A Multi-agent System to Facilitate Knowledge Discovery: an application to Bioinformatics

Luciana Fernandes Schroeder, Ana L. C. Bazzan

Instituto de Informática

Universidade Federal do Rio Grande do Sul

Av. Bento Gonçalves, 9500 – CP. 15064

91.501-970 Porto Alegre, RS, Brazil

{luciana,bazzan}@inf.ufrgs.br

ABSTRACT

Very few works exist on Multi-Agent systems aiming to improve symbolic learning through knowledge exchange. The motivation of this work is to mimic human beings interaction in order to reach better solutions. This aims at supporting a recent practice in Data Mining which is the use of collaborative systems. These systems can be based on agents which interact with each other and with the environment, cooperating to solve a problem. This article proposes an architecture for such an environment which combines different symbolic Machine Learning algorithms encapsulated in agents that collaborate to improve their knowledge. We use this environment to acquire rules for annotation of the field "Keywords" in the SWISS-PROT database.

Keywords

Multi-Agent Systems, Data Mining, Bioinformatics, Machine Learning.

1 INTRODUCTION

The motivation of this work is the application of a multi-agent system for improving symbolic learning through knowledge sharing. A recent practice in Data Mining is the use of collaborative multi-agent systems. These systems are usually based on agents which interact with each other and with the environment, cooperating to solve a problem.

Data Mining is a powerful technique that makes use of Machine Learning algorithms for knowledge extraction. However**,** 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 [11]: the strong point of one can be the other's weakness. Therefore, different Machine Learning techniques applied to the same dataset hardly generate the same result [20]. For example, figure 1 shows the result of two different Machine Learning algorithms (algorithm A and B) applied to the same dataset with two distinct concepts, x and y. The A tool constructed an accurate model for concept x and a weak description for concept y. On the other hand, the B tool builds a precise model for concept y and failed in the concept x description. In general, the combination of inductors increases the accuracy by reducing the bias. This integration aims at overcoming limitations of individual techniques through hybridization or fusion of various techniques. These ideas have lead to the emergence of many different kinds of system architectures.

Our aim is to evaluate the possibility of improving the classification with the MASKS (**M**ulti-**A**gent **S**ystem based on **K**nowledge **S**haring) environment, which combines inductors in a multi-agent system with autonomy to improve individual models through knowledge sharing. Besides, we describe the environment architecture and compare the first results driven from its use.

Following previous works on automated annotation using symbolic machine learning techniques, we applied a preliminary version of MASKS environment to produce rules for automatic annotation of the SWISS-PROT [2] Keywords field.

The paper is organized as follows. Section 2 describes some related work regarding Data Mining on biological data and Data Mining using multi-agent systems. Section 3 describes the MASKS environment architecture. Section 4 illustrates the application of this environment using a portion of SWISS-PROT data. Section 5 discusses future work, and Section 6 concludes the paper.

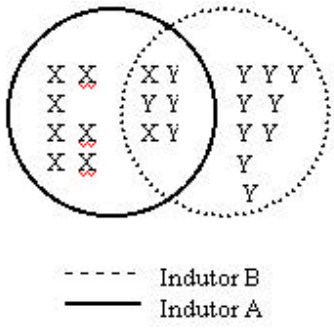

Figure 1 - Learning bias of Inductor A and B

2 RELATED WORK

The application of Data Mining techniques in biological databases is one of the most exciting activities in modern biology because there is much unexplored knowledge in the data. There are many works describing fair good results obtained from its use in molecular biology domain. In [9], two clustering techniques (k-means algorithm and hierarchic grouping) are matched to assemble genes with similar phenotypes; in [18], the Open Reading Frames (ORF or DNA sequences which contain nucleotides marking the beginning of a gene but not its end) of a yeast are classified in diverse functional categories; in [8], a Bayesian system is used to anticipate the place where the proteins became active in the cell; in [5], a decision network is constructed to classify the proteins in soluble or not soluble; in [14], the C4.5 algorithm is used to generate rules for automatic annotation of Keywords regarding proteins; and in [4], an alternative usage of the previous method [14] is discussed which reached 60% of automated annotation correctness of keywords for proteins related to the *Mycoplasmataceae* family.

A recent practice in Data Mining is the use of collaborative multi-agent systems. Some examples are given in [19, 21]. JAM [19] is a multi-agent system for mining distributed data. There are two agent types - the learner and the meta-learner. The learner has a Machine Learning algorithm - each learner applies its technique separately and brings the result to be combined by the meta-learner. The CILT system [21] 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.

As for the use of Multi-agent systems, in [7] a prototype is described aiming at automating the annotation of a virus sequence. This work is based on Multi-agent information gathering: search, filtering, integration, analysis, and presentation of the data to the user.

3 THE AGENT-BASED COOPERATIVE LEARNING

The MASKS environment groups different symbolic inductors encapsulated in agents in order to classify data. The model is build over a statistical analysis of a number of instances that describe the predetermined categories or concepts. If the model is considered acceptable, then it is used to classify future instances.

The MASKS environment defines the methods of interaction between the participants and the rules exchanged between the agents. The environment contains the learning problem to be solved by its members.

The environment goal is to improve the individual result and to make sure that all the learners benefit from the interaction. The cooperative agent-based environment success is measured by the improvement of the average accuracy in the knowledge base of each agent.

The environment architecture has one component called Splitter, which separates the input set in two disjoints samples: one for learning and the other for final evaluation. This component is optional.

3.1. Agent Architecture

The agent architecture consists of five components depicted in Figure 2.

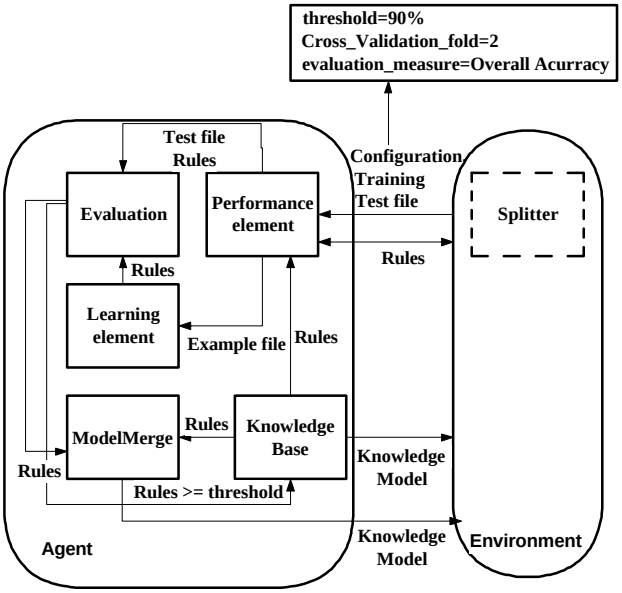

Figure 2 - Agent architecture

- Learning element: contains the rule induction technique. In the present work, the learning element contains two rule of these techniques: CN2 [10] and C4.5 [17]. The first approach extracts rules to describe the concepts contained in the data and the second produces a decision tree.
- Performance element: controls, monitors and guides the learning element progress. It is responsible for the data input/output between the agent itself and between the agent and the environment.
- Knowledge Base element: stores the rules generated by the learning element approved by the Evaluation element.
- ModelMerge element: has a role similar to a knowledge base. The ModelMerge element stores other agents' best rules together with the agent's own rules.

- Evaluation element: calculates the rules reliability. This is the element that decides whether or not to store the rules produced by the Learning element or those delivered by the Performance element.

3.2. Individual Learning

As learning happens in two stages, the first one is dedicated to the individual learning. The input here is the pre-processed training set and the configuration set. After that, the agent applies its rules inductor to the examples.

The objective of the individual learning is to create an individual domain model. This model is composed of rules approved by the Evaluation component in order to achieve a compact result.

As soon as the individual learning is over, the rules created are evaluated using the test file (data that had not been used to generate the model). The Evaluation element measures the quality of each rule by executing the CN2 rule evaluation function, and stores those that are equal or better than the threshold (informed in the configuration file) to the agent knowledge base. This function estimates the rule accuracy by applying the Laplace expected error estimate (Formula 1).

LaplaceAccuracy = (TP + 1)/(TP + FP + K) (1)

The formula depends on TP (true positives which means the number of examples correctly covered by the rule), FP (false positives which means the number of examples wrongly covered by the rule) and K (the number of classes in the domain).

Most of the time, the individual learning stage produces a rule set with well described concepts along with poor described ones. This happens due to the algorithm heuristics applied to the data for extracting knowledge.

Each Machine Learning algorithm that induces symbolic classifiers makes use of a proper syntax to describe the induced model. Since in the cooperative learning stage the agents will look for better rules, it is necessary to have them transformed into the same format. The transformation process takes place after the agent applies its algorithm to the training file, prior the evaluation step.

The format adopted in this work is called PBM [16] which look like this: *if* <*condition*> *then* <*concept* = C_i>. The PBM format has a library that converts some of the most common Machine Learning symbolic algorithms to its proper format. The transformation process followed by the agents is depicted in Figure 3.

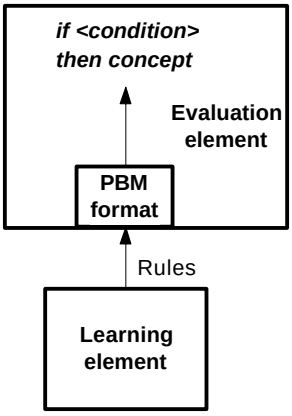

Figure 3 - Transformation process

3.3. Cooperative Learning

The goal of the cooperative learning is to improve the quality of the result. This stage input consists of the knowledge bases that store the knowledge obtained during the individual learning.

The cooperative learning consists of two further steps. During the first one, a agent queries other agents' knowledge bases. The first agent to start the interaction is the one that got the poorest overall accuracy (measured by the number of examples correctly classified by the agent's model). The agent searches for its equivalent rules with better quality. The rules that fill this requisite are added to the agent ModelMerge component. Each agent repeats this process from the poorer to the richer overall accuracy. We say that a rule is equivalent to another one when the two describe the same concept and the attributes used for them overlap. This way, a high quality rule is added to the agent's ModelMerge component either when it is similar, or overlaps, subsumes, or is in conflict with a low quality rule. For example, consider the rules R1 and R2 that describe concept C. The R1 rule contains the attribute-value test for attributes x and y, while the R2 rule includes tests for attributes x and z [21]. We than say that these two rules are related or equivalent.

When the communication ends, the rules taken from the knowledge base that were not changed are copied in the ModelMerge component. At this moment, each agent has two distinct models about the problem domain. It is necessary to evaluate the newest one which is stored in the ModelMerge component.

The agent incorporates the one (ModelMerge or Knowledge Base) that covered the highest number of instances from the test file and this becomes the final model (for that agent). The output generated by the environment is the best agent model at all.

3.4. Environment Implementation

The MASKS environment provides the definition of the parent class (Figure 4). Each Machine Learning algorithm is defined as a subclass of the parent class. The parent class provides the definitions that make the communication between the learner agents possible, and between the environment and its population of agents.

After describing the approach used to formulate the agent-based cooperative learning environment, we next show an application to the bioinformatics domain.

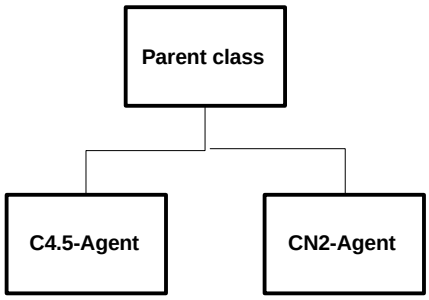

Figure 4 - Class Hierarchy of MASKS environment

4 APPLICATION OF THE ENVIRONMENT

The good results achieved in [14] has motivated us to use the same approach. Basically, the authors have developed a method to automate the process of annotation regarding the Keywords field in the SWISS-PROT database [2] which is based on the C4.5 algorithm. This field is a very important one, used mainly when a researcher wants to compare an unknown sequence s/he is working with, to the sequences already deposited in the SWISS-PROT.

Since dealing with the whole data at once would be prohibitive, all proteins from the SWISS-PROT database were divided in groups according to INTERPRO classification [1] and then, further in Keywords. The C4.5 algorithm expects an input in a tabular format where the last column contains the target (in this particular case the information whether a given keyword is present or not). The previous columns store data about the proteins like taxonomy details or the presence of sequence patterns.

Our approach is slightly different from the one used in [14] but the same applied in [4]. Instead of using all SWISS-PROT database we reduced our universe to the proteins related to *Mycoplasmataceae* family due to the PIGS project [22]. This project aims at providing the infrastructure for the sequencing of the genome of the *Mycoplasma hyopneumonia*.

The data collection was done in February 2002 by means of SRS web site (version 6 a www.srs6.ebi.ac.uk). Basically, we have performed a query on the SWISS-PROT and TrEMBL databases in which the Organism field included *Mycoplasmataceae.*

Also, we have created a view for the SWISS-PROT database which associates this database with the INTERPRO. This view included:

- for SWISS-PROT: AccNumber and Keywords
- for INTERPRO: AccNumber

The data gathered from the SWISS-PROT database was used for training purpose, and the TrEMBL data was used for test. The data was extracted from TrEMBL in the same way already explained regarding the extraction of data from the SWISS-PROT database. Unfortunately, regarding the *Mycoplasmataceae* family, a high number of proteins in both databases are classified as "hypothetical protein" and "complete proteome". Those were useless to our experiment, therefore we drop them out. As a result the final training data had 722 proteins or lines and 396 different INTERPRO characteristics or columns with 119 keywords (the test data had 952 proteins).

The training file is partially depicted in Table 1. Each line corresponds to a protein from the SWISS-PROT database and the columns, the presence or absence of sequence patterns (INTERPRO Accnumber).

To test the validity of MASKS, we applied two symbolic inductors - CN2 and C4.5 – on the Keyword field. We decided to work with just two inductors in order to give a didactic explanation of how the environment would function. Each algorithm induced one rule per Keyword with the format shown in Table 2.

The second stage, model validation, follows the individual model construction. It consists of translating each rule to the PBM format (Table 3) and then measuring its quality against the test file (according to the function explained in Section 3.2). The accuracy value of some rules appear in the last column of Table 4 and 5. For the sake of example, some concepts (keywords) mined by the CN2 and C4.5 algorithm are visible in the first column of Table 4 and 5 respectively.
When the communication (explained in Section 3.3) begins, the agent with the poorer overall accuracy starts by asking its colleagues if at least one of them has rules for class "Nucleotidyltransferase" (for instance). If yes, it asks for its quality. If the agent notices a superior quality in it, then it is necessary to check if the rules are related (explained in Section 3.3). If yes the agent adds the new rule to its ModelMerge component.

For example, the rule of Agent-C4.5 for the Keyword "Nucleotidyltransferase" has 36,36% accuracy (Table 5) which means, that around 30% of the examples in the test file were correctly classified by the rule. However, the Agent-CN2's rule for the same keyword has 45,45% accuracy (Table 4). As each agent has the goal of improving their knowledge by acquiring better rules, in the Nucleotidyltransferase case, Agent-C4.5 adds the rule to its ModelMerge component. All agents repeat this process until there are no more related rules to be exchanged.

Table 6 shows the average quality of the six rules, measured using the test file, produced by the agents during the individual and collaborative learning stages. Both Agents produced satisfactory results during individual learning: the Agent-C4.5 correctly predicted around 72% of the given Keywords, while Agent-CN2 predicted around 80%. The collaborative learning stage between them led to an equal individual final model. This happened because all rules were related. In the end, C4.5 learned more from the interaction (comparing Table 4 with Table 5 we can see that only the "Ligase" rule of C4.5 was better than CN2's) with quality increase near 10% in its final model (before exchange Agent-C4.5 rules covered around 72% of the instances present in the test file; after interaction with the Agent-CN2, this number increased to 81%).

Table 1 - Training file

Proteins	IPR000644	IPR001694	.....	IPR003617	Class
P47695	True	False		False	Yes
P75120	false	False		True	No
P47631	False	False		False	No
P75206	False	False		False	Yes
P47707	False	true		True	No
.....					

Table 2 - Rule generated for the Keyword Nucleotidyltransferase by CN2 and C4.5 algorithms respectively

CN2	C4.5
IF IPR001825 = no AND IPR002606 = no AND IPR004821 = no THEN class = no ELSE (DEFAULT) class = Nucleotidyltransferase	IPR004821 = TRUE: Nucleotidyltransferase IPR004821 = FALSE | IPR001825 = TRUE: Nucleotidyltransferase | IPR001825 = FALSE: no

Table 3 - PBM format example of the Keyword Nucleotidyltransferase

PBM Format	
CN2	**C4.5**
IF IPR001825 = no AND IPR002606 = no AND IPR004821 = no THEN class = no ELSE (DEFAULT) class = Nucleotidyltransferase	IF IPR004821 = TRUE THEN class = Nucleotidyltransferase ELSE IF IPR004821 = FALSE AND IPR001825 = TRUE THEN class = Nucleotidyltransferase

Table 4 - CN2 Rules Accuracy measured by its function before cooperation. Column TP stands for True positive and FN for False negative

CN2			
Keyword	**TP**	**FN**	**Accuracy**
Ligase	79	8	90.80%
Lyase	23	3	88.46%
Methyltransferase	24	7	77.42%
Nucleotidyltransferase	5	6	45.45%
Repeat	13	2	86.67%
Transport	40	1	97.56%

Table 5 - C4.5 Rules Accuracy measured by CN2 evaluation function before cooperation. Column TP stands for True positive and FN for False negative

C4.5			
Keyword	**TP**	**FN**	**Accuracy**
Ligase	81	6	93.10%
Lyase	21	5	80.77%
Methyltransferase	18	13	58.06%
Nucleotidyltransferase	4	7	36.36%
Repeat	10	5	66.67%
Transport	40	1	97.56%

Table 6 - Total Result

Overall Rule Set Accuracy			
CN2		C4.5	
Before exchange	**After exchange**	**Before exchange**	**After exchange**
81.06 %	81.44%	72.09%	81.44%

5 FUTURE WORK

The present version of the MASKS environment is still under construction. There are only two inductors in it and the communication is just simulated.

The communication among the agents will be implemented using KQML. Actually, we will develop the communication using the SACI Tool (Simple Agent Communication Infrastructure) [12] because it provides a transparent way of doing it following the KQML specification. SACI enables distributed agents to communicate in an easy way. The message

content will be expressed in the SQL language once the agents' knowledge bases will be constructed using MySQL [15]. A message example is shown below.

(ask-one

:ontology ML-ontology

:language SQL

:receiver CN2-agent

:sender C4.5-agent

:reply-with q1

:content "Select rules from CN2-agent.knowledge_base where concept = "Nucleotidyltransferase")

Initially four agents will be part of the MASKS environment – Agent-C4.5, Agent-CN2, Agent-Ripper [6] and Agent-T2 [3]. These Machine Learning algorithms are found in the MLC++ library [13], which was developed by the Stanford University to facilitate the usage of Machine Learning algorithms.

6 CONCLUSION

The target approach - intersection of distributed artificial intelligence and Machine Learning – provides a promising technology to address the complexity of modern information environments.

This paper describes the architecture of the MASKS environment that consists of several learning agents to induce rules from training examples. Agents cooperate to improve their knowledge by sharing it with others to achieve better results.

The main goal of this architecture is to preserve the learning bias of each Machine Learning algorithm, just improving the misleading rules by agent cooperation.

We have described an application in bioinformatics whose data were obtained from the SWISS-PROT and TrEMBL databases, for training and for test purposes respectively. Since we are interested in the annotation of keywords for proteins related to the *Mycoplasmataceae* family (subject of the PIGS project), we have used this data.

We have achieved a positive result from the initial validation of the MASKS environment. The final model constructed by it have produced an increase in quality.

7 ACKNOWLEDGEMENTS

We are grateful for the support of CNPq to the authors L.F. Schroeder and A.L.C. Bazzan, as well as to the PIGS project.

8 REFERENCES

[1] Apweiler R. et. al. (2001) InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucl. Acids. Res.* 29 (1):37-40.

[2] Apweiler R. Functional information in SWISS-PROT: The basis for large-scale characterisation of protein sequences. In *Bioinformatics* 2:9-18(2001).

[3] Auer P., Holte R. and Maass W. Theory and applications of agnostic pac-learning with small decision trees. In *Proc. of the 12th International Machine Learning Conference*, Morgan Kaufmann, 1995.

[4] Bazzan, A.L.C. et. al. Automated annotation of keywords for proteins related to mycoplasmataceae using machine learning techniques, *Bioinformatics*, vol. 18, no. 2, pp.1-9,2002.

[5] Bertone, P.; Kluger, Y.; Lan, N.; Zheng, D.; Christendat, D.; Yee, A.; Edwards, A.M.; Arrowsmith, C.H.; Montelione, G.T.; Gerstein, M.B. SPINE: An integrated tracking database and data mining approach for identifying feasible targets in high-throughput structural proteomics, *Nucleic Acids Res.*, vol. 29, no. 13, pp. 2884-2898, 2001.

[6] Cohen, W. W. Fast effective rule induction. In *Proc. of the 12th International Machine Learning Conference*, p. 115:123, San Francisco, CA. Morgan Kaufman, 1995.

[7] Decker, K. A Multi-Agent System for Automated Genomic Annotation. In *Proc. of the Int. Conf. Autonomous Agents*. Montreal, 2001.

[8] Drawid A., Gerstein M. A Bayesian system integrating expression data with sequence patterns for localizing proteins: compresensive application to the yeast genome, *J. Mol. Biol.*, vol. 301, pp. 1059-1075, 2000.

[9] Clare A., King R. Knowledge Discovery in Multi-Label Phenotype Data, In *European Conference on Principles of Data Mining and Knowledge Discovery* (PKDD), 2001.

[10] Clark P., Niblett T. The CN2 Induction Algorithm. In *Machine Learning Journal*, 3, pp 261-283, 1989.

[11] Dietterich, T.G. Limitations on inductive learning (extended abstract), 1997.
ftp://ftp.cs.orst.edu/pub/tgd/papers.

[12] Hübner, J. F. & Sichman, J. S. Saci Programming Guide, version 0.8, 2001.

[13] Kohavi, R. & Sommerfield, D. MLC++ Machine Learning library in C++, 1996, http://www.sgi.com/Technology/mlc.

[14] Kretschmann, E.; Fleischmann, W.; Apweiler, R. Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on Swiss-prot. *Bioinformatics*, vol. 17(10), p. 920:926, 2001.

[15] MySQL. The MySQL server, 2000. http://www.mysql.com.

[16] Prati, R. C. Baranauskas, J. A. & Monard, M. C. A Proposal for Unification of the Concept Representation Language to Symbolic Machine Learning Algorithms. Technical Report 137, ICMC-USP.
ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/RT_137.ps.zip (in Portuguese).

[17] Quinlan, J.R. C4.5: Programs for Machine Learning, Morgan Kafmann: San Mateo/CA, 1994.

[18] Ross-Macdonald, P.; Coelho, P.S.; Roemer, T.; Agarwal, S.; Kumar, A. et al. Large-scale analysis of the yeast genome by transposon tagging and gene disruption, *Nature*, vol. 402, pp. 413-418, 1999.

[19] Stolfo, S; Prodromidis, A.; Tselepis, S.; Lee, W.; Fan, D. JAM: Java Agents for Meta-Learning over Distributed Databases, In David Heckerman, Heikki Mannila, Daryl Pregibon, and Ramasarny Uthurusamy, editors, The Third International Conference on Knowledge Discovery & Data Mining. AAAI Press, 1997.

[20] Viktor, H. and H Arndt, Combining data mining and human expertise for making decisions, sense and policies, Journal of Systems and Information Technology, 4(2), pp.33-56.

[21] Viktor, H. The CILT multi-agent learning system, *South African Computer Journal (SACJ)*, 24, pp.171-181, 1999.

[22] Zaha, A. et al. Projeto rede sul de análise de genomas e biologia estrutural, 2001 (in Portuguese).