	0.53	0.50	0.54	0.50	0.93
Coiled-coil	0.60	0.52	0.60	0.53	0.91
DNA-binding	0.53	0.51	0.52	0.51	0.73
Glycoprotein	0.55	0.52	0.61	0.54	0.96
Heme	0.64	0.51	0.69	0.62	0.74
Hydrolase	0.52	0.51	0.53	0.52	0.82
Iron	0.58	0.54	0.59	0.56	0.99
Membrane	0.55	0.59	0.61	0.52	0.65
Metal-binding	0.55	0.53	0.56	0.52	0.96
Mitochondrion	0.55	0.51	0.65	0.51	0.96
Nuclear-protein	0.51	0.50	0.52	0.50	0.70
Oxidoreductase	0.55	0.51	0.52	0.53	0.98
Phosphorylation	0.57	0.54	0.58	0.53	0.99
Plant-defense	0.66	0.58	0.59	0.58	0.98
Protein-transport	0.58	0.60	0.56	0.60	0.92
Repeat	0.53	0.51	0.53	0.51	0.89
Ribosomal-protein	0.66	0.99	0.56	0.50	0.99
Signal	0.52	0.51	0.53	0.52	0.81
Transcription-regulation	0.53	0.51	0.54	0.52	0.73
Transferase	0.53	0.51	0.52	0.51	0.85
Transit-peptide	0.53	0.50	0.53	0.50	0.94
Transmembrane	0.51	0.51	0.52	0.51	0.86
Transport	0.53	0.56	0.53	0.55	0.89
Zinc	0.56	0.54	0.65	0.53	0.99
Zinc-finger	0.61	0.55	0.96	0.54	0.96
Average	0.59	0.54	0.60	0.53	0.89



stance, considering the keyword “Iron”, agent B (CN2) has the poorest overall accuracy and generated the first proposal. The proposed rule has an accuracy equal to 0.58 and suggests the non annotation of the keyword “Iron”. This rule was then evaluated by the agents A (C4.5), C (C4.5), and D (CN2). Agents C and D do not find an equivalent rule with better accuracy and they accept the proposed rule. However, agent A found an equivalent one with accuracy equal to 0.99, thus it generated a counter-proposal. Then, agents B, C, and D evaluate the counter-proposal and accepted it. Finally, each agent mark the equivalent rules as evaluated. This way, in each round, the best rule that satisfied the threshold is added to the integrated model.

Also experiments with only two agents, each using all training data, were performed. In this case, each agent encapsulates one of the machine learning algorithms – C4.5 and CN2. Both used all 2430 instances of the training set to induce its model. The validation was performed using the instances of the validation set. The C4.5 agent obtained an overall accuracy (for all keywords) equal to 0.52 and the CN2 agent produced an accuracy of 0.50. For both agents, the quality of rules was poor when compared to the results obtained by the integrated model (0.89 – overall accuracy for all keywords). In one specific case, considering only the keyword “Iron”, the C4.5 agent had an accuracy equal to 0.51 and CN2 agent presented an accuracy of 0.53, while the integrated model obtained an accuracy of 0.99. This happens because the amount of data is very high, thus the algorithms do not induce good models. Also, this task is time-consuming.

These results show that the proposed approach can be applied to improve the accuracy of the individual models, integrating the better representations of each one.

## 5 Related Work

A multi-agent system for mining distributed data is presented in [17]. There are two types of agents: the learner and the meta-learner. The learner has a machine learning algorithm, and each learner applies its technique separately and brings the results to be combined by the meta-learner.

The CILT system [18, 19] is based on agents with different machine learning algorithms that collaborate with each other to improve the classification task. Due to this collaboration, the agents generate new data and add it to the training file, which is further presented to the agents for the sake of classification improvement. In [16] an architecture for an environment is proposed, which combines different machine learning algorithms encapsulated in agents that collaborate to improve their knowledge. The learning process happens in two stages: individual and cooperative. The objective of the first stage is to create an individual domain model, based on different classifiers. In the cooperative learning stage, agents cooperate to improve their knowledge by sharing it with others to achieve better results. The data is not divided among the agents, i.e. all agents use the same dataset.

In general, the above mentioned approaches are based on selecting the best rules to compose a global domain model. The main contribution of the approach proposed here is a model of cooperative negotiation to solve conflicts and redundancies among the several models, hold by different learning agents having different views of the data set.

## 6 Final Remarks and Future Work

The application of a multi-agent system for improving symbolic learning through knowledge integration has shown high potential for many applications and it is opening important and interesting research questions. The possibility of combining different machine learning algorithms can lead to better results and using subsets of considerably smaller size and dimensionality improves the accuracy and performance of these algorithms.

This paper describes an approach based on cooperative negotiation to construct an integrated domain model from several sources. Several agents encapsule different symbolic machine learning algorithms to induce their individual models and the global model results from the cooperative negotiation of these agents.

The main achievement of this approach is to integrate knowledge from models which are generated by different machine learning agents, using different views of the training data, thus improving the accuracy of the individual models.

We have described an application in bioinformatics, namely the automated annotation of proteins, with data obtained from the Swiss-Prot database regarding the model organism *A. thaliana*. In our initial experiments, we obtained an integrated model which presents good classification rates on the validation data. The best representations of each learning agent were added to the integrated model, improving the performance of the annotation.

Cooperative negotiation to solve the conflicts posed by the partial views is still little explored in the process of knowledge integration. This work is a step towards an effective use of agents in knowledge integration and data mining. The negotiation process presented here is a basic one, which depends on a mediator. Further improvements can be done towards agents anticipating other proposals and counter-proposals, as well as learning from past deals among them. Possibly, this will require domain knowledge so that there will be a trade-off between better negotiation protocols and domain-independence. This issue will be investigated in the near future.

## Acknowledgement

Authors partially supported by CNPq; the project is partially supported by CNPq and by FAPERGS.

## References

1. Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D. R., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Hermjakob, H., Hulo, N., Jonassen, I., Kahn, D., Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N. J., Oinn, T. M., Pagni, M., Servant, F., Sigrist, C. J. A., Zdobnov, E. M.: The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Research* **29**(1) (2001) 37–40
2. Beer, M., d'Inverno, M., Luck, M., Jennings, N., Preist, C., Schroeder, M.: Negotiation in multi-agent systems. In *Workshop of the UK Special Interest Group on Multi-Agent Systems* (1998)
3. Bigham, J., Du, L.: Cooperative negotiation in a multi-agent system for real-time load balancing of a mobile cellular network. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM Press (2003) 568–575
4. Clark, P., Niblett, T.: The CN2 induction algorithm. *Machine Learning* **3** (1989) 261–283
5. Dietterich, T.: Limitations on inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning* (1989) 124–128
6. Fatima, S., Wooldridge, M., Jennings, N.: An agenda-based framework for multi-issue negotiation. *Artificial Intelligence* **152** (2004) 1–45
7. Gasteiger, E., Jung, E., Bairoch, A.: Swiss-prot: Connecting biological knowledge via a protein database. *Curr. Issues Mol. Biol.* **3** (2001) 47–55
8. Gatti, N., Amigoni, F.: A cooperative negotiation protocol for physiological model combination. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems* (2004) 655–662
9. Green, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F., Evans, R.: Software agents: A review. Technical report, Trinity College (1997)
10. Guttman, R. H., Maes, P.: Cooperative vs. competitive multi-agent negotiations in retail electronic commerce. *Lecture Notes in Computer Science* **1435** (1998) 135–147
11. Lander, S., Lesser, V.: Understanding the role of negotiation in distributed search among heterogeneous agents. In *Proceedings of the International Joint Conference on Artificial Intelligence* (January 1993)
12. Lomuscio, A. R., Wooldridge, M., Jennings, N.: A classification scheme for negotiation in electronic commerce. *Agent-Mediated Electronic Commerce: A European Agent Link Perspective* (2001) 19–33
13. Mailler, M., Lesser, V., Horling, B.: Cooperative negotiation for soft real-time distributed resource allocation. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*. ACM Press (2003) 576–583
14. Prati, R. C., Baranauskas, J. A., Monard, M. C.: Uma proposta de unificação da linguagem de representao de conceitos de algoritmos de aprendizado de máquina simbólicos. Technical Report 137, ICMC-USP (2001)
15. Quinlan, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann (1993)
16. Schroeder, L. F., Bazzan, A. L. C.: A multi-agent system to facilitate knowledge discovery: an application to bioinformatics. In *Proceedings of the Workshop on Bioinformatics and Multi-Agent Systems*, Bologna, Italy (2002) 44–50

17. Stolfo, S., Prodromidis, A., Tselepis, S., Lee, W., Fan, D.: Jam: Java agents for meta-learning over distributed databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*. AAAI Press (1997)
18. Viktor, H., Arndt, H.: Combining data mining and human expertise for making decisions, sense and policies. *Journal of Systems and Information Technology* **4**(2) (2000) 33–56
19. Viktor, H., Arndt, H.: Data mining in practice: From data to knowledge using hybrid mining approach. *The International Journal of Computers, Systems and Signal*, **1**(2) (2000) 139–153
20. Zhang, X., Lesser, V., Podorozhny, R.: Multi-dimensional, multistep negotiation for task allocation in a cooperative system. *Autonomous Agents and Multi-Agent Systems* **10** (2005) 5–40